IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 22, 2019, accepted May 29, 2019, date of publication June 7, 2019, date of current version June 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2921578

Air Pollution Forecasting Using a Deep Learning Model Based on 1D Convnets and Bidirectional GRU

QING TAO[1], FANG LIU[1], (Member, IEEE), YONG LI[2], (Senior Member, IEEE), AND DENIS SIDOROV[3], (Senior Member, IEEE)

[1]School of Automation, Central South University, Changsha 410083, China
[2]College of Electrical and Information Engineering, Hunan University, Changsha 410082, China
[3]Energy Systems Institute of Russian Academy of Sciences, 130 Lermontov Str., Irkutsk 664033, Russia

Corresponding author: Fang Liu (csuliufang@csu.edu.cn)

This work was supported in part by the National Natural Science Foundation of Hunan Province of China under Grant 2018JJ2529, in part by the NSFC-RFBR Exchange Program under Grant 61911530132/195853011, in part by the Fundamental Research Funds for the Central Universities of Central South University under Grant 2019zzts567, and in part by the Fundamental Research of Siberian Branch of the Russian Academy of Sciences under State Assignment, Project 17.3 (reg. no. AAAA-A17-117030310442-8).

ABSTRACT Air pollution forecasting can provide reliable information about the future pollution situation, which is useful for an efficient operation of air pollution control and helps to plan for prevention. Dynamics of air pollution are usually reflected by various factors, such as the temperature, humidity, wind direction, wind speed, snowfall, rainfall, and so on, which increase the difficulty in understanding the change of air pollutant concentration. In this paper, a short-term forecasting model based on deep learning is proposed for PM2.5 (particulate matter with an aerodynamic diameter less than or equal to 2.5 μm) concentration, and the convolutional-based bidirectional gated recurrent unit (CBGRU) method is presented, which combines 1D convnets (convolutional neural networks) and bidirectional GRU (gated recurrent unit) neural networks. The case is carried out by using the Beijing PM2.5 data set in UCI Machine Learning Repository. Comparing the prediction results with the traditional ones, it is proved that the error of the CBGRU model is lower and the prediction performance is better.

INDEX TERMS Air pollution forecasting, deep learning, 1D convolutional neural networks, bidirectional gated recurrent unit.

I. INTRODUCTION

Nowadays, many cities have suffered from massive smog attacks, which have affected people's daily life and caused serious harm to their health. The main component of smog is the Particulate Matter (PM) 2.5. The primary task of dealing with smog pollution and improving air quality is to control PM2.5, so the PM2.5 concentration prediction is the main content of air quality prediction. It is of great significance to identify the evolution law of PM2.5 concentration and achieve efficient and accurate prediction for air pollution prevention and control.

The concentration of PM2.5 is often related to various meteorological factors, so the prediction of PM2.5 is actually a multivariate time series prediction problem. Till now, various air quality forecasting approaches have been proposed, which can be mainly classified into the statistical methods, the shallow machine learning methods and the deep learning methods. Statistical methods include correlation coefficient method, principal component analysis method, Newton interpolation method [1], nonlinear regression model [2], and so on. Accuracy obtained is limited in these methods because of their inability to model non-linear and multivariate data. Shallow machine learning methods include multilayer perceptron (MLP), radial basis function (RBF) [3], genetic algorithm (GA) [4], support vector machines (SVM) [5], artificial neural networks (ANN) [6], and so on.

In recent years, with the development of deep learning and big data technology, the use of deep learning methods for air quality prediction has become an active research field, and the commonly used models are recurrent neural networks (RNN)

The associate editor coordinating the review of this manuscript and approving it for publication was Javier Medina.

and its variations. Long Short-Term Memory Unit (LSTM), as a state-of-the-art model of RNN, is used in the air quality forecasting [7], [8]. Besides, manifold learning method and deep belief network [9], deep uncertainty learning [10] and Encoder-Decoder model [11] are also used for PM2.5 pollution concentration. Recently, GRU (gated recurrent unit) is applied to the PM2.5 forecasting task and is performing well [12].

In view of the dynamic instability and long-term dependence of the time series of air pollutants, a model combining the recurrent neural networks and the convolutional neural networks is proposed for air pollution forecasting in this paper, which comprehensively utilizes the ability of feature extraction of convolutional neural networks and the capability of time series forecasting of recurrent neural networks. As a first, the convolution neural networks is used to carry out downsampling of data to reduce the size and complexity of data and improve the generalization and learning ability of the model. Then, the reduced-dimensional data are fed into the recurrent neural networks to further mine the information characteristics provided by different data sources in meteorological data, and establish the nonlinear relationship between the time series of multivariable and air pollutant PM2.5. In order to verify the effectiveness of the proposed method, we analyze a Support Vector Regression (SVR), Gradient Boosting Regressor (GBR), Decision Tree Regressor (DTR), simple RNN, Long Short-Term Memory Networks (LSTM), Gated Recurrent Unit (GRU) and bidirectional Gated Recurrent Unit (BGRU), and all models are compared regarding their forecasting performance of PM2.5 concentration.

The remainder of this article is organized in the following way: In Section II, we highlight data description and correlation analysis of PM2.5 time series. Section III outlines the framework of the PM2.5 forecasting model based on 1D convnets and bidirectional GRU. In Section IV, we describe our experimental setup and results. Finally, the conclusion of this article is given in Section V.

II. DATA AND CORRELATION ANALYSIS OF PM2.5 TIME SERIES

A. DATA DESCRIPTION

The proposed forecasting approach is tested by using the database from UCI machine learning repository [13], which contains the PM2.5 data of US Embassy in Beijing located at (116.47 E, 39.95 N) and meteorological data from Beijing Capital International Airport. Although the embassy and the airport are 17 km apart, they experience very much the same weather. This dataset covers hourly data from January 1, 2010 to December 31, 2014, contains 8 characteristics including PM2.5 concentration, dew point, temperature, air pressure, wind direction, wind speed, snowfall, and rainfall. Eliminate missing points in the data, the total amount of data is 43,800 rows, select the first 30,000 rows of data as training set, 30001-38000 rows as validation set and, 38001-43800 rows as test set. The attribute of wind direction

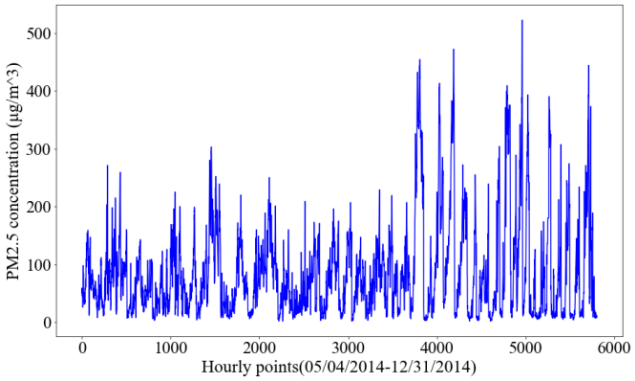


FIGURE 1. PM2.5 concentration change on test set.

in the data contains 4 features: NW, CV, SE and NE, which need to be encoded as float data, assigned to −10, 0, 10 and 20 respectively. In addition, for the few missing values of PM2.5 pollution in the data set due to sensor errors, we filled them in accordance with the data of the previous timestamp. Finally, the entire dataset is normalized by subtracting the mean of each feature and dividing by the variance of each feature:

$$x_{std}^{i} = \frac{x^{i} - x_{mean}^{i}}{\sigma_{x}^{i}} \tag{1}$$

where, x_{mean}^{i} and σ_{x}^{i} are the mean and variance of the i-th characteristic variable, respectively. It should be noted that the calculation of the mean and variance in this equation is only for the training set, because in reality the distribution of the validation set and the test set are unknown.

The Figure 1 shows the actual situation of the PM2.5 concentration on the test set. It can be seen from this plot that there is no obvious periodic law in the trend of change, and the fluctuation range is large.

B. CORRELATION ANALYSIS OF PM2.5 TIME SERIES

To develop a good prediction model, it is crucial to identify the correlation between the various influencing factors and the PM2.5 concentration before the model is built, which ensures that the model uses the proper input prognostic features for prediction. PM2.5 is affected by many measurable factors, but not all of them are effective for the prediction task, and the irrelevant factors will become burdensome for the model. Therefore, we need to calculate the correlation coefficient between each factor and the target feature, and judge the correlation between PM2.5 concentration and the selected feature indirectly via the value of the correlation coefficient. Suppose one characteristic time series is the vector $\boldsymbol{X}=(x_1, x_2, \ldots, x_n)$, the other time series is vector $\boldsymbol{Y}=(y_1, y_2, \ldots, y_n)$, the correlation coefficient r between them is calculated by formula (2).

$$r = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2}} \tag{2}$$

TABLE 1. Correlation coefficient (R) between meteorological variables and PM2.5 concentration.

R	Pollution	Dew point	Temperature	Pressure	Wind direction	Wind speed	Snow	Rain
Pollution	1.00	0.18	-0.09	-0.06	0.19	-0.24	0.02	-0.05
Dew point	0.18	1.00	0.82	-0.74	0.23	-0.30	-0.03	0.13
Temperature	-0.09	0.82	1.00	-0.79	0.18	-0.15	-0.09	0.05
Pressure	-0.06	-0.74	-0.79	1.00	-0.16	0.18	0.07	-0.07
Wind direction	0.19	0.23	0.18	-0.16	1.00	-0.20	0.01	-0.05
Wind speed	-0.24	-0.30	-0.15	0.18	-0.20	1.00	0.02	-0.01
Snow	0.02	-0.03	-0.09	0.07	0.01	0.02	1.00	-0.01
Rain	-0.05	0.13	0.05	-0.07	-0.05	-0.01	-0.01	1.00

TABLE 2. Model performance with different meteorological data input.

Features	RMSE	MAE	SMAPE
8 (All)	21.5657	15.8092	0.3089
6 (Without snow and rain)	20.4836	15.4214	0.3123
5 (Pollution, dew point, wind direction, wind speed, temperature)	17.2372	13.3568	0.2914
4 (Pollution, dew point, wind direction, wind speed)	19.6121	14.2006	0.2987

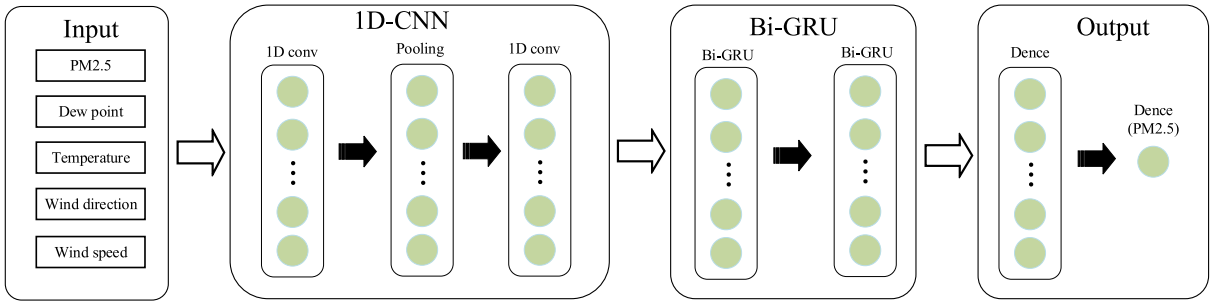


FIGURE 2. Structure of CBGRU model for PM2.5 forecasting.

When $0< r <1$, there is a positive correlation, and if $-1 < r < 0$, there is a negative correlation. The absolute of r is closer to 1, the gap between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is smaller and the correlation is greater.

For Beijing PM2.5 dataset, correlation coefficient between each feature and PM2.5 concentration was calculated respectively. As shown in Table 1, dew point, wind direction and snowfall are positive correlation with PM2.5, while temperature, air pressure, wind speed and rainfall are negative correlation with PM2.5 concentration. It is found that all the meteorological variables are weakly correlated with each other, which indicates that there is no information duplication between the meteorological variables and they can be directly used as the input of the prediction model.

In order to select the proper input variables of forecasting model, corresponding experiments were performed. As shown in Table 2, we constructed the prediction model by gradually reducing the input variables (Detailed modeling is described in Section IV). The error results of different models are compared by three measures (The smaller the error measure value, the better the prediction effect of the model), they are root mean square error (RMSE), mean absolute error (MAE) and symmetric mean absolute percent error (SMAPE). It is found that the model performed better with inputs of pollution, dew point, wind direction, wind speed and temperature than with inputs of only the first four. But when the pressure is increased as input, the performance of the model begins to decline, and the performance is even worse when all the weather factors in the data set are taken as inputs. This phenomenon is consistent with what is shown in Table 1, the correlation coefficients of air pressure, snowfall and rainfall are quite small, unrelated inputs increase the model's complexity and the difficulty of learning useful features. So, dew point, historical PM2.5, temperature, wind direction and wind speed are selected as the input variables of the forecasting model.

III. METHODOLODIES

A. CBGRU MODEL FOR PM2.5 FORECASTING

The historical meteorological data and PM2.5 concentration data are used as model inputs, the future PM2.5 concentration is used as output to perform multi-step prediction. Figure 2 shows the structure of forecasting model.

The model consists of three parts. In the first part, the one-dimensional convolutional neural networks (convnets) performs local feature learning and dimensionality reduction

on five input variables, the original data is processed by convolution and pooling to form low-dimensional feature sequences. Second, the feature sequences is fed into the bidirectional GRU neural networks, which reset gate and update gate constantly adjust their parameters in a large amount of training, so that it can learn the time dependence relationship between the information extracted from the convolutional neural networks. At the end of the model, the fully connected layers is stacked, the last layer contains only one neuron without any activation function, generating the predicted value of the PM2.5 concentration. Theoretically, the innovation of this method is the combination of the local feature extraction ability and lightness of convnets with the time series prediction ability of GRU by using 1D convnet as a preprocessing step before a GRU. On the other hand, by processing a sequence both way, a bidirectional GRU is able to catch patterns that may have been overlooked by a one-direction GRU.

B. 1D CONVNETS FOR LOCAL TREND FEATURES LEARNING

The 1D convnets is used for local trend features learning. Convnets can perform convolution operation, extracting features from local input patches, allowing for representation modularity and data efficiency. These properties make convnets not only excellent in computer vision, but also suitable for sequence processing [14]. In this forecasting case, time can be treated as a spatial dimension just like the height or width of a two-dimensional image. The local perception and weight sharing feature of convnets can reduce the number of parameters for processing multivariate time series, thereby improving learning efficiency. With the peculiarity of temporal translation invariants [15], a pattern learned at a certain position in a sequence can be identified at other locations later, because the same input transformation is performed for each subsequence.

As shown in the Figure 3, using a convolution window in each convolutional layer to process the meteorological and PM2.5 time series, it is possible to learn sequence fragments within a window size, and should be able to identify these subsequences anywhere in the entire time series, so that the local trend change features of the multivariate time series over time can be captured. After the 1D convolution operation, the max pooling operation should be used for subsampling, which outputs the maximum value of subsequences extracted from the input time series. In this way, the length of one-dimensional input time series is reduced.

C. BIDIRECTIONAL GATED RECURRENT UNIT FOR TIME SERIES FORECASTING

In this paper, bidirectional Gated Recurrent Unit (GRU) is used for processing prediction as shown in the Figure 4. As everyone knows, RNN is a special neural network developed for processing sequence data. But there are some drawbacks with simple RNN, like the vanishing gradient and exploding gradient, which makes it difficult for RNN to learn the long-term dependencies tasks. To solve these problems,

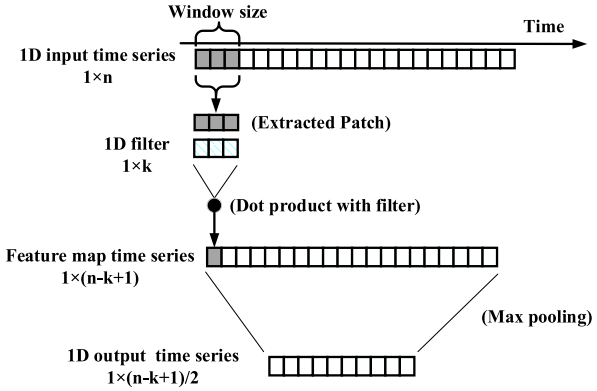


FIGURE 3. Graphical illustration of the 1D convnets processing time series.

a custom RNN structure, i.e., LSTM and GRU, is developed. The former can track long-term information via the gates it contains (input gate, forget gate and output gate) [16]. The latter is an improved version of the LSTM, which can also learn long-term dependencies [17]. Unlike LSTM, GRU has no memory unit and has 2 gates (update gate and reset gate) instead of 3 gates, having a simpler architecture requires less computation and can be trained faster. Although the structure of GRU is not so complicated, the research shows that its performance is comparable to LSTM [18].

The graphical illustration of GRU neural networks is included in Figure 4. Inside a GRU, the update gate (z) specifies which information can be retained to the next state, and the reset gate (r) specifies how the previous state information is combined with the new input information. The calculation formula for the next output and state value in the GRU unit is as follows:

$$z_t = \sigma(W_z * [x(t), h(t-1)]) \tag{3}$$

$$r_t = \sigma(W_r * [x(t), h(t-1)]) \tag{4}$$

$$\hat{h}(t) = \sigma(W_h * [x(t), (r_t * h(t-1))]) \tag{5}$$

$$h(t) = (1 - z_t) * h(t-1) + z_t * \hat{h}(t) \tag{6}$$

where σ is the activation function, $x(t)$ is the input, $h(t-1)$ is the previous output, w_z, w_r and w_h are the weights of the update gate, reset gate, and candidate output, respectively.

The bidirectional GRU consists of two ordinary GRUs, which process the input sequence from two directions of time series (chronologically and antichronologically), then merge their representations together. Factors such as air quality and meteorological are subject to a continuous function, we can fit a function according to the historical observation values (time series) through the observation values to predict the future values. In the same way, future data can be used to fit a function to predict the value of the previous moment. For time series forecasting tasks, we know that only historical data can provide predictive power when making predictions, but this method of bidirectional training model can provide more useful information in modeling. By viewing meteorological and PM2.5 data from two directions enables the model to get richer representations and capture patterns that may be

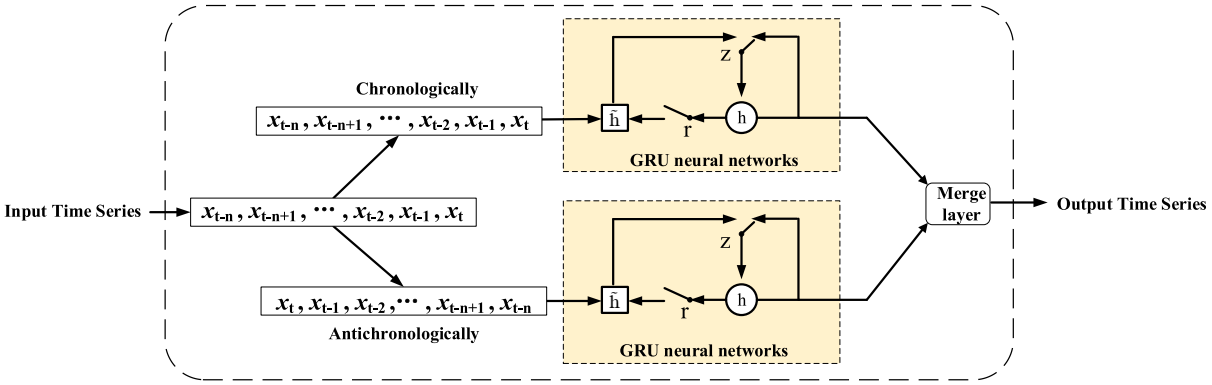


FIGURE 4. Bidirectional GRU processing time series. *r* and *z* are the reset and update gate, *h* an $\tilde{h}$ are the activation and the candidate activation of GRU neural networks.

ignored when using one-direction GRU, thereby improving the performance of ordinary GRU.

IV. CASE STUDY

The real air quality data set described in Section II is used to evaluate the proposed model, which performance is compare with the other seven models. All deep models are trained on Keras framework with TensorFlow backend, while traditional machine learning methods are implemented through the scikit-learn library. All recurrent architecture are trained using backpropagation through time (BPTT) with *RMSprop* as an optimizer.

A. ERROR MEASURES

Loss function is defined by mean absolute error (MAE), MAE can better reflect the actual situation of the prediction error, backpropagation operation based on MAE value in each mini-batch during training. At the same time, root mean square error (RMSE) and symmetric mean absolute percentage error (SMAPE) are selected as the error evaluation metrics of the model, which can evaluate the degree of change and accuracy of data, measuring the prediction quality of model. The calculation formula is as shown in equation (7), (8) and (9).

$$MAE_{(y',y)} = \frac{1}{n}\sum_{i=1}^{n}|y'_i - y_i| \tag{7}$$

$$RMSE_{(y',y)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y'_i - y_i)^2} \tag{8}$$

$$SMAPE_{(y',y)} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y'_i - y_i|}{(y'_i + y_i)/2} \tag{9}$$

where n is the total number of samples, y_i is the measured time series, and y'_i is the predicted time series.

B. EXPERIMENTAL SETUP

Seven reference models were built to evaluate the performance of the proposed model, i.e., support vector regression (SVR), gradient boosting regressor (GBR), decision tree regressor (DTR), simple RNN, LSTM, GRU and bidirectional GRU (BGRU). The training is carried out in mini-batches with the batch size of 50, and all the models are trained for 100 epochs. In order to avoid the overfitting problem, *Dropout* is widely used between layers with the probability of 0.2. If the loss of the past epoch is greater than that of the current epoch, the weight matrices are stored. Furthmore, all models used an early stopping condition during the training, which stops the training if the validation loss on the validation data does not change within 10 training epochs. *RMSprop,*a variant of stochastic gradient descent (SGD), is chosen as the optimizer of these models, as it is usually a good choice for recurrent neural networks, which taking into account previous weight updates when computing the next weight update, rather than just looking at the current value of the gradients. Furthermore, *Momentum* of RMSprop addresses two issues with SGD: convergence speed, and local minima. After obtaining the trained models, each data points in testing set are tested and, MAE, RMSE and SMAPE are calculated.

In order to achieve the best prediction performance, several hyperparameters should be preset before building the CBGRU prediction model. In order to prove the superiority of CBGRU model proposed in this paper, GRU networks was selected as the benchmark. CBGRU model based on the structure of benchmark was established after the limit of GRU prediction ability was reached. Mainly examined parameters are lookback and number of neurons, where the lookback specifies how many timesteps back should the input data go, the number of neurons specifies which neuron nodes achieve an optimal prediction effect.

First of all, the number of neurons was set to an equivalent value chosen from a candidate set of {32, 64, 80, 128, 256}. Several experiments were performed and the corresponding errors (MAE and RMSE calculated by standardized data) were recorded as shown in Table 3. The results show that with the increase of neurons of GRU hidden layer, the forecasting performance first improves greatly and then begins to deteriorate. Under the same configuration, over-fitting problems arise when neurons exceed 80. Thus, we set the number of neurons to 80 in the successive experiments.

TABLE 3. Effect of the number of neuron nodes in GRU model.

neurons	RMSE(standardized)	MAE(standardized)
32	0.3615	0.2517
64	0.3617	0.2522
80	**0.3330**	**0.2063**
128	0.3437	0.2231
256	0.3504	0.2517

TABLE 4. Effect of lookback.

lookback	RMSE	MAE
4	19.1396	15.9706
6	17.7966	15.3451
8	**17.2372**	**13.3568**
10	17.2869	14.1488
12	18.7551	14.8502
16	18.8912	15.7737

Next, making neurons as a constant, changing the lookback, we can see from Table 4 that the model is best fitted when the lookback is 8, as indicated by the RMSE and MAE. That is to say, a small lookback cannot guarantee evenough long-term memory inputs for this deep learning model, but large lookback allows for more redundant information inputs, which is not conducive to modeling. Furthermore, the temporal correlations among the PM2.5 concentration time series were analyzed by autocorrelation functions. For time delay k, the autocorrelation coefficients can be calculated as follows:

$$\rho_k = \frac{Cov(y_t, y_{t+k})}{\sigma_{y_t}\sigma_{y_{t+k}}} \tag{10}$$

where y_t and y_{t+k} denote the PM2.5 concentrations at time t and time $t+k$, respectively, $Cov(\cdot)$ is the covariance and $\sigma(\cdot)$ is the standard deviation. The results are shown in Figure 5. An obvious descending trend is observed with increasing time lag, which means earlier events have a weaker effect on the current status. Bisides, the autocorrelation coefficients is higher than 0.7 when the time lag is less than 7, indicating a high temporal correlation. As a compromise, the lookback was set to 8, which was the most appropriate setting for this forecasting model.

After a lot of experiments, the values of each parameter are determined. Both the meteorological data and the PM2.5 data of the past 8 hours are used to predict the PM2.5 concentration 2 hours later. For fairness, all reference deep learning models in this experiment used the same hidden layers and the number of neurons, the difference between these models and the CBGRU is the absence of convolutional neural networks. For CBGRU model, after adjusting the parameters of different model structures and parameters, the final parameters are as follows.

- Convolutional neural networks: Contains 2 layers of convolutional layers with the activation functions of ReLu, each have 40 and 80 feature detectors, the length of the 1D convolution window is 3. There is a MaxPooling1D layer between the two convolution layers with the pool size of 2, which halve the input tensor.
- Bidirectional GRU networks: Contains 2 layers of bidirectional GRUs with 80 neurons per layer.
- Fully connected layers: contains 1 fully connected layer with only 1 neuron.

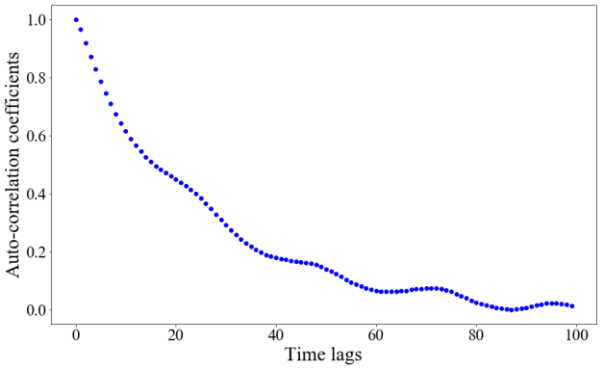


FIGURE 5. Variations among the autocorrelation coefficients of PM2.5 concentration with respect to different time lags.

TABLE 5. Comparison of model performance.

Method	Parameter setting	RMSE	MAE	SMAPE
SVR	RBF kernel, C=32, gamma=0.05515674	27.7597	16.7203	0.2610
DTR	Criterion = MAE, maximum depth of the tree is 10	29.1560	17.5150	0.2624
GBR	Loss function is least squares regression	27.6418	16.9185	0.2595
RNN	Hidden layers = 2(each layer node is 80)	20.9359	16.3941	0.3331
GRU	Hidden layers = 2(each layer node is 80)	17.2372	13.3568	0.2914
LSTM	Hidden layers = 2(each layer node is 80)	17.3050	12.5859	0.2879
BGRU	Hidden layers = 2(each layer node is 80)	15.6291	12.4655	0.2861
CBGRU	See the fifth paragraph of Section IV.B	**14.5319**	**10.4798**	**0.2055**

C. FORECASTING RESULTS AND ANALYSIS

After training to convergence, the optimal model weights of CBGRU prediction model is obtained. The evaluations were conducted using the test set (Hourly points between May 6, 2014 and December 30, 2014), and the predicted and observed PM2.5 concentrations are presented in Figure 6. It can be observed from the figure that the CBGRU model produced results which can follow the fluctuations of actual values during the testing set successfully.

To verify the efficiency and accuracy of the proposed approach, several comparative models were developed for PM2.5 prediction.

Table 5 lists the quantitative results by RMSE, MAE and SMAPE, which gives comparative analysis of SVR, DTR, GBR, RNN, GRU, LSTM, BGRU and our proposed model of CBGRU. As shown in the Table 5, shallow machine learning

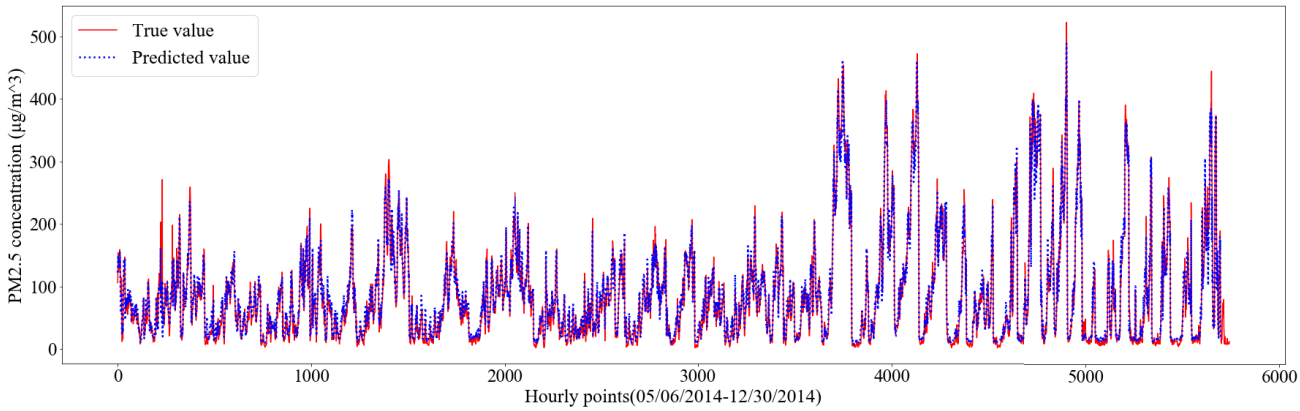


FIGURE 6. PM2.5 concentration forecasting results of CBGRU model.

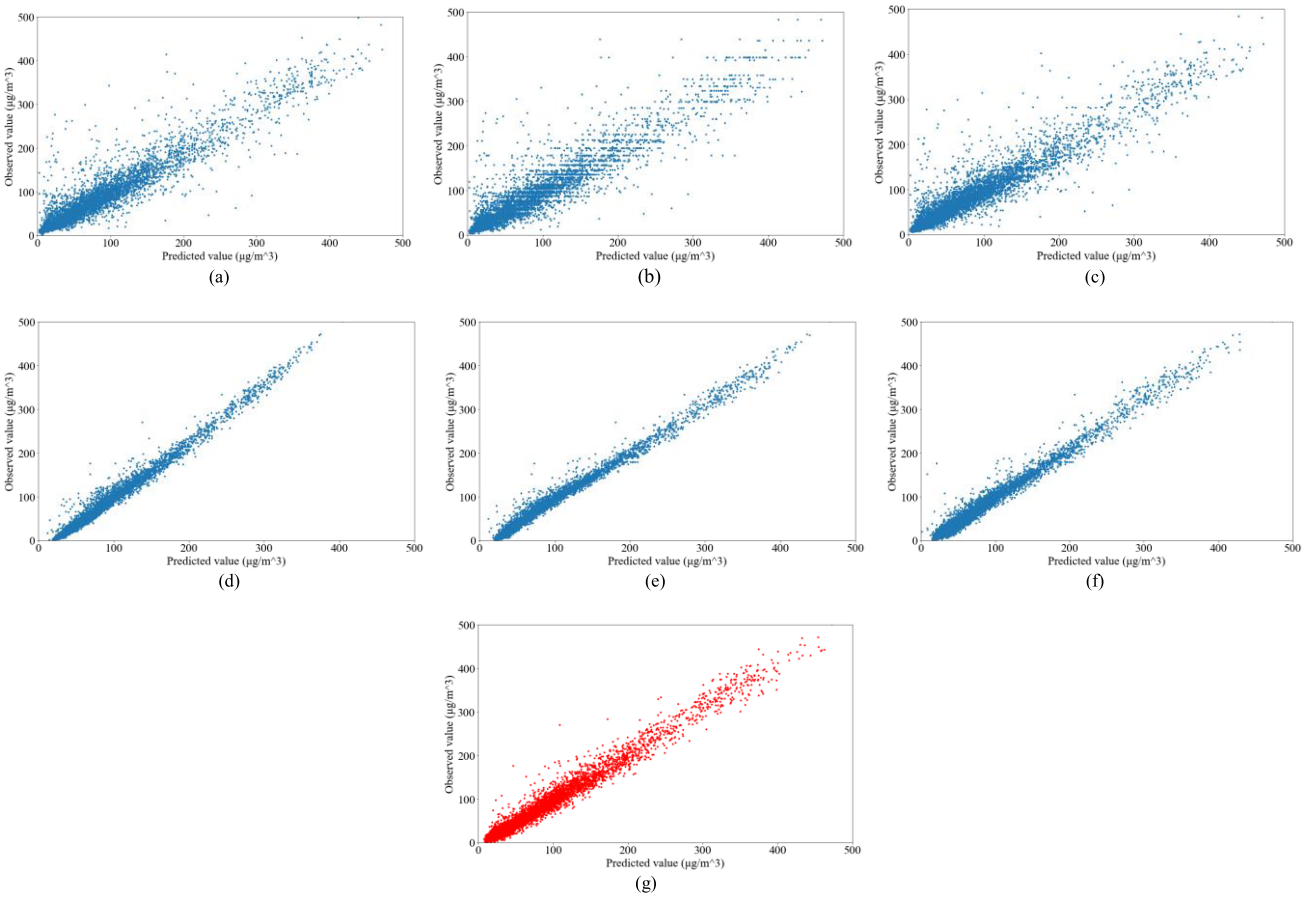


FIGURE 7. Scatter plots with the comparison models. (a) DTR, (b) SVR, (c) GBR, (d) RNN, (e) GRU, (f) LSTM, and (g) CBGRU.

models (SVR, DTR and GBR) have similar performance. Compared with traditional deep learning methods (RNN, GRU, LSTM), shallow machine learning methods have larger RMSE and MAE, while the SMAPE are smaller. For deep learning methods, LSTM and GRU have similar performance, both of them are significantly superior to RNN. Furthermore, the model error of BGRU is lower than GRU, which shows the bi-direction training model is improved obviously compared with the traditional model, indicating that the bidirectional model can improve the prediction performance. More significantly, compared to other seven methods, our model exhibited higher forecasting precision, as indicated by the RMSE, MAE and SMAPE values. This result confirms that our model CBGRU can learn local trend information and

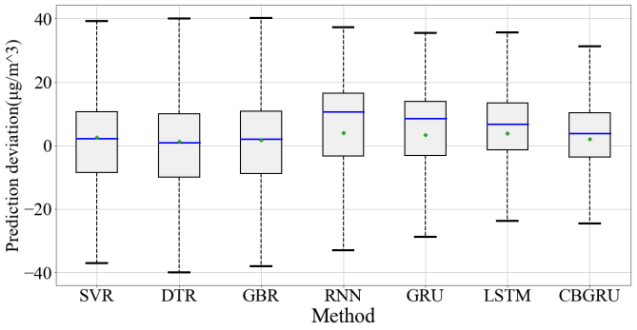


FIGURE 8. Boxplot of comparison models' prediction deviation. The blue solid line in the box represents the median of data, and the green diamond represents the mean of data.

long term dependencies features of meteorological data and PM2.5 concentration data.

In order to compare the prediction effect of each model more intuitively, the scatter plots of observed and predictid PM2.5 concentrations during the whole test set is illustrated in Figure 7. It can be seen from the figure (a, b and c) that the distribution between predicted and observed values of shallow machine learning models (SVR, DTR, GBR) is divergent. For deep learning methods, RNN (Figure 7 (d)) showed the worst forecasting effect, it fail at some peak and valley values, causing the distribution between the forecasted and observed values deviate from the diagonal. Apparently, variants of RNN (LSTM and GRU, Figure 7 (e and f)) show better results. Compared with all the models described above, it can be find that the model proposed in this paper (Figure 7(g)) is more sensitive to local sharp changes (the distribution between the predicted value and the observed value is more inclined to the diagonal), which mainly attributed to the existence of convolution networks that capture richer local change information.

Besides, the prediction deviation analysis is also conducted. The prediction deviation is obtained by subtracting the observed values from the predicted values of each model. The boxplot of the prediction deviation is shown in Figure 8. The height of the box partly reflects the fluctuation of the deviation data, the flatter the box, the more centralized the data is. Similarly, the shorter the whisker, the more centralized the data is. According to Figure 8, although the mean and median of SVR, DTR and GBR are closer to 0, they are highly volatile. With narrower box and whisker, the CBGRU performs much better compared to other models.

In terms of the comparison analysis above, the proposed method outperforms all other models, including mainstream approaches like LSTM. It fully proves the effectiveness and superiority of the combination of 1D convnets and bidirectional GRU.

V. CONCLUSION

In this study, time series forecasting experiments on PM2.5 concentration using 1D convnets and bidirectional GRU, which is a special type of RNN are conducted. The performances of the proposed model have been investigated. The results are compared with traditional mechine learning models and conventional deep learning models. The results show that the proposed method can be suitable and competitive on the PM2.5 data time series forecasting. To be more specific, compared with shallow machine learning models, such as DTR, SVR and GBR, deep learning-based methods exhibited better prediction performance. Furthermore, compared with GRU, bidirectional GRU has lower error value, which indicates that the use of bidirectional GRU can improve the prediction effect. This is because the bidirectional GRU processes the time series chronologically and antichronologically, it captures patterns that may be ignored by one-direction GRUs, improving feature learning capabilities in time series. In addition, compared with the other benchmark models, the accuracy of the CBGRU model is significantly improved, which shows that the convnets can help the GRU to obtain better prediction performance, because convnets uses its local feature learning ability and subsampling ability to obtain a sequence pattern that is more conducive to GRU processing.

REFERENCES

[1] Y. Zhang, Y. He, and J. Zhu, "Research on forecasting problem based on multiple linear regression model PM2.5," *J. Anhui Sci. Technol. Univ.*, vol. 30, no. 3, pp. 92–97, 2016.

[2] K. R. Baker and K. M. Foley, "A nonlinear regression model estimating single source concentrations of primary and secondarily formed $PM_{2.5}$," *Atmos. Environ.*, vol. 45, no. 22, pp. 3758–3767, 2011.

[3] J. B. Ordieresa, E. P. Vergara, R. S. Capuz, and R. E. Salazar, "Neural network prediction model for fine particulate matter ($PM_{2.5}$) on the US–Mexico border in El Paso (Texas) and Ciudad Juárez (Chihuahua)," *Environ. Model. Softw.*, vol. 20, no. 5, pp. 547–559, 2005.

[4] Z. Wang and Z. Long, "$PM_{2.5}$ prediction based on neural network," in *Proc. 11th Int. Conf. Intell. Comput. Technol. Automat. (ICICTA)*, Changsha, China, Sep. 2018, pp. 44–47.

[5] R. Chuentawat and Y. Kan-Ngan, "The comparison of $PM_{2.5}$ forecasting methods in the form of multivariate and univariate time series based on support vector machine and genetic algorithm," in *Proc. 15th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol. (ECTI-CON)*, Chiang Rai, Thailand, Jul. 2018, pp. 572–575.

[6] M. A. Elangasinghe, N. Singhal, and K. N. Dirks, "Complex time series analysis of PM_{10} and $PM_{2.5}$ for a coastal site using artificial neural network modelling and k-means clustering," *Atmos. Environ.*, vol. 94, pp. 106–116, Sep. 2014.

[7] A. G. Salman, Y. Heryadi, E. Abdurahman, and W. Suparta, "Single layer & multi-layer long short-term memory (LSTM) model with intermediate variables for weather forecasting," *Procedia Comput. Sci.*, vol. 135, pp. 89–98, Jan. 2018.

[8] Y. Tsai, Y. Zeng, and Y. Chang, "Air pollution forecasting using RNN with LSTM," in *Proc. IEEE 16th Int. Conf. Dependable, Autonomic Secure Comput., 16th Int. Conf. Pervasive Intell. Comput., 4th Int. Conf. Big Data Intell. Comput. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech)*, Athens, Greece, 2018, pp. 1074–1079.

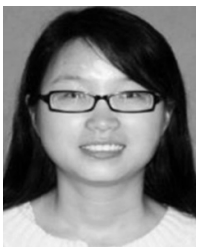
[9] J. Xie, "Deep neural network for $PM_{2.5}$ pollution forecasting based on manifold learning," in *Proc. Int. Conf. Sens., Diag., Prognostics, Control (SDPC)*, Shanghai, China, Aug. 2017, pp. 236–240.

[10] B. Wang, Z. Yan, H. Luo, T. Li, J. Lu, and G. Zhang, "Deep uncertainty learning: A machine learning approach for weather forecasting," 2018, [Online]. Available: https://arxiv.org/abs/1812.09467v2

[11] L. Yan, Y. Wu, L. Yan, and M. Zhou, "Encoder-decoder model for forecast of $PM_{2.5}$ concentration per hour," in *Proc. 1st Int. Cogn. Cities Conf. (IC3)*, Okinawa, Japan, Aug. 2018, pp. 45–50.

[12] V. Athira, P. Geetha, R. Vinayakumar, and K. P. Soman, "DeepAirNet: Applying recurrent networks for air quality prediction," *Procedia Comput. Sci.*, vol. 132, pp. 1394–1403, Dec. 2018.
[13] X. Liang, T. Zou, B. Guo, S. Li, H. Zhang, S. Zhang, H. Huang, and S. X. Chen, "Assessing Beijing's $PM_{2.5}$ pollution: Severity, weather impact, APEC and winter heating," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 471, no. 2182, 2015, Art. no. 20150257.
[14] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. IJCAI*, 2015, pp. 3995–4001.
[15] F. Chollet, *Deep Learning With Python*. New York, NY, USA: Manning Publications, 2017, pp. 208–209.
[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[17] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473v1*. [Online]. Available: https://arxiv.org/abs/1409.0473v1
[18] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: https://arxiv.org/abs/1412.3555

QING TAO received the B.S. degree from the School of Mechanical, Electrical, and Information Engineering, Shandong University, Weihai, China, in 2018. Currently, he is pursuing the M.S. degree with the School of Automation, Central South University, Changsha, China. His research interests include deep learning, artificial neural network, and time series forecasting.



FANG LIU was born in Jiangxi, China, in 1982. She received the Ph.D. degree from Waseda University, Japan, in 2011, and the B.S. degrees from the College of Electric and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou, China, in 2005. She is a Professor with the School of Information Science and Engineering, Central South University, Changsha, China, from 2017. Her main research interests include stability analysis of time-delay system and power system, and robust control of FACTS with wide-area signals.

YONG LI (S'09-M'12-SM'14) was born in Henan, China, in 1982. He received the B.Sc. and Ph.D. degrees from the College of Electrical and Information Engineering, Hunan University (HNU), Changsha, China, in 2004 and 2011, respectively, and the second Ph.D. degree from TU Dortmund University, Dortmund, Germany, in June 2012, all in electrical engineering. Since 2009, he has been a Research Associate with the Institute of Energy Systems, Energy Efficiency, and Energy Economics, TU Dortmund University. Since 2014, he has been a Full Professor of Electrical Engineering with HNU. His research interests include ac/dc energy conversion systems, analysis and control of power quality, and HVDC and FACTS technologies.

DENIS SIDOROV (M'08–SM'18) was born on October 30, 1974, in Irkutsk, Russia. He received the DSc (Habilitation) degree in applied mathematics, in 2014, and a Professor of Russian Academy of Sciences, in 2018. He is a Leading Researcher with the Energy Systems, Institute of Russian Academy of Sciences. He defended his Ph.D. thesis "Modeling of nonlinear dynamic systems with Volterra series: theory and applications," in 1999. He was with the Department of Electronic and Electrical Engineering, Trinity College Dublin (Ireland), with CNRS (Compiegne, France) as a Research Fellow, and with ASTI Holding (Singapore) as Vision Engineer involved in different DSP and NDT projects from 2001–2007. His research interests include integral and differential equations, machine learning, wind energy, and inverse problems. He has authored more than 140 scientific papers and 3 monographs.