1. INTRODUCTION

Most coal mines in China are gas-bearing mines, and gas accidents occur frequently, which seriously threatens the safety of coal production.13 The monitoring and forecasting of gas concentrations is of great significance for preventing gas accidents and ensuring coal mine production safety. At present, coal companies have installed safety monitoring systems to monitor gas concentrations, but the functions performed by these systems are primarily disaster identification and response. The value of these monitoring data has not been fully exploited, resulting in an insufficient ability to predict and warn about gas accidents.

To improve the forecasting accuracy of coal mine gas concentrations, scholars around the world have performed a great deal of research. In general, gas concentration forecasting methods in recent years can be classified into two categories: methods based on traditional machine learning and methods based on deep learning. The methods based on traditional machine learning are as follows: He et al. proposed a gas content forecasting model based on linear quadratic exponential smoothing;3 Guo et al., Zhao et al., Ren, and Wang et al. proposed gas concentration forecasting methods based on the autoregressive integrated moving average model;4−7 Yao and Qiu proposed a mine gas concentration forecasting algorithm based on an improved back-propagation neural network;8 Zhou et al. proposed a differential gray model and radial basis function forecasting model for the gas content of deep coal seams;9 Li et al. and K. J. Wang and J. Y. Wang proposed a gas concentration forecasting model for a fully mechanized coal mining face based on a support vector machine;10,11 Gong and Li proposed a forecasting model for underground gas concentrations based on data fusion;12 Jia and Deng proposed a combined forecasting model for mine gas concentrations based on a pan-average calculation;13 and Zhang et al. proposed a dynamic gas concentration forecasting model.14 These studies have made certain contributions to the construction and optimization of mine gas forecasting models. However, the data sets targeted by these studies are relatively small. With the continuous increase in the amount of data in monitoring systems, building a model for big data to improve the accuracy and stability of forecasting gas concentrations is still an urgent problem to be solved.15

At present, the most effective type of method for processing big data is deep learning. Since the Hinton team won the ImageNet competition using a deep learning method in 2012,16 an increasing number of scholars have begun to pay attention to deep learning and apply it. In natural language processing,17,18 speech recognition,19 machine translation,20 image understanding,21 and other fields, breakthrough results have been achieved. In the view of the advantages of deep learning, some scholars have also applied it in the field of gas concentration forecasting. Li et al. preliminarily discussed the application of long short-term memory (LSTM) in coal mine gas forecasting and warning systems,22 which confirmed the effectiveness of LSTM in gas concentration forecasting. Sun et al. also performed dynamic forecasting of the
gas concentration in a mining face based on LSTM.\textsuperscript{23} Li et al., Ma et al., and Jia et al. predicted the gas concentration in a coal mining face based on recurrent neural network (RNN) and gated recurrent unit (GRU) models.\textsuperscript{24–26} These studies initially verified the effectiveness of deep learning models in the field of gas concentration forecasting. However, LSTM has many parameters and a complicated structure, which is prone to overfitting problems. As a variant of LSTM, a GRU has a simple structure, fewer parameters, and a short training time, which is more advantageous than LSTM. However, a GRU only considers the forward information in a sequence and does not consider the reverse information. In most cases, considering the reverse information can improve the forecasting accuracy of the model. The forward information implements the “consider the above information” function by adding the results of the previous operation to the current operation. The reverse information is the addition of the reverse operation to the GRU. It can be understood as reversing the input sequence and calculating the output in the way of GRU again. The final result is a simple stack of the result of forward GRU and reverse GRU.

Thus, the author proposes a forecasting model of gas concentration based on Adamax-BiGRU. First, the mine gas monitoring data are preprocessed, and then the Laida criterion\textsuperscript{27} and Lagrange interpolation method\textsuperscript{28} are applied to preprocess the outliers and missing values in the monitoring data; next, the monitoring data are spatially reconstructed to obtain the input of the GRU training sample. Second, based on the GRU model, the mean-square error (MSE) is used as the loss function, and the Adamax optimization algorithm is used to construct a two-way GRU learning model suitable for gas concentration time series and to determine the GRU model parameters for learning.

Finally, the gas concentration data of a coal mine working face are used for the test. The results show that compared with other deep learning models and optimization algorithms, the use of Adamax-BiGRU for gas concentration forecasting achieves good application results.

2. THEORY AND MODELING

2.1. Data Preprocessing. The Laida criterion and Lagrange interpolation method are applied to preprocess the outliers and missing values in the coal mine safety monitoring data.

2.1.1. Laida Criterion. Suppose that the measured value is measured with equal accuracy, and \( x_0, x_1, \ldots, x_n \) are obtained independently, and the arithmetic mean \( x \) and residual error \( \sigma = x_i - x \) \((i = 1, 2, \ldots, n)\) are calculated. The standard deviation \( \sigma \) is calculated according to Bessel’s formula, if the residual error \( |x_i| = x_i - x \leq 3\sigma \), it is considered that \( x_i \) is a bad value with a gross error, and therefore should be eliminated.

2.1.2. Lagrange Interpolation. In general, if the function value \( y_0, y_1, \ldots, y_n \) of \( y = f(x) \) at different \( n + 1 \) points \( x_0, x_1, \ldots, x_n \) is known, then one can consider constructing a polynomial whose degree does not exceed \( n \) through these \( n + 1 \) points to satisfy the formula

\[
P(x_k) = y_k, \quad k = 0, 1, \ldots, n
\]

To estimate any point \( x \neq x_k \), \( k = 0, 1, \ldots, n \), you can use the value of \( P_n(x) \) as the approximate value of the exact value \( f(x) \). This method is called "interpolation".

The formula \( 1 \) is the interpolation condition (criteria), including the minimum interval \( [a, b] \) of \( x_i \) \((i = 0, 1, \ldots, n)\), where

\[a = \min\{x_0, x_1, \ldots, x_n\}, \quad b = \max\{x_0, x_1, \ldots, x_n\}.
\]

2.2. Bidirectional Gated Circulation Unit. A GRU is a variant of an RNN. An RNN has a “memory” function for processing sequence data. It adopts a chain structure, taking the “memory” of the previous moment and the state of the current moment as the input information, and enters a loop structure to obtain a new output. Because an RNN repeats the same operation at each time point, it has a very deep calculation graph. Therefore, similar to multilayer neural networks, RNNs face the problems of gradient explosion, gradient disappearance, and long-term dependence. To solve these problems for RNNs, Hochreiter and Schmidhuber proposed LSTM in 1997.\textsuperscript{29} Like an RNN, an LSTM neural network also has a chain structure, but compared with the simple layer of neurons in the RNN, the LSTM neuron cycle module has more parameters, its structure is more complicated, and it is not easy for it to converge. To overcome the shortcomings of LSTM, Cho proposed the LSTM variant GRU in 2014.\textsuperscript{30} The GRU simplifies LSTM, so its calculation speed is much faster than that of LSTM.

The classic GRU structure uses unidirectional propagation along the sequence transfer direction, and each time \( t \) is only related to past times. However, in some cases, when building a model, if the feedback of a future sequence value at a certain moment is considered, the model can be modified with this future information. Therefore, a two-way GRU model is built, as shown in Figure 1. Its basic idea is as follows: for each training sequence, two RNN models are established in the forward and backward directions, and the hidden-layer nodes of the two models are connected to the same output layer. This data processing method can provide complete historical and future information for each time point in the input sequence of the output layer.

In Figure 1, each GRU neural unit is processed in two directions: GRU\(^1\) is a forward GRU, and its internal structure is shown in Figure 2; GRU\(^2\) is a reverse GRU, and its internal structure is shown in Figure 3.

The forward calculation in Figure 2 proceeds as follows:

Suppose \( \vec{z}_t \) is the reset gate of the positive input GRU at time \( t \). The formula is as follows

\[
\vec{z}_t = \sigma(\vec{W}_t \vec{x}_t + \vec{U}_t \vec{h}_{t-1})
\]

In the formula, \( \sigma \) is the sigmoid function; \( \vec{x}_t \) and \( \vec{h}_{t-1} \) are the current input value and the last activation value, respectively; \( \vec{W}_t \) is the input weight matrix; and \( \vec{U}_t \) is the weight matrix for cyclic connections.

Similarly, suppose \( \tilde{z}_t \) is the update gate of the forward GRU at time \( t \); the formula is as follows

\[
\tilde{z}_t = \sigma(\vec{W}_t \vec{x}_t + \vec{U}_t \vec{h}_{t-1})
\]
Suppose $\vec{h}_t$ is the activation value of the positive input GRU at time $t$, which is a compromise between the last activation value $\vec{h}_{t-1}$ and the candidate activation value $\vec{h}_{\tilde{t}-1}$.

$$\vec{h}_t = (1 - \vec{z}_t) \vec{h}_{t-1} + \vec{z}_t \vec{h}_{\tilde{t}}$$  \hspace{1cm} (4)

The formula for $\vec{h}_{\tilde{t}}$ is as follows

$$\vec{h}_{\tilde{t}} = \tanh (\vec{W}_h \vec{x}_t + \vec{r}_t \vec{U}_h \vec{h}_{t-1})$$  \hspace{1cm} (5)

In the above formula, $\cdot$ is the Hadamard product.

When the reset gate $\vec{r}_t$ is closed, that is, its value is close to 0, the GRU ignores the previous activation value $\vec{h}_{t-1}$ and is only affected by the current input $\vec{x}_t$. This allows $\vec{h}_t$ to discard irrelevant information.

On the other hand, the update gate $\vec{z}_t$ controls how much information in $\vec{h}_{t-1}$ can be passed to the current $\vec{h}_t$. This is the key to the design of the results of this unit. Its function is similar to that of the memory unit in LSTM, which can help the GRU remember long-term information.

Similarly, the reverse GRU calculation formula in Figure 3 is shown in formulas 6-9.

$$\vec{r}_t = \sigma (\vec{W}_e \vec{x}_t + \vec{U}_e \vec{h}_{t+1})$$  \hspace{1cm} (6)

$$\vec{z}_t = \sigma (\vec{W}_g \vec{x}_t + \vec{U}_g \vec{h}_{t+1})$$  \hspace{1cm} (7)

$$\vec{h}_{t+1} = (1 - \vec{z}_t) \vec{h}_{t+1} + \vec{z}_t \vec{h}_t$$  \hspace{1cm} (8)

$$\vec{h}_{\tilde{t}+1} = \tanh (\vec{W}_h \vec{x}_t + \vec{r}_t \vec{U}_h \vec{h}_{t+1})$$  \hspace{1cm} (9)

The results of the two directions are averaged to obtain the final output $h_r$.

$$h_t = (\vec{h}_t + \vec{h}_{\tilde{t}})/2$$  \hspace{1cm} (10)

2.3. Forecasting Model Structure of the Gas Concentration. The change in the mine gas concentration monitoring data is time-varying and nonlinear and cannot be predicted by a simple linear relationship. A neural network has obvious advantages in predicting complex nonlinear time-varying sequences. Therefore, a gas concentration forecasting model based on a BiGRU is constructed to predict the trend of the mine gas concentration.

For a mine gas concentration time series $X$, the structure of the forecasting model based on an optimized BiGRU is designed as shown in Figure 4.

Among them, the input is the gas concentration time series, the learning parameters are learned through BiGRU, and the output is the predicted gas data.

3. MODEL OPTIMIZATION

Deep learning is a highly iterative process, and training takes a long time. To shorten the model convergence time, researchers have proposed a variety of optimization algorithms. The most commonly used deep learning optimization algorithm is the gradient descent method and its variants, including batch gradient descent, stochastic gradient descent (SGD), gradient descent with momentum (GDM), adaptive gradient descent (AdaGrad), adaptive delta gradient descent (AdaDelta), Nesterov accelerated gradient (NAG) descent, and other methods. However, the gradient descent method has the disadvantages that it is difficult to choose a suitable learning rate, it easily converges to a local optimum, and the convergence process produces fluctuations. Therefore, some scholars have proposed algorithms such as root-mean-square-prop (RMSProp), adaptive moment estimation (Adam), and adaptive moment estimation max (Adamax); the Adam algorithm in particular has achieved good results. Because Adam is an RMSProp with a momentum term, it is good at handling nonstationary targets and has low memory requirements. It can calculate different adaptive learning rates for different parameters. Adamax is an improved version of Adam and often achieves better results than Adam. Therefore, the Adamax algorithm is employed in this study. The results of this study are more stable and accurate.
is proposed to optimize the gas concentration BiGRU forecasting model.

3.1. Optimization Framework for a Deep Neural Network. The parameter to be optimized in the DNN is $\omega$, and the objective function is $f(\omega)$. The optimization problem in deep learning can be described as finding a suitable set of parameters $\omega$ to minimize the loss function. To find this set of parameters more quickly, we often update the parameters by calculating the gradient of the objective function with respect to the current parameters.

The optimization framework of DNNs can be described as follows: the initial learning rate is set to $w$, and then iterative optimization begins. In the $t$-th training cycle (epoch), the update parameters are calculated using the following steps:

1. Calculate the gradient $g_t$ of the objective function with respect to the current parameters:

   $$g_t = \nabla f(\omega)$$

2. Calculate the first-order momentum $m_t$ and second-order momentum $V_t$, based on the historical gradient, different optimization algorithms use different formulas:

   (a) SGD: The characteristic is that it does not use momentum, converges slowly, and easily falls into local extremes, so its gradient update $m_t$ directly equals $g_t$.

   $$m_t = g_t$$  \hspace{1cm} (12)

   $$\eta_t = \alpha \cdot m_t$$  \hspace{1cm} (13)

   (b) SGD with momentum: uses the momentum of the gradient and converges faster than SGD.

   $$m_t = \beta \cdot m_{t-1} + (1 - \beta) \cdot g_t$$  \hspace{1cm} (14)

   $$\eta_t = \alpha \cdot m_t$$  \hspace{1cm} (15)

   (c) Adagrad: The second-order momentum is used, and the step size can be adjusted adaptively, but its second-order momentum accumulates the entire history, and the learning may be stopped early.

   $$v_t = \sum_{i=1}^{t} g_i^2$$  \hspace{1cm} (16)

   (d) AdaDelta/RMSProp: Second-order momentum is used, but its update method is different from Adagrad. AdaGrad accumulates all historical second-order momentum, and AdaDelta accumulates part of it. It can avoid ending the study early.

   $$v_t = \beta \cdot v_{t-1} + (1 - \beta) \cdot g_t^2$$  \hspace{1cm} (17)

   (e) Adam: Adam combines the first-order momentum with the second-order momentum, which greatly accelerates the convergence rate. Its first-order momentum update method is the same as SGD with momentum; the second-order momentum update method is the same as AdaGrad.

   $$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$  \hspace{1cm} (18)

   $$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$$  \hspace{1cm} (19)

   (f) Adamax: Adamax is a variant of Adam. This method provides a simpler range formula change on the upper limit of the learning rate.

   $$\eta_t = \max(\nu \times |\eta_{t-1}|, |g_t|)$$  \hspace{1cm} (20)

3. Calculate the descending gradient $\eta_t$ at the current moment:

   $$\eta_t = \alpha \cdot m_t/\sqrt{V_t}$$  \hspace{1cm} (21)

4. Update $\omega$ according to the descending gradient:

   $$\omega_{t+1} = \omega_t - \eta_t$$  \hspace{1cm} (22)

The difference between the different optimization algorithms is reflected in the different calculation methods of $g_t$, $m_t$, and $V_t$, in steps (1) and (2).

3.2. Adamax Optimization Algorithm. The Adamax algorithm is a variant of Adam, and it sets a simpler range for the upper limit of the learning rate. In Adamax, the gradient of formula 11 is calculated using formula 23:

   $$g_t = \nabla^2 f(\omega_{t-1})$$  \hspace{1cm} (23)

The first-order momentum is calculated using formula 24:

   $$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$  \hspace{1cm} (24)

The second-order momentum is calculated using formula 25:

   $$V_t = \max(\beta_2 \cdot V_{t-1}, |g_t|)$$  \hspace{1cm} (25)

Among these variables, $\beta_1$ and $\beta_2$ are hyperparameters that control the first-order momentum and second-order momentum, respectively. Its value is determined by optimization algorithm.

Formula 23 is substituted into formulas 24 and 25, and then formulas 24 and 25 are substituted into formulas 21 and 22 to update the parameter $\omega$.

4. EXPERIMENT

4.1. Experimental Data. A total of 103,598 monitoring data points obtained from a comprehensive coal mining face from 16:25:00 on January 2, 2019, to 23:50:00 on December 31, 2019, at a sampling interval of 5 min were used as the experimental data. Because of the partial absence of the original data, 1142 data points were missing. The original data were preprocessed by applying the Laida criterion and Lagrange interpolation method, and the number of processed data points was 104,740. The average value of the preprocessed data was 0.143033; the standard deviation was 0.083732; the minimum value was 0; and the maximum value was 0.984. The data were divided into a training set and a test set according to the ratio 9:1, and there were no missing or imputed data in the test set.

4.2. Evaluation Index. To test the effectiveness of the gas concentration forecasting method based on Adamx-BiGRU, a certain index must be used to comprehensively measure and evaluate the forecasting effect. Here, the MSE is used to evaluate the model.

Models using an RNN and LSTM are compared, as well as optimization algorithms using SGD, SGDM, NAG, RMSProp, AdaGrad, AdaDelta, and Adam.

4.3. Experimental Steps. The experiment is performed with the following steps:

   Step 1: Data preprocessing. First, the Laida criterion is applied to address the noise in the gas concentration monitoring data.

   Second, the Lagrange interpolation method is used to interpolate the missing values of the monitoring data to obtain the processed monitoring data.

   Finally, the values of the application data set are normalized to $[0, 1]$.
Step 2: The gas concentration time series monitoring data are spatially reconstructed, and input samples are constructed for training the Adamax-BiGRU model.

Step 3: The structure of the BiGRU neural network is determined—that is, the number of hidden layers of the BiGRU, the number of neurons in each layer, and the value of the epoch.

Step 4: In order to update and calculate the network parameters that affect the training and output of the BiGRU model, make it approximate or reach the optimal value, thereby minimizing the loss function. The Adamax optimization algorithm is applied to optimize the BiGRU model and establish a forecasting model.

Step 5: The model forecasting effect on the test set is verified, and the forecasting data is denormalized to visualize the forecasting result with MSE.

The MSE is the expected value of the square of the difference between the estimated value of the parameter $\hat{y}_t$ and the true value of the parameter $y_t$. The calculation formula is as follows:

$$\text{MSE} = \mathbb{E}[(\hat{y}_t - y_t)^2]$$

Figure 5. Error of the training set in the BiGRU with different epochs.

Figure 6. Prediction effect of the RNN model on the test set. (a) All data of the test set. (b) 6000th to the 6500th data of the test set.

Figure 7. Prediction effect of the LSTM model on the test set. (a) All data of the test set. (b) 6000th to the 6500th data of the test set.
Through the above five steps, gas concentration forecasting based on the Adamax-BiGRU model can be performed.

5. RESULTS AND DISCUSSION

5.1. Determination of the Model Parameters. To determine the parameters of the Adamax-BiGRU model, the convergence of the MSE on the training set is compared for different parameter values to select the optimal number of hidden layers, the number of nodes in the hidden layers and the number of iterations. The following parameters are determined by comparison: the number of hidden layers is 2; the number of neurons is 32; the value of the batch size is 20; the learning rate is 0.01; and the number of iterations is 5.

First, the number of training rounds is preset to epochs = 100; then, the error of the BiGRU on the training set for each round is shown in Figure 5. It can be seen from Figure 5 that after epochs = 40, the training error and verification error are basically stable. Therefore, setting epochs = 40 can meet the training requirements.

5.2. Comparison of Different Models. The same structure is constructed based on the RNN, LSTM, GRU, and BiGRU gas concentration forecasting models, and the optimization algorithm uses Adam. The experimental results of the RNN, LSTM, GRU, and BiGRU models on the test set are shown in Figures 6–9. All data are shown in subgraph (a), and the 6000th to the 6500th data of the test set.

\[
\text{MSE} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - \hat{\hat{y}}_i)^2
\]  \hspace{1cm} (26)

Figure 8. Prediction effect of the GRU model on the test set. (a) All data of the test set. (b) 6000th to the 6500th data of the test set.

Figure 9. Prediction effect of the BiGRU model on the test set. (a) All data of the test set. (b) 6000th to the 6500th data of the test set.
are shown in Subgraph (b). The data comparison of the experimental results is shown in Table 1.

From Figures 6−9, it can be seen that the forecasting effect of the RNN is poor, followed by LSTM, and the forecasting effect of the GRU and BiGRU models is good.

As seen from Table 1, the running time of the BiGRU is the same as that of the RNN, but the error of the training and test sets is reduced by 18.97 and 25.58%, respectively; compared to LSTM, the running time of the BiGRU is reduced by 25%, and the error of the training set and test set is reduced by 11.61 and 12.53%, respectively. Compared with the unidirectional GRU model, the running time is consistent, but the error of the training set and test set is reduced by 4.19 and 3.01%, respectively. Taken together, the BiGRU achieves the best forecasting results.

5.3. Comparison of the Different Optimization Algorithms. To further improve the forecasting effect of the BiGRU model, the GRU model is optimized. The experimental results of different optimization algorithms based on the GRU model are shown in Table 2, and the experimental results of different optimization algorithms based on the BiGRU model are shown in Table 3.

As seen in Tables 2 and 3, compared with other optimization algorithms, Adamax achieves the best forecasting effect on both the training set and the test set. Compared with the Adamax-GRU model and the Adamax-BiGRU model, the Adamax-BiGRU model achieves better forecasting results.

6. CONCLUSIONS AND THE FUTURE WORK

(1) A gas concentration forecasting algorithm based on Adamax-BiGRU is proposed, which forecasts gas concentration time series. (2) By comparing optimization algorithms, such as SGD, SGDM, NAG, RMSProp, AdaGrad, AdaDelta, Adam, and Adamax, the optimization algorithm for the BiGRU model that is most suitable for mine gas concentration data forecasting is determined to be Adamax. (3) The application to an actual data set shows that compared with the RNN and LSTM models, the Adamax-BiGRU model achieves better forecasting results with a lower time complexity and has greater practical application value. (4) In future work, the internal network structure of the BiGRU model will be further studied and optimized to achieve faster convergence speed and prediction accuracy.

AUTHOR INFORMATION

Corresponding Author

Rong Liang − Department of Safety Science and Engineering and Department of Computer Science and Technology, Xi’an University of Science and Technology, Xi’an 710054, China; orcid.org/0000-0001-7671-1640; Email: liangr@xust.edu.cn

Authors

Xintan Chang − Department of Safety Science and Engineering, Xi’an University of Science and Technology, Xi’an 710054, China
Pengtao Jia − Department of Computer Science and Technology, Xi’an University of Science and Technology, Xi’an 710054, China
Chengxiong Xu − Department of Computer Science and Technology, Xi’an University of Science and Technology, Xi’an 710054, China

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.0c03417

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This work was supported in part by the State Key Research Development Program of China (grant no. 2018YFC0808303), the National Natural Science Foundation of China (grant no. 51974236), and the Scientific and Technology Program Funded by Xi’an City (grant no. 2020KJRC0069).

REFERENCES

(1) Liu, Y. J.; Yuan, L.; Xue, J. H.; Tian, Z. C. Analysis on the occurrence law of gas disaster accidents in coal mine from 2007 to 2016. Min. Saf. Environ. Protect. 2018, 45, 124−128.
(2) Yuan, L. Innovation and development of safety science and technology in coal industry of China. Saf. Coal Mines 2015, 46, 5−11.
(3) He, J.; Liu, L. B.; Tang, Y. J. Gas content prediction based on linear double exponential smooth method. Coal Sci. Technol. 2014, 42, 48−50.
(4) Guo, S. W.; Tao, Y. F.; Li, C. Dynamic forecasting of gas concentration based on time series. Ind. Mine Autom. 2018, 44, 20−25.
(5) Zhao, M. C.; He, A. M.; Qu, S. J. Research on time series forecasting method of gas data on fully mechanized mining face. Ind. Mine Autom. 2019, 45, 80−85.
(6) Ren, X. D. Application research of gas concentration forecasting and early earning model in Luming Mine. Coal Miner. Mod. 2020, 2, 47−49.
(7) Wang, P.; Wu, Y. P.; Wang, S. L.; Wu, X. M. Study on Lagrange-ARIMA real-time forecasting model of mine gas concentration. Coal Sci. Technol. 2019, 47, 141−146.

Table 1. Comparison of the Experimental Results of Different Models

| model  | running time/s | MSE of the training set | MSE of the test set |
|--------|----------------|-------------------------|---------------------|
| RNN    | 4              | 0.000564                | 0.000563            |
| LSTM   | 5              | 0.000517                | 0.000479            |
| GRU    | 4              | 0.000477                | 0.000432            |
| BiGRU  | 4              | 0.000457                | 0.000419            |

Table 2. Comparison of the Experimental Results of Different Optimization Algorithms for the GRU Model

| model  | running time/s | MSE of the training set | MSE of the test set |
|--------|----------------|-------------------------|---------------------|
| SGD    | 3              | 0.000899                | 0.000836            |
| SGDM   | 3              | 0.000470                | 0.000435            |
| NAG    | 4              | 0.000469                | 0.000433            |
| RMSProp| 3              | 0.000459                | 0.000421            |
| AdaGrad| 4              | 0.000478                | 0.000439            |
| AdaDelta| 4            | 0.000474                | 0.000436            |
| Adam   | 4              | 0.000463                | 0.000432            |
| Adamax | 4              | 0.000458                | 0.000423            |

Table 3. Comparison of the Experimental Results of Different Optimization Algorithms for the BiGRU Model

| model  | running time/s | MSE of the training set | MSE of the test set |
|--------|----------------|-------------------------|---------------------|
| SGD    | 3              | 0.000544                | 0.000497            |
| SGDM   | 3              | 0.000467                | 0.000433            |
| NAG    | 5              | 0.000460                | 0.000428            |
| RMSProp| 3              | 0.000455                | 0.000415            |
| AdaGrad| 4              | 0.000471                | 0.000432            |
| AdaDelta| 4            | 0.000477                | 0.000433            |
| Adam   | 4              | 0.000457                | 0.000419            |
| Adamax | 4              | 0.000447                | 0.000413            |
(8) Yao, Q. H.; Qiu, B. H. Prediction algorithm of gas concentration in coal mine based on improved BP neural network. Coal Technol. 2017, 36, 182–184.

(9) Zhou, X. L.; Zhang, G.; Lyu, C.; Huang, H. A grey model for predicting the gas content in the deep coal seam and its application via the neural network of the difference radial basis function. J. Saf. Environ. 2017, 17, 2050–2055.

(10) Li, H.; Jia, J.; Yang, X. Y.; Song, C. R. Gas concentration forecasting model for fully mechanized coal mining face. Ind. Mine Autom. 2018, 44, 48–53.

(11) Wang, K. J.; Wang, J. Y. Forecasting of gas concentration based on SVM optimized by memetic algorithm. Coal 2020, 29, 4–6.

(12) Gong, S. F.; Li, Y. S. Forecasting of underground gas concentration state based on data fusion. J. Xi’an Univ. Sci. Technol. 2018, 38, 506–514.

(13) Jia, P. T.; Deng, J. Coal mine gas concentration Combination prediction model based on universal average operation. China Saf. Sci. J. 2012, 22, 41–46.

(14) Zhang, Z. Z.; Qiao, J. F.; Yu, W. Forecasting coalmine gas concentration based on dynamic neural Network. Contr. Eng. China 2016, 23, 478–483.

(15) Wang, G. F.; Wang, H.; Ren, H. W.; Zhao, G. R.; Pang, Y. H.; Du, Y. B.; Zhang, J. H.; Hou, G. 2025 scenarios and development path of intelligent coal mine. J. China Coal Soc. 2018, 43, 295–305.

(16) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. Adv. Neural Inf. Process. Syst. 2012, 25, 1097–1105.

(17) Sak, H.; Senior, A. W.; Beaufays, F. Long short term memory based recurrent neural network architectures for large vocabulary speech recognition. Comput. Sci. 2014, 13, 338–342.

(18) Sundermeyer, M.; Schlüter, B.; Ney, H. LSTM neural networks for Language modeling. Interspeech 2012, 31, 601–608.

(19) Nammous, M. K.; Saeed, K.; Kobojek, P. Using a small amount of text-independent speech data for a BiLSTM large-scale speaker identification approach. J. King Saud Univ. Comput. Inf. Sci. 2020, DOI: 10.1016/j.jksuci.2020.03.011.

(20) Sánchez-Gutierrez, M. E.; González-Pérez, P. P. Discriminative neural network pruning in a multiclass environment: A case study in spoken emotion recognition. Speech Commun. 2020, 120, 20–30.

(21) Liu, C.; Wang, C.; Sun, F. C.; Rui, Y. Image2 Text: a multimodal caption generator. ACM Multimedia, 2016; pp 746–748.

(22) Li, W. S.; Wang, L.; Wei, C. Application and design of LSTM in coal mine gas forecasting and warning system. J. Xi’an Univ. Sci. Technol. 2018, 38, 1027–1035.

(23) Sun, Z. Y.; Cao, Y. L.; Yang, D.; Han, C. J.; Ma, X. M.; Zhao, Y. R. Dynamic forecasting of gas concentration in mining face based on long short-term memory neural network. Saf. Coal Mines 2019, 50, 152–157.

(24) Li, S. G.; Ma, L.; Pan, S. B.; Shi, X. L. Research on forecasting model of gas concentration based on RNN in coal mining face. Coal Sci. Technol. 2020, 48, 33–38.

(25) Ma, L.; Pan, S. B.; Dai, X. G.; Song, S.; Shi, X. L. Gas concentration forecasting model of working face based on PSO-Adam-GRU. J. Xi’an Univ. Sci. Technol. 2020, 40, 363–368.

(26) Jia, P.; Liu, H.; Wang, S.; Wang, P. Research on a mine gas concentration forecasting model based on a GRU network. IEEE Access 2020, 8, 38023–38031.

(27) Yang, Z.; Tian, L.; Li, C. A Fast Video Shot Boundary Detection Employing OTSU’s Method and Dual Pauta Criterion. 2017 IEEE International Symposium on Multimedia (ISM), Taichung, 2017; pp 583–586.

(28) Li, X.; Zhao, J.; Zhao, X.; Xu, J. Application of an Improved Lagrangian Interpolation Prediction Method in Continuous Power Flow Calculation. 2019 6th International Conference on Systems and Informatics (ICSAI), Shanghai, China, 2019; pp 209–212.

(29) Hochreiter, S.; Schmidhuber, J.; Schmidhuber, J. Long short-term memory. Neural Comput. 1997, 9, 1735–1780.

(30) Cho, K.; Van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. 2014, arXiv:1406.1078 [cs.CL].