



Article

A Deep CNN-LSTM Model for Particulate Matter (PM_{2.5}) Forecasting in Smart Cities

Chiou-Jye Huang ¹  and Ping-Huan Kuo ^{2,*} 

¹ School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou 341000, China; chioujye@163.com

² Computer and Intelligent Robot Program for Bachelor Degree, National Pingtung University, Pingtung 90004, Taiwan

* Correspondence: phkuo@mail.nptu.edu.tw; Tel.: +886-8-7663800 (ext. 32620)

Received: 31 May 2018; Accepted: 8 July 2018; Published: 10 July 2018



Abstract: In modern society, air pollution is an important topic as this pollution exerts a critically bad influence on human health and the environment. Among air pollutants, Particulate Matter (PM_{2.5}) consists of suspended particles with a diameter equal to or less than 2.5 μm. Sources of PM_{2.5} can be coal-fired power generation, smoke, or dusts. These suspended particles in the air can damage the respiratory and cardiovascular systems of the human body, which may further lead to other diseases such as asthma, lung cancer, or cardiovascular diseases. To monitor and estimate the PM_{2.5} concentration, Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) are combined and applied to the PM_{2.5} forecasting system. To compare the overall performance of each algorithm, four measurement indexes, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Pearson correlation coefficient and Index of Agreement (IA) are applied to the experiments in this paper. Compared with other machine learning methods, the experimental results showed that the forecasting accuracy of the proposed CNN-LSTM model (APNet) is verified to be the highest in this paper. For the CNN-LSTM model, its feasibility and practicability to forecast the PM_{2.5} concentration are also verified in this paper. The main contribution of this paper is to develop a deep neural network model that integrates the CNN and LSTM architectures, and through historical data such as cumulated hours of rain, cumulated wind speed and PM_{2.5} concentration. In the future, this study can also be applied to the prevention and control of PM_{2.5}.

Keywords: PM_{2.5} forecasting; deep learning; big data analytics; CNN-LSTM model

1. Introduction

As the International Energy Agency (IEA) [1] had pointed out, air pollution causes the premature death of 6.5 million people every year [2], and thus far, energy production and utilization are the largest man-made air pollution sources. Air pollution abatement technology has become a part of public knowledge, and clean air is extremely important to ensure human health. Although people have an increasing recognition as to its urgency, air pollution problems are still unsolved in many countries, and global health risks will be extended further in future decades [2]. Among pollution sources, suspended particles with a diameter equal to or less than 2.5 μm are called PM_{2.5}. As the particles of this pollution source are small, they can penetrate the alveoli, and even pass through the lungs and affects other organs of the body [3].

In some major cities of the world (e.g., New York, Los Angeles, Beijing, and Taipei), air pollution has been identified as one of the main health hazards [3]. The air pollution in big cities also negatively impacts the environment around the city. One reference [4] pointed out that high PM_{2.5} concentration has even been detected in regions such as the East China Plain, Sichuan Province, and the Taklimakan

desert. Studies about the relationship between $PM_{2.5}$ and mortality in US cities had also been discussed in detail by Kioumourtzoglou et al. [5]. Thus, for urban residents, solving $PM_{2.5}$ air pollution is a critically urgent and important topic. Although Walsh [6] pointed out that the main $PM_{2.5}$ pollution source for the major cities in China is motor vehicles, there are a large number of sources of air pollution, and the degree of air pollution is also related to weather and wind direction. Therefore, the management and control of city air pollution is rather complex.

For a smart city, to create a smarter environment and improve the quality of its citizens' lives, it is indispensable to equip the city with the functions of sensing the weather and the surrounding environment. Liu et al. [7] put forward an idea to establish a smart urban sensing system architecture using Internet of Things (IoTs) which is equipped with sensing and monitoring systems for $PM_{2.5}$, temperature, and noise. This system can efficiently monitor the condition of air pollution and other environmental pollution of the city and collect data for analysis and strategy evaluation. Zhang et al. [8] used IoTs technology and combined information such as social media, air quality, taxi trajectory, and traffic conditions, and integrated machine learning technology, which has become one valuable application in smart cities. Zeng and Xiang [9] also proposed an air pollution sensing and monitoring system applied to smart cities. This system adopts a Q-learning algorithm to realize the computation of an adaptive sampling scheme. Apart from these studies, there are also other studies about developing various sensors to investigate air pollution. For instance, Ghaffari et al. [10] puts forward a nitrate sensor whose sensitivity and accuracy had been well verified in experiments.

Since the topic of $PM_{2.5}$ air pollution has received increasing attention, there is currently much relevant analysis and many studies about $PM_{2.5}$. Lary and Sattler [11] proposed a method to estimate the $PM_{2.5}$ concentration using machine learning. This method collected the air pollution indexes from 55 countries from 1997 to 2014, and used a machine learning method for modeling. It produced decent results, but it could only carry out tiny estimation among intervals, and could not forecast future $PM_{2.5}$ conditions. In 2016, Li et al. adopted the Stacked Autoencoder (SAE) architecture to forecast the $PM_{2.5}$ concentration of various regions [12]. Although SAE requires the pre-training step and cannot perform training directly, its performance is good. In 2017, the latest research by Li et al. showed that the air pollution estimation system based on a Long Short-Term Memory (LSTM) neural network is more accurate [13]. Therefore, the application of LSTM to the research topic of air pollution is a good approach. Additionally, Yu et al. [14] uses an Eta-Community Multiscale Air Quality (Eta-CMAQ) forecasting model to forecast the air pollution index of $PM_{2.5}$. This method can perform the $PM_{2.5}$ forecasting according to the chemical composition of $PM_{2.5}$, such as Organic Carbon (OC) or Elemental Carbon (EC). This approach belongs to the traditional $PM_{2.5}$ forecasting algorithm, and its result is effective and feasible. However, it cannot perform comprehensive forecasting according to weather information (e.g., wind speed, rainfall, etc.).

The particles and molecules generate a light scattering phenomenon under illumination, and at the same time absorb partial energy of the illumination. When a collimated monochromatic light is projected on a measured particle field, it is affected by the light scattering and absorption around the particles, and the light intensity is attenuated. This way, the relative attenuation ratio of the light projected through the concentration field can be measured and obtained. Therefore, the relative attenuation rate can fundamentally reflect the linearity of the relative concentration of dust in the pending field. The intensity of light is proportional to the strength of the electrical signal of the optical to electrical conversion, by measuring the electrical signal, the relative attenuation rate can be obtained, and the concentration of dust in the field to be measured can be determined [15,16]. Furthermore, using geospatial assessment tools [17] or more complex statistic algorithms [18,19] are also feasible and practical for the forecasting of $PM_{2.5}$ pollution issue.

Summing up, $PM_{2.5}$ forecasting is absolutely a vital topic for the development of smart cities. In this paper, a deep learning model based on the main architectures of CNN and LSTM is proposed to forecast future $PM_{2.5}$ concentration. This architecture can conduct the forecast of the future $PM_{2.5}$ concentration according to the past $PM_{2.5}$ concentration and even other weather conditions. To compare

the overall efficiency of each algorithm, two measurement indexes, MAE and RMSE are also applied to the experiments in this paper. In addition, other traditional machine learning algorithms are compared. The performance of all algorithms is also graded and verified in each experiment. As for the aspect of database selection, a $PM_{2.5}$ dataset of Beijing is used. Aimed at the problems in smart cities that urgently need to be solved, $PM_{2.5}$ forecasting is integrated into the air pollution forecasting system of the smart city, thus achieving the prospect of creating a better and smarter city.

The major contributions of this paper are: (1) designing a high precision $PM_{2.5}$ forecasting algorithm; (2) comparing the performances of the several popular machine learning methods in the air pollution forecasting problem; and (3) validating the practicality and feasibility of the proposed network in $PM_{2.5}$ forecasting application.

This paper is organized as follows. The $PM_{2.5}$ monitoring and forecasting in smart cities is described in Section 2; the background knowledge of the artificial neural network is presented in Section 3; the design of the proposed APNet is illustrated in Section 4; the forecasting and comparison results are demonstrated in Section 5; and conclusions are given in Section 6.

2. $PM_{2.5}$ Monitoring and Forecasting in Smart Cities

The $PM_{2.5}$ source analyses of two major cities, Beijing and Shanghai, are shown in Figure 1 [20]. As shown, in Beijing, the biggest $PM_{2.5}$ pollution source comes from transboundary pollution (25%), and the second biggest source is motor vehicles (22%); while in Shanghai, the biggest $PM_{2.5}$ pollution source comes from motor vehicles (25%), and the second biggest source is pollution from other provinces (20%). This indicates that $PM_{2.5}$ pollution caused by vehicles has a great effect on urban air pollution. As the air pollution condition can be changed to some degree by the wind direction, pollution sources from other regions is another one of the main reasons. Additionally, there are still many other factors that cause $PM_{2.5}$ pollution, such as coal combustion, road dust, industrial Volatile Organic Compound (VOC), biomass burning, and combustion installations. All of these can affect the overall $PM_{2.5}$ concentration of a city. Therefore, the tracking and forecasting of $PM_{2.5}$ concentration is a challenging and important topic in smart cities.

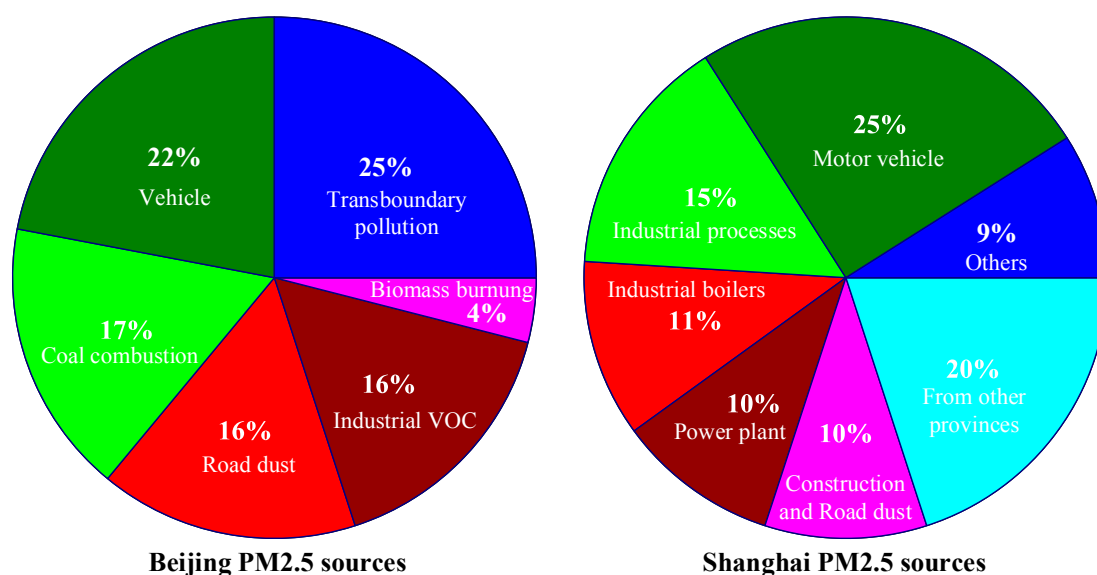


Figure 1. The Particulate Matter ($PM_{2.5}$) source pie charts of Beijing and Shanghai [20].

To effectively monitor and forecast the $PM_{2.5}$ concentration in smart cities, an urban sensing application in big data analysis is set up whose architecture is shown in Figure 2. First, various sensors can be installed at various corners in the city, such as $PM_{2.5}$ sensors and meteorological sensors to sense the urban weather conditions and degree of air pollution. Next, to monitor each index effectively,

Internet of Things (IoTs) can be used to transfer the information and data to the monitoring servers for performing long-term data monitoring and tracking. However, for a smart city, merely monitoring the collected data above is insufficient since the large amount of collected data are a valuable resource. Therefore, relevant big data analysis techniques can be used to analyze and track the various data so as to reach the goal of effectively monitoring, managing, and maintaining citizens' health. In this paper, the proposed CNN-LSTM is an advanced algorithm which adopts artificial intelligence and big data, and combines various data indexes to accurately forecast the future $PM_{2.5}$ concentration. The detailed algorithm architecture is introduced in the following sections.

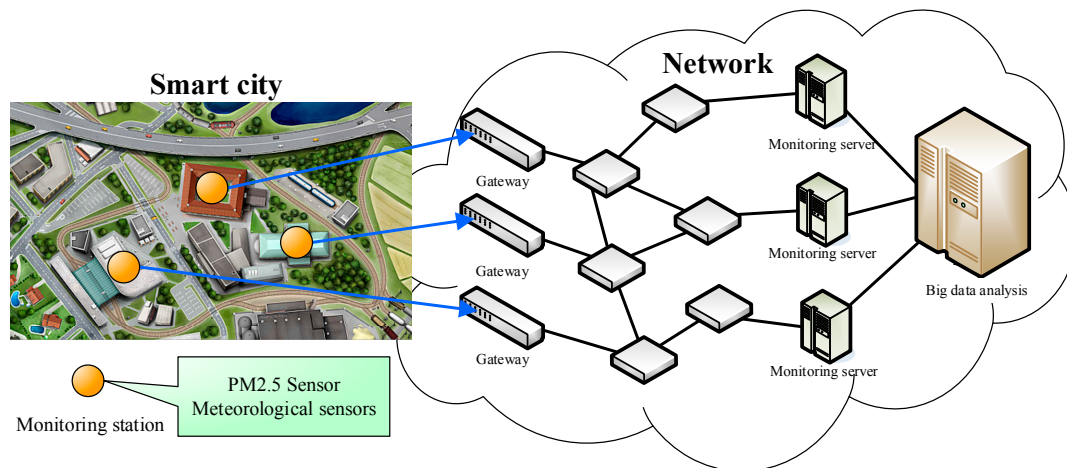


Figure 2. Urban sensing application in big data analysis; meteorological parameters: Temperature, Relative Humidity, and Precipitations.

3. The Background Knowledge of the Artificial Neural Network

An Artificial Neural Network (ANN) is a kind of mathematic model that imitates the operation of biological neuron. It is a strong, non-linear modeling tool. An earlier ANN architecture is Multilayer Perceptron (MLP) [21], a neural network with a fully-connected architecture. Basically, MLP already has a good performance, and has been applied widely. However, if the data complexity is high, the MLP architecture alone may fail to learn all the conditions effectively. At present, many new architectures have been developed for ANN. In this paper, the main architectures are Convolutional Neural Network (CNN) [22] and Long Short-Term Memory (LSTM) [23,24].

3.1. Convolutional Neural Network

A one-dimensional (1D) convolution operation is shown in Figure 3. The difference between CNN and MLP is that CNN uses the concept of weight sharing. In Figure 3, x_1 to x_6 are inputs, and c_1 to c_4 are the feature maps after 1D convolution. What connects the input layer and convoluting layer are red, blue, and green connections. Each connection has its own weight value, and the connections of the same color have the same weight value. Therefore, in Figure 3, it only needs 3 weight values to perform the convolution operation. The advantage of CNN is that the training is relatively easy because the number of weights is less than that of fully-connected architecture. Moreover, important features can be effectively extracted.

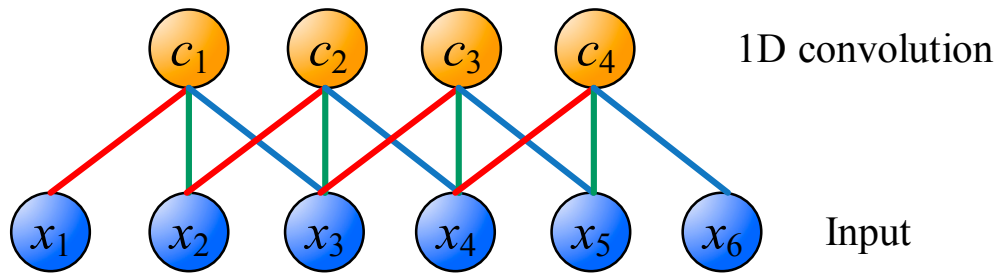


Figure 3. The one-dimensional (1D) convolution operation.

3.2. Long Short-Term Memory

Another important technology of ANN is Recurrent Neural Network (RNN), which differs from CNN and MLP in its consideration of the time sequence. LSTM [18] is one of the RNN models. The schematic of LSTM is shown in Figure 4, where σ is a sigmoid function, as shown in Equation (1). LSTM contains an input gate, an output gate and a forget gate. The interactive operation among these three gates makes LSTM have the sufficient ability to solve the problem of long-term dependencies which general RNNs cannot learn. In addition, a common problem in deep neural networks is called gradient vanishing, i.e., The learning speed of the previous hidden layers is slower than the deeper hidden layers. This phenomenon may even lead to a decrease of accuracy rate as hidden layers increase [25]. However, the smart design of the memory cell in LSTM can effectively solve the problem of gradient vanishing in backpropagation and can learn the input sequence with longer time steps. Hence, LSTM is commonly used for solving applications related to time serial issues. The specific formula derivation of LSTM is illustrated in Equations (2)–(11):

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

$$\bar{z}^t = W_z x^t + R_z y^{t-1} + b_z \quad (2)$$

$$z^t = \tanh(\bar{z}^t) \quad (3)$$

$$\bar{i}^t = W_i x^t + R_i y^{t-1} + p_i \odot c^{t-1} + b_i \quad (4)$$

$$i^t = \text{sigmoid}(\bar{i}^t) \quad (5)$$

$$\bar{f}^t = W_f x^t + R_f y^{t-1} + p_f \odot c^{t-1} + b_f \quad (6)$$

$$f^t = \text{sigmoid}(\bar{f}^t) \quad (7)$$

$$c^t = z^t \odot i^t + c^{t-1} \odot f^t \quad (8)$$

$$\bar{o}^t = W_o x^t + R_o y^{t-1} + p_o \odot c^{t-1} + b_o \quad (9)$$

$$o^t = \text{sigmoid}(\bar{o}^t) \quad (10)$$

$$y^t = \tanh(c^t) \odot o^t \quad (11)$$

where W_z , W_i , W_f , and W_o are input weights; R_z , R_i , R_f , and R_o are recurrent weights, p_i , p_f , and p_o are peephole weights; b_z , b_i , b_f , and b_o are bias weights; z^t is the block input gate; f^t is the forget gate; c^t is the cell; o^t is the output gate; y^t is the block output; and \odot represents point-wise multiplication. To reach the goal of parameter optimization, either CNN or LSTM can use backpropagation to adjust the parameters of the model during the process of training.

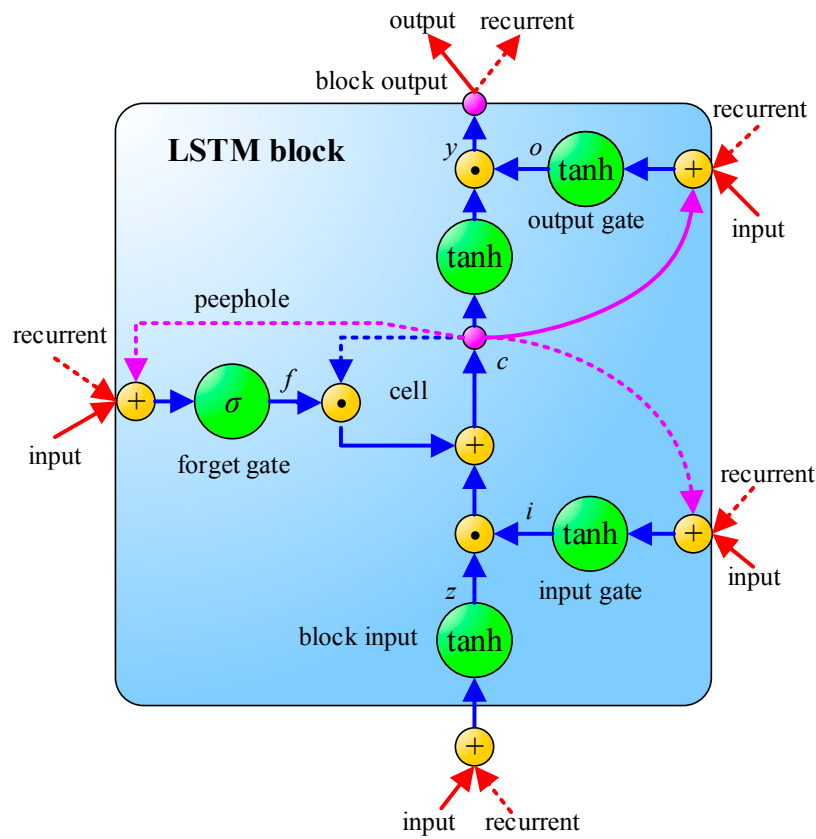


Figure 4. The schematic of Long Short-Term Memory (LSTM) [24].

3.3. Batch Normalization

During the training of deep neural network, some problems still emerge. For instance, due to the large number of layers within deep neural networks, a change of the parameters of one layer can usually affect the outputs of all the succeeding layers, which leads to frequent parameter modifications, and thus, a low training efficiency. Additionally, before passing the activation function, if the output value of a nerve cell exceeds dramatically the appropriate range of the activation function itself, it may also result in the failure of the work of the nerve cell. To solve these problems, batch normalization [26] is designed. The detailed formulas of batch normalization are shown in Equations (12)–(15):

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (12)$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (13)$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \quad (14)$$

$$y_i = \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad (15)$$

where x_i is the input value and y_i is the output after batch normalization; m refers to the mini-batch size, i.e., the one mini-batch that has m inputs; μ_B is the mean of all the inputs in the same mini-batch; and σ_B^2 is the variance of the input in a mini-batch. Next, according to the values of μ_B and σ_B^2 , all the x_i are normalized as \hat{x}_i and substituted into Equation (15) to obtain y_i , in which γ and β are

learnable parameters. Through batch normalization, the neurons in the deep neural network can be fully exploited and the training efficiency can be improved.

4. The Proposed Deep CNN-LSTM Network

The architecture of the proposed APNet is shown in Figure 5. The inputs of APNet are the records of the PM_{2.5} concentration, cumulated wind speeds, and cumulated hours of rain over the last 24 h. The output is the PM_{2.5} concentration of the next hour. Different from traditional pure CNN or pure LSTM architectures, the first half of APNet is CNN, and used for feature extraction. The latter half of APNet is LSTM forecasting, which is used to analyze the features extracted by CNN and then to estimate the PM_{2.5} concentration of the next point in time. The CNN part of the APNet contains three 1D convolution layers. Moreover, to improve the efficiency, batch normalization is added after the second and third convolution layers of the APNet.

Usually Rectified Linear Unit (ReLU), as shown in (6), is widely used as the activation function. However, for the activation function of APNet here, Scaled Exponential Linear Units (SELU), as shown in (7), is used. This is because, compared with ReLU, SELU has better convergence and can effectively avoid the problem of gradient vanishing, which is discussed specifically in Klambauer et al. [27]. In Equation (7), $\lambda = 1.05$, $\alpha = 1.67$, and the numerical values are specifically defined by Klambauer et al. [27]. The output of LSTM goes through the fully-connected architecture and the sigmoid activation function to produce the final output. The results represent the PM_{2.5} concentration of the next point in time.

$$\text{ReLU}(x) = \max(0, x) \quad (16)$$

$$\text{SELU}(x) = \lambda \begin{cases} x & \text{if } x > 0 \\ \alpha e^x - \alpha & \text{otherwise} \end{cases} \quad (17)$$

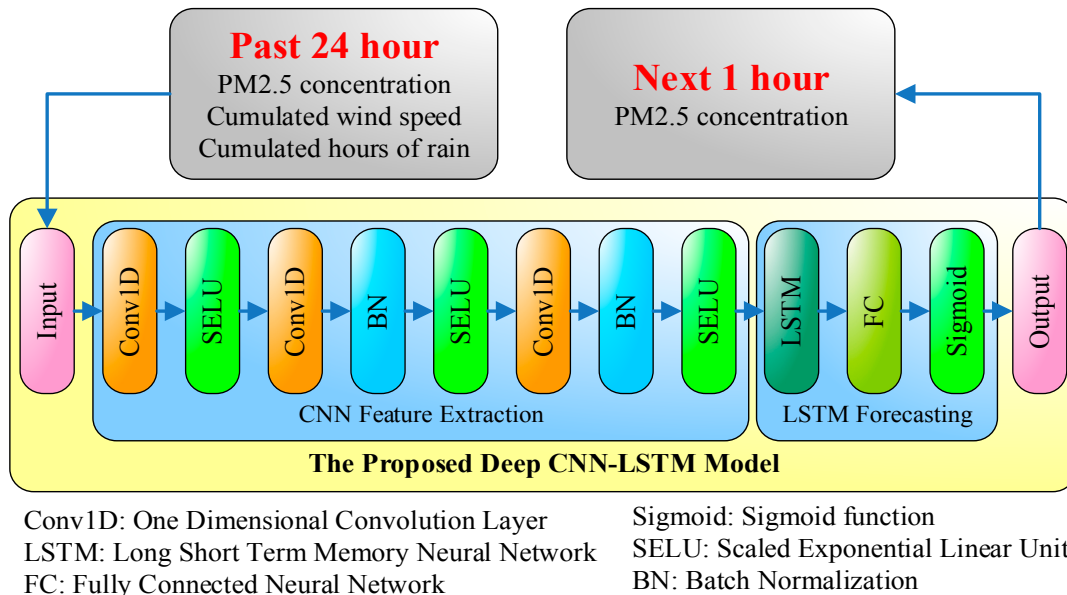


Figure 5. The architecture of the proposed APNet.

The system flow diagram of the proposed APNet is shown in Figure 6. During data processing, the original dataset first normalized, i.e., the numerical values of all dimensions are restricted to a range of 0 to 1, so as not to be overly partial to a certain dimension during training. Next, the normalized data is separated into two parts: training data and testing data. To keep the impartiality of performance evaluation, only the training data is used during the training, while the testing data is not used.

Each time the training data are input to the APNet, a loss value is generated, according to which the optimizer uses a backpropagation method to adjust the parameters of APNet. The forecast result of APNet will be more and more accurate with the increase of training iterations. After the APNet training is finished, the testing data is input into the APNet, and the testing results and real results are compared to evaluate the performance of the APNet.

When there is not enough training data or when there is overtraining, overfitting may occur. However, there are many ways to avoid overfitting, such as regularization [28], data augmentation [22], dropout [29], dropconnect [30], or early stopping [31]. Regularization, which is very popular in the field of deep learning, can be divided into L1 regularization and L2 regularization. Both of these methods will reduce the weight value of the neuronal network as much as possible to prevent overfitting [32]. The concept of data augmentation is to amplify the dataset as much as possible, for example adding random bias or noise, etc., to make the training data more diversified to achieve better training results. Dropout is similar to the dropconnect concept in that the former randomly stops the operation of the neuro, while the latter removes the connection randomly. The method used in this paper is early stopping. Before the experiment, we decided when to stop training according to the prediction condition of the validation data. For example, when training loss continues to decrease but validation loss increases, this means there is already overfitting [31], so at this time we would stop training. In the experiment, we selected an epoch value that does not generate overfitting, and let each neural network model be trained based on this epoch to maintain the fairness of the performance comparison.

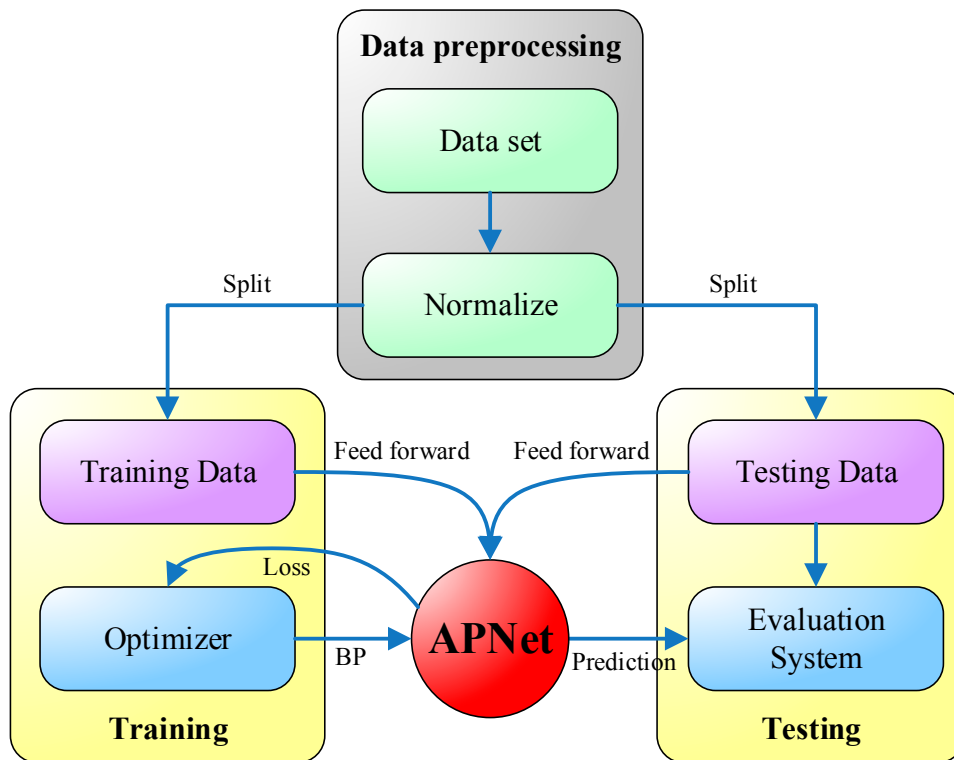


Figure 6. The system flow diagram of the proposed APNet.

5. Experimental Results and Discussion

This section is divided into two parts: data descriptions and experimental results. Support Vector Machine (SVM) [33–38], Random Forest (RF) [39–44], Decision Tree (DT) [45–50], MLP, CNN, and LSTM are used for comparison to fully demonstrate the performance of the proposed APNet.

5.1. Data Descriptions

Beijing is a cosmopolis with a population of more than 21.5 million, and Particulate Matter (PM) is one of the main factors that affect human health directly [51]. Thus, the PM_{2.5} dataset of Beijing is selected for this study. Figure 7 shows the weather condition, pollution degree reported and its histograms in each hour by the US embassy in Beijing, China, from 2010 to 2014. The dataset includes PM_{2.5} concentration, cumulated wind speed, and cumulated hours of rain. In this experiment, information from these factors over the past 24 h are used to forecast the PM_{2.5} concentration of the next hour. These three types of useful information are expected to be integrated into the machine learning model to perform supervised learning and analysis, to realize accurate forecasting.

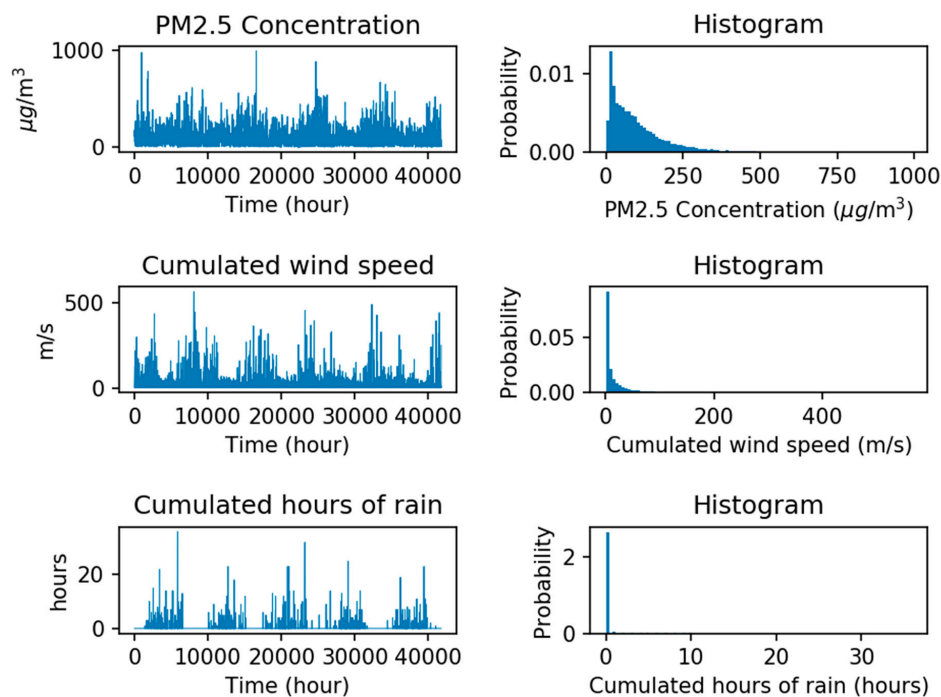


Figure 7. Beijing PM_{2.5} dataset.

5.2. Experiment Results

In this experiment, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Pearson correlation coefficient and Index of Agreement (IA) are taken for the performance evaluation. These four kinds of measurement indexes with their equations are shown in (18)–(21). r is the Pearson correlation coefficient. p_n denotes the predicted value, and o_n represents the observed values. \bar{o} is the average value of o_n , and N is the predicted length. To test the performance comprehensively, 10 intervals in the database are selected, with each interval containing six months' data as training data, and two months' data as testing data. The Pearson residuals of all forecasting methods is shown in Figure 8. The results are distinguished between those with an absolute value less than 1, an absolute value between 1 and 3, and an absolute value greater than 3, the results are plotted as shown in Figure 8. From the statistical results, it can be found that the distribution of the Pearson residuals for each machine learning is not too wide, this also means that these methods have a considerable degree of predictability.

$$\text{MAE} = \frac{1}{N} \sum_{n=1}^N |o_n - p_n| \quad (18)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{n=1}^N (o_n - p_n)^2}{N}} \quad (19)$$

$$r = \frac{N \sum_{n=1}^N o_n p_n - \sum_{n=1}^N o_n \sum_{n=1}^N p_n}{\sqrt{N \sum_{n=1}^N o_n^2 - \left(\sum_{n=1}^N o_n\right)^2} \sqrt{N \sum_{n=1}^N p_n^2 - \left(\sum_{n=1}^N p_n\right)^2}} \quad (20)$$

$$\text{IA} = 1 - \frac{\sum_{n=1}^N (|p_n - o_n|)^2}{\sum_{n=1}^N (|p_n - \bar{o}| + |o_n - \bar{o}|)^2} \quad (21)$$

Figures A1–A7 in Appendix A are the forecast results from each algorithm, and Figure A8 is the forecast results comparison of all the algorithms. In order to be able to perform a more complete evaluation of the effectiveness of all algorithms, we devised 10 tests for the experiments of this paper. Considering the length of this paper, we only list the results of six tests in Figures A1–A8, the detailed numerical analysis and comparison is presented in detail in Tables 1–4. From the figures, it can be found that SVM is slightly weak on PM_{2.5} forecasting and deviated greatly from the trend of the real result at some parts. Although the performance of DT is a little better than SVM, its error is still large. The efficiencies of MLP and RF are acceptable. Although at some parts the forecasting is still not accurate, the overall trend followed that of the real results. It should be noted that the efficiency of the CNN-LSTM based APNet proposed in this paper is better than that of CNN and LSTM. Therefore, it is proven that the application of APNet to PM_{2.5} forecasting is quite effective and accurate. In these experiments, the computer specifications used for the experiment of this paper are described below: CPU: Intel Xeon E3-1245 V6; Random Access Memory (RAM): 8 GB DDR4; graphics card: GTX 1080 Ti; hard disk drive: 1 TB SATA3; Operating System: Linux Ubuntu 16.04. Because the calculation times for predicting PM_{2.5} concentration through various algorithms are all very short, all experiment methods used in this paper are within a reasonable range for predicting PM_{2.5} concentration for the next hour.

Moreover, the detailed MAE, RMSE, Pearson correlation coefficient, and IA values are shown in Tables 1–4. In the ranking of MAE, there are, from low to high, APNet (14.63446), LSTM (15.34655), CNN (16.13498), RF (17.56451), MLP (20.65027), DT (21.41757), and SVM (37.90581). While in the ranking of RMSE, there are, from low to high, APNet (24.22874), LSTM (24.2925), CNN (24.59636), RF (28.87602), MLP (29.09238), DT (39.45547), and SVM (50.02137). Besides, in the ranking of Pearson correlation coefficient, there are, from high to low, APNet (0.959986), CNN (0.958363), LSTM (0.95794), RF (0.942276), MLP (0.940557), DT (0.880291), and SVM (0.81133). Finally, in the ranking of IA, there are, from high to low, APNet (0.97831), LSTM (0.976797), CNN (0.975972), RF (0.966298), MLP (0.964021), DT (0.933533), and SVM (0.852398). Experiments show that the APNet algorithm proposed in this paper is very good when the Pearson correlation coefficient is presented, in which the first, third, fifth, seventh, eighth, and tenth tests all have the highest r value, and the average value is also the best among all machine learning methods. In terms of IA, APNet also scored highest in IA in the first, third, fifth, seventh, eighth, and tenth tests, the average score is also the best. Overall, CNN, LSTM, and APNet are the best performers; while APNet, which combines the advantages of CNN and LSTM, wins out. This result also confirms that the combination of CNN and LSTM is very effective for the prediction of PM_{2.5}. As shown by the experiment results, the performances of CNN and LSTM are both good, but that of APNet is even better. It is also proven that for PM_{2.5} air pollution source forecasting, it is very beneficial to first perform feature extraction using CNN, and then input the feature values into the LSTM architecture.

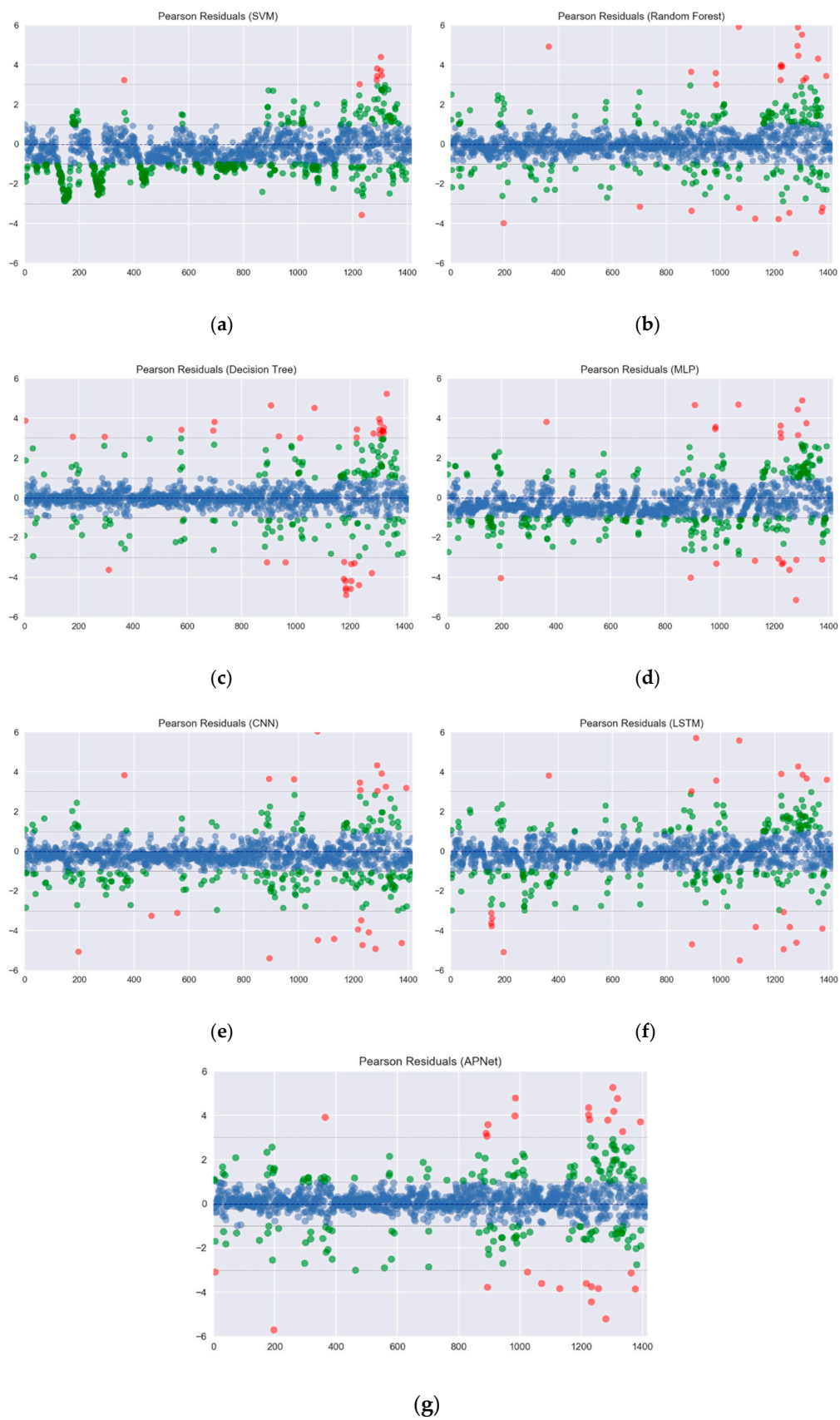


Figure 8. The Pearson residuals of all forecasting methods: (a) Partial results A; (b) Partial results B; (c) Partial results C; (d) Partial results D; (e) Partial results E; (f) Partial results F; (g) Partial results G.

Table 1. The experimental results in terms of Mean Absolute Error (MAE).

Test	SVM	RF	DT	MLP	CNN	LSTM	APNet
#1	42.57556	18.68328	23.90568	22.4221	18.9675	18.5217	16.7474
#2	35.40574	14.92391	19.53063	22.0437	14.8997	16.2908	14.2053
#3	43.37174	16.74816	17.93104	20.2441	16.9613	15.8297	14.9131
#4	50.19538	31.64949	36.57292	23.1328	20.7791	18.1417	18.2807
#5	40.38873	19.54953	27.66294	22.8951	17.1051	16.505	17.2492
#6	34.57838	17.80561	21.3065	18.5993	15.1543	13.9768	14.0047
#7	37.10853	12.3846	15.37398	19.9247	15.3203	13.1789	11.9718
#8	21.85433	9.96139	11.07522	13.9672	11.1243	11.1574	9.85554
#9	40.47121	21.13339	25.09194	26.0607	18.954	17.2029	18.9953
#10	33.1085	12.80574	15.72481	17.213	12.0842	12.6606	10.1216
Average	37.90581	17.56451	21.41757	20.65027	16.13498	15.34655	14.63446

Table 2. The experimental results in terms of Root Mean Square Error (RMSE).

Test	SVM	RF	DT	MLP	CNN	LSTM	APNet
#1	56.55255	26.59535	36.90484	29.98992	26.36855	25.2699	23.83181
#2	47.07641	26.84212	38.17991	30.86026	25.24918	27.20435	25.95273
#3	55.9933	25.46634	29.14463	27.68189	24.43146	23.31643	22.56656
#4	66.58581	47.20812	58.96869	35.14076	31.38514	29.63356	31.08485
#5	50.32762	31.14631	55.65785	31.59871	26.4418	27.15832	26.77069
#6	47.23936	32.32307	43.69507	27.00565	23.87708	23.05538	24.81823
#7	48.11796	22.96514	33.33885	28.78185	24.29253	23.04227	20.83558
#8	27.70533	16.61144	19.44406	19.52802	16.63667	17.22178	16.44391
#9	57.49434	39.29988	44.9455	38.8347	31.03137	30.14096	35.23974
#10	43.12105	20.30241	34.27529	21.50208	16.24985	16.88207	14.7433
Average	50.02137	28.87602	39.45547	29.09238	24.59636	24.2925	24.22874

Table 3. The Pearson correlation coefficient ($n = 1415$).

Test	SVM	RF	DT	MLP	CNN	LSTM	APNet
#1	0.638786	0.926131	0.857044	0.907166	0.935633	0.940295	0.941237
#2	0.92699	0.973356	0.945972	0.968823	0.977848	0.973044	0.975517
#3	0.754792	0.944363	0.926856	0.936873	0.950255	0.953075	0.955411
#4	0.872546	0.924315	0.868861	0.957647	0.970539	0.970023	0.966768
#5	0.70376	0.893368	0.699291	0.89043	0.922092	0.919221	0.932416
#6	0.870895	0.938605	0.879954	0.956404	0.966881	0.967185	0.964074
#7	0.843806	0.966459	0.927678	0.947582	0.964757	0.966151	0.972383
#8	0.887029	0.957205	0.943408	0.941748	0.95875	0.953544	0.96088
#9	0.914454	0.959145	0.940049	0.961928	0.9731	0.97354	0.963773
#10	0.700245	0.939808	0.8138	0.936971	0.963777	0.963319	0.967397
Average	0.81133	0.942276	0.880291	0.940557	0.958363	0.95794	0.959986

Table 4. The Index of Agreement (IA).

Test	SVM	RF	DT	MLP	CNN	LSTM	APNet
#1	0.745175	0.958607	0.923722	0.943082	0.959601	0.963882	0.968546
#2	0.952324	0.98613	0.972305	0.980782	0.988124	0.985715	0.987253
#3	0.716799	0.968534	0.962342	0.964832	0.972961	0.974219	0.976896
#4	0.873168	0.95108	0.92713	0.975282	0.979759	0.983128	0.981386
#5	0.790755	0.940903	0.82489	0.93198	0.958817	0.957693	0.961527
#6	0.897091	0.960562	0.924618	0.974253	0.982193	0.982024	0.978416
#7	0.904886	0.982324	0.961747	0.970803	0.979047	0.982588	0.985856
#8	0.924705	0.977994	0.97085	0.967449	0.977596	0.975862	0.979732
#9	0.934477	0.973919	0.967458	0.974426	0.984648	0.985924	0.980962
#10	0.784602	0.962931	0.900264	0.957321	0.976973	0.976935	0.982527
Average	0.852398	0.966298	0.933533	0.964021	0.975972	0.976797	0.97831

Figure 9 shows the detailed comparison results of each model, where the blue bold line refers to the real data, and the other colored lines are the forecast results of each algorithm. As shown in the blue frame of Figure 9, the forecast results of SVM barely coincided with the actual results. Among all the algorithms, the performances of RF, MLP, CNN, LSTM, and APNet are better. As shown in the green frame of Figure 9, when the PM_{2.5} pollution source concentration is unstable, the forecasting result of many algorithms could not follow the real trend and showed a rather disordered pattern. This also indicates that it is still difficult in terms of PM_{2.5} forecasting. Overall, the performances of CNN and LSTM are very stable and accurate, but the CNN-LSTM based APNet proposed in this paper is even better. The forecasting ability of APNet for PM_{2.5} forecasting is also verified in this experiment.

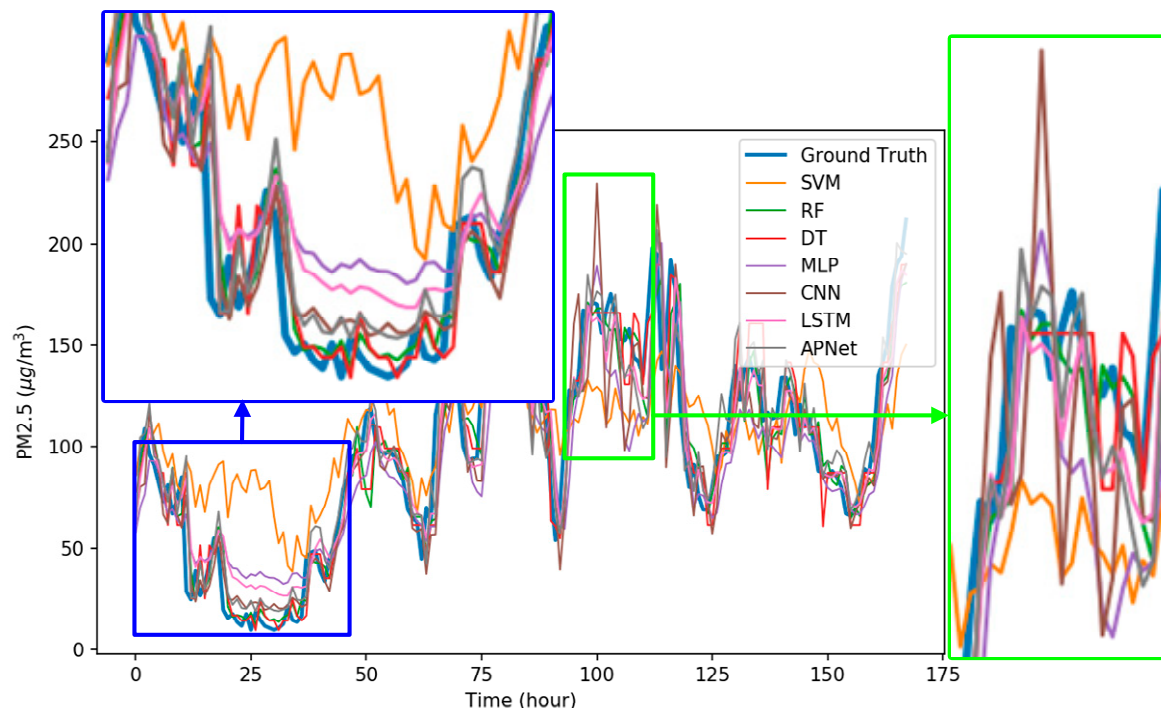


Figure 9. The comparisons details of forecasting results.

For ease of analysis, we classified air quality according to PM_{2.5} concentration as follows: Good: PM_{2.5} does not exceed 35 µg/m³; Pollution: PM_{2.5} is greater than 35 µg/m³; Severe Pollution: PM_{2.5} is greater than 150 µg/m³. Good quality air conditions appear in Beijing for about 23% of the time, more than half of the time (about 55%), the city is in a state of general pollution; about 22% of the time Beijing is in a state of serious pollution, general pollution and severe pollution together accounts for 77%. The proportion of the three air quality conditions has not changed much from 2010 to 2014. Compared to spring and summer, more days of clean air and severe pollution exist during autumn and winter. The former is due to Beijing's northerly winds in autumn and winter, which facilitates air diffusion and increases the proportion of clean air. The latter is likely due to winter heating and straw burning during autumn, which causes heavy pollution to occur frequently, so the proportion of serious pollution is also relatively high. The proportion of severe pollution days in summer in Beijing is less than 17%, but the proportion of clean air days in the summer is also the lowest among the four seasons with less than 16%. Although emission from residential heating using coal is lower in summer than in winter, the temperature and humidity is higher in Beijing in the summer; at the same time, the northerly winds are reduced in summer and wind speed is low, some factors are favorable for the generation of secondary aerosols and PM_{2.5} concentration increases [52].

Because the concentration of PM_{2.5} is closely related to city area, urban population, number of vehicles, and urban industrial activity increase [53], this paper proposes a prediction model (APNet) to

make short term predictions of $PM_{2.5}$ concentrations in order to provide more effective and accurate early warnings of high concentrations of suspended particulate matter, in order to protect the people's respiratory health and prevent cardiovascular disease.

The advantages of separate monitoring are as follows: (1) From an academic research point of view, the shorter the monitoring data collection cycle the better, that is, the more data collected in the same time period, the more applicable research can be done in the future, because the data sampling period required for each applied research is different, so separate monitoring can avoid the failing of missing data; (2) Before smart city is reached, there are still many researches and technological developments that need big data to support. In the future, big data will become a very important research asset. Figure 2 is only a schematic diagram, it is not necessary to measure data at different locations during the data collection process, it could also be done at the same location. However, in the smart city, sensors could be installed more densely in different locations so that the smart city and even neighboring areas are covered with a network of sensors, and more innovative prediction algorithms can be developed and more accurate spatiotemporal data analysis can be achieved.

The main contribution of this paper is to develop a deep neural network model that integrates the CNN and LSTM architectures, and through historical data such as cumulated hours of rain, cumulated wind speed and $PM_{2.5}$ concentration. We allow this model to use such information to learn and predict $PM_{2.5}$ concentration for the next hour. In the experiment process, the testing data is entirely new for the neural network model, the purpose being to verify the predictive power of APNet developed in this paper. The APNet predicted results are also analyzed and compared based on actual observed values to verify the performance of each forecasting model. Therefore, in addition to modeling past data, APNet's output value also represents the forecasting result.

This paper mainly applies the deep neural network method to predict $PM_{2.5}$, and compares it with many other popular and widely used machine learning algorithms. However, deep neural network is also a type of machine learning, whether the data is sufficient and correct will determine the success or failure of the algorithm prediction. Therefore, when using machine learning for data molding or forecasting, data collection and processing is very important. This does not mean however that the traditional rule-based approach is superior, because in modern society with large data resources, machine learning technology can more subtly discover information that humans cannot intuitively reflect, and thus produce more accurate forecasts.

6. Conclusions

In this paper, a deep neural network model (APNet) based on CNN-LSTM is proposed to estimate $PM_{2.5}$ concentration. APNet can forecast the $PM_{2.5}$ concentration of the next hour according to the $PM_{2.5}$ concentration, cumulated wind speed, and cumulated hours of rain over the last 24 h. A $PM_{2.5}$ dataset of Beijing was used in this experiment to perform model training and performance evaluation. The experimental data in this paper were classified into two parts: training data and testing data. Training data was used for model training. The testing data that was unused in the training process was used for the computation of MAE, RMSE, Pearson correlation coefficient, and IA for performance evaluation, the results of which were comprehensively compared with that of the SVM, RD, DT, MLP, CNN, and LSTM architectures. Experimental results showed that compared with the traditional machine learning methods, the forecasting performance of the APNet proposed in this paper was proven to be the best, and its average MAE and RMSE were both the lowest. As for the CNN-LSTM based model, its feasibility and practicality for forecasting the $PM_{2.5}$ concentration were also verified in this paper. This technology is significantly beneficial for improving the ability of estimating the air pollution in smart cities. In the future, this study can be applied to the prevention and control of $PM_{2.5}$. In particular, in light of the severe situation of atmospheric particulate matter pollution in recent years, we must come up with appropriate countermeasures to curb the deterioration of urban air conditions. However, an urban forest can be introduced as a large air filter which is non-toxic, harmless, and non-polluting, and also saves time, labor, and resources in reducing air pollution.

Urban forests have the effect of preventing air particles from lingering in the air, and it also controls and eliminates airborne particles. Research in this area may become a new direction for regulating airborne particulates with plants [54].

Author Contributions: C.-J.H. wrote the program and designed the CNN-LSTM model. P.-H.K. designed this study and collected the dataset. C.-J.H. and P.-H.K. drafted and revised the manuscript.

Funding: This work was supported by the Jiangxi University of Science and Technology, People's Republic of China, under Grant jxxjbs18019.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

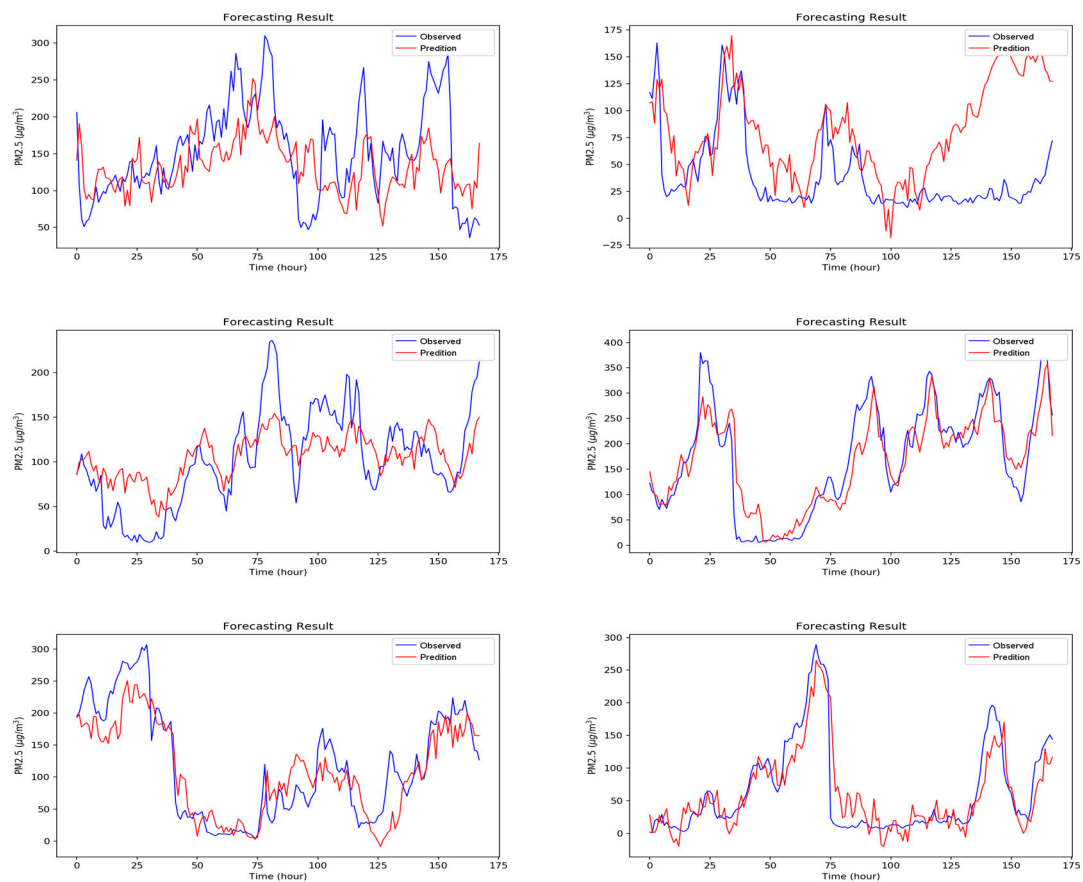


Figure A1. The forecasting results of Support Vector Machine (SVM).

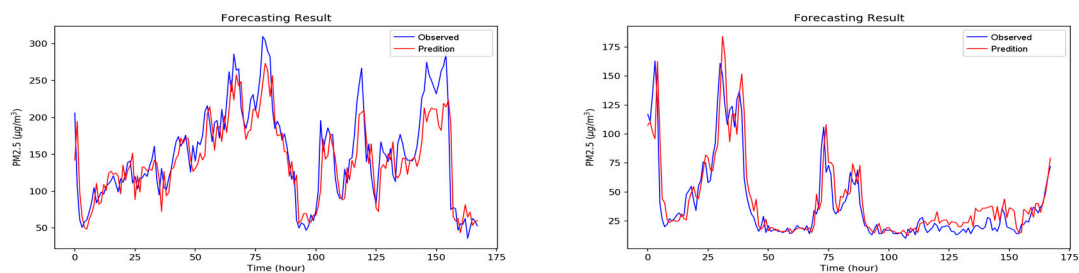


Figure A2. Cont.

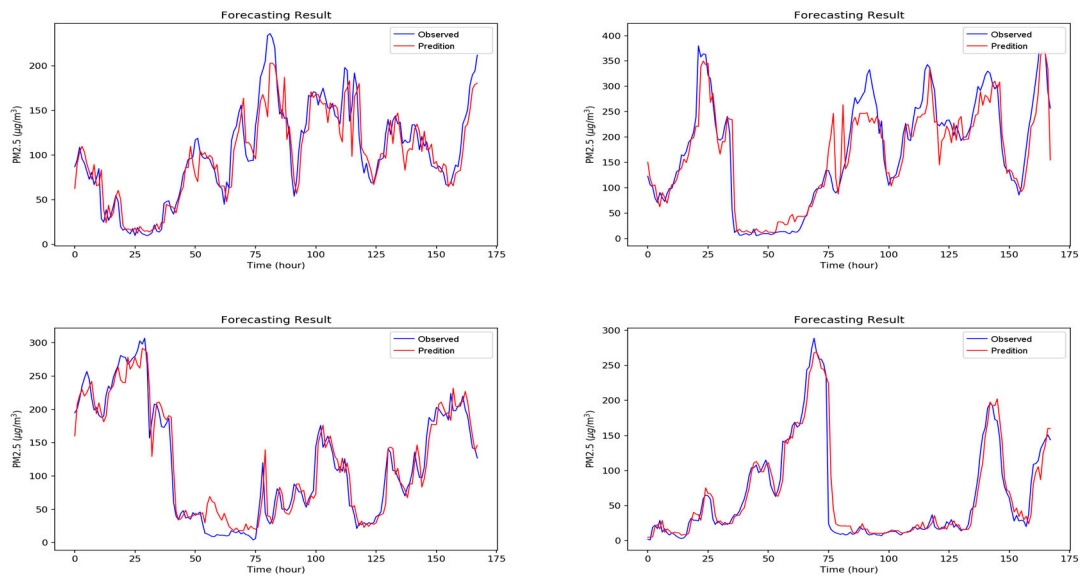


Figure A2. The forecasting results of random forest.

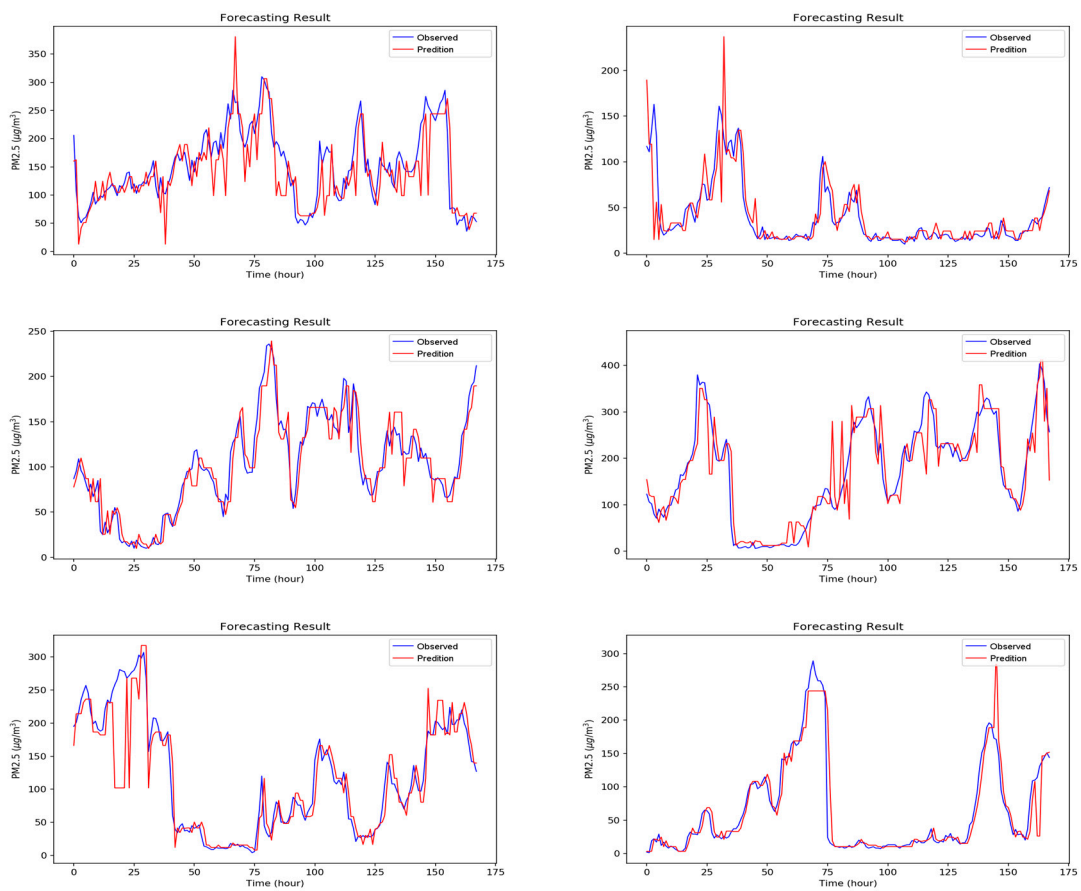


Figure A3. The forecasting results of decision tree.

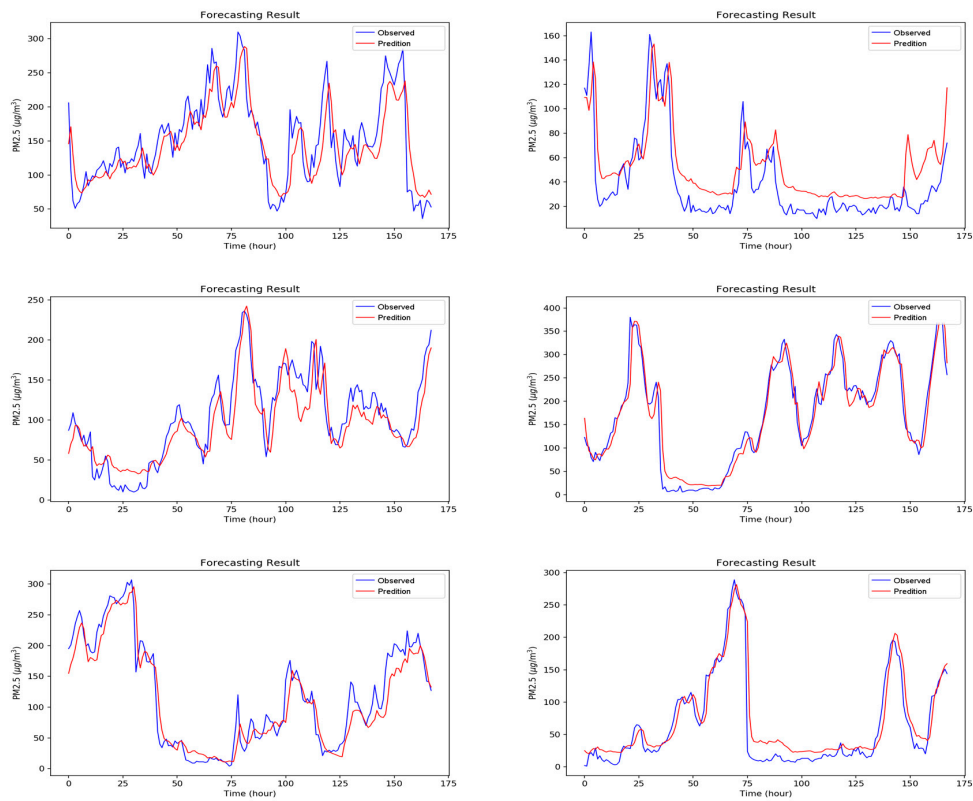


Figure A4. The forecasting results of Multilayer Perceptron (MLP).

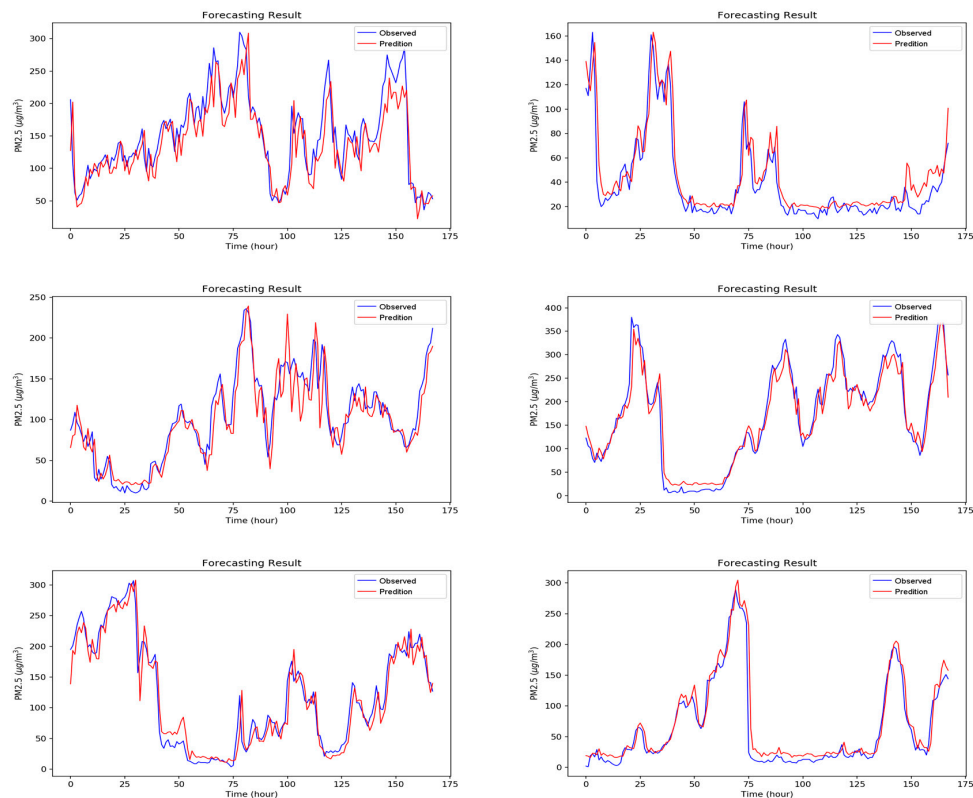


Figure A5. The forecasting results of Convolutional Neural Network (CNN).

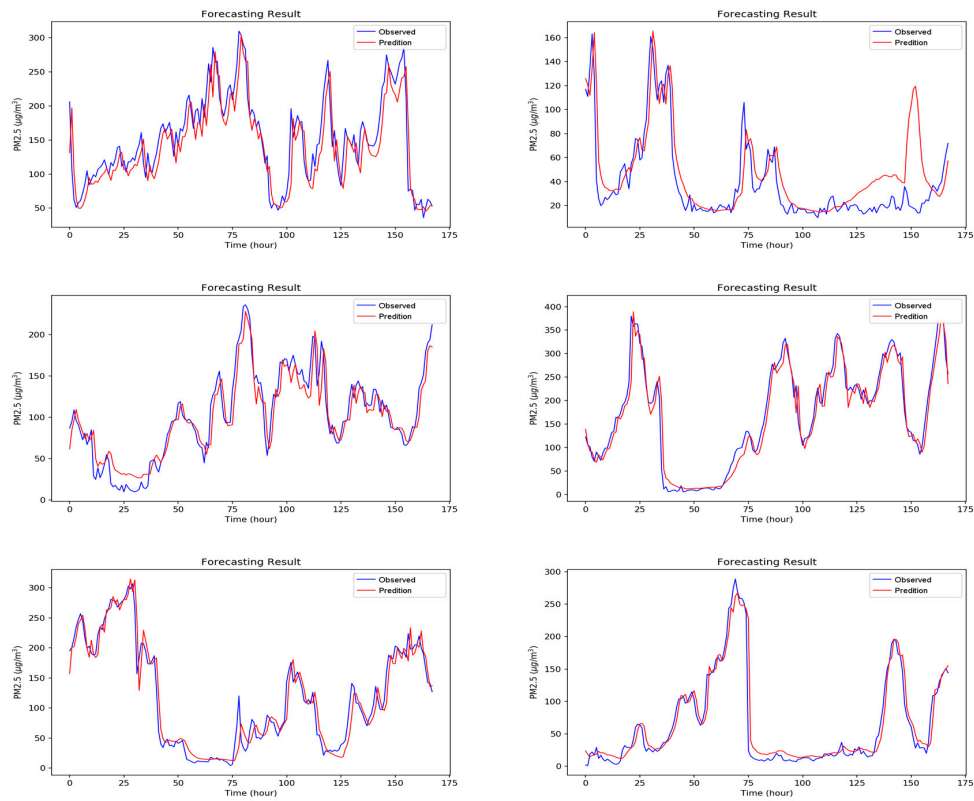


Figure A6. The forecasting results of LSTM.

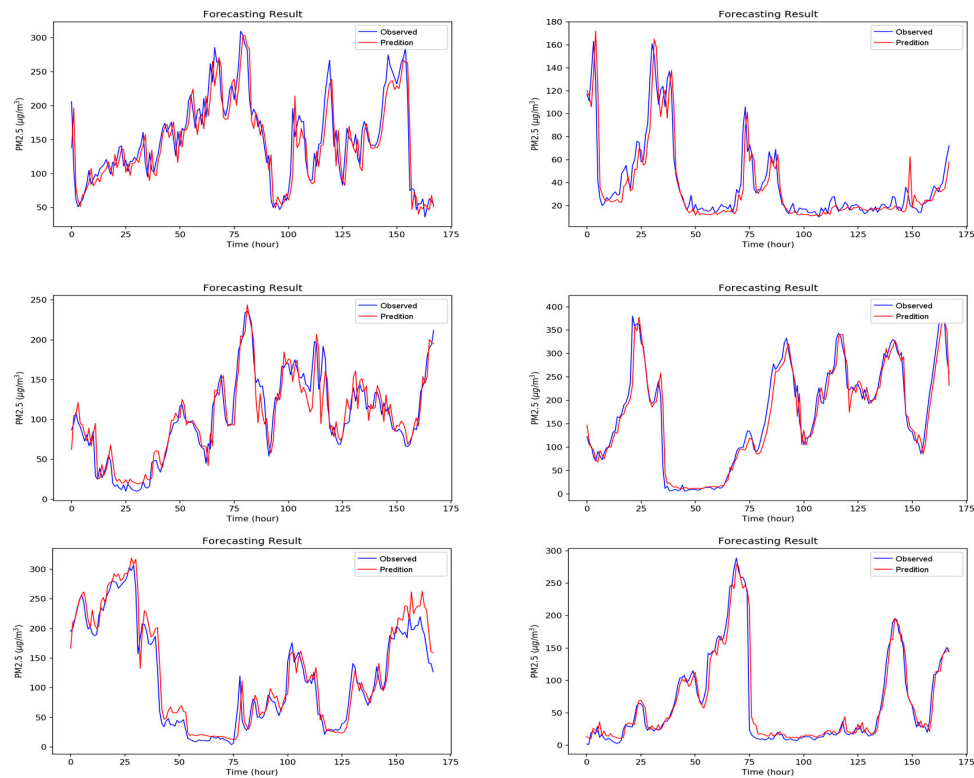


Figure A7. The forecasting results of the proposed APNet.

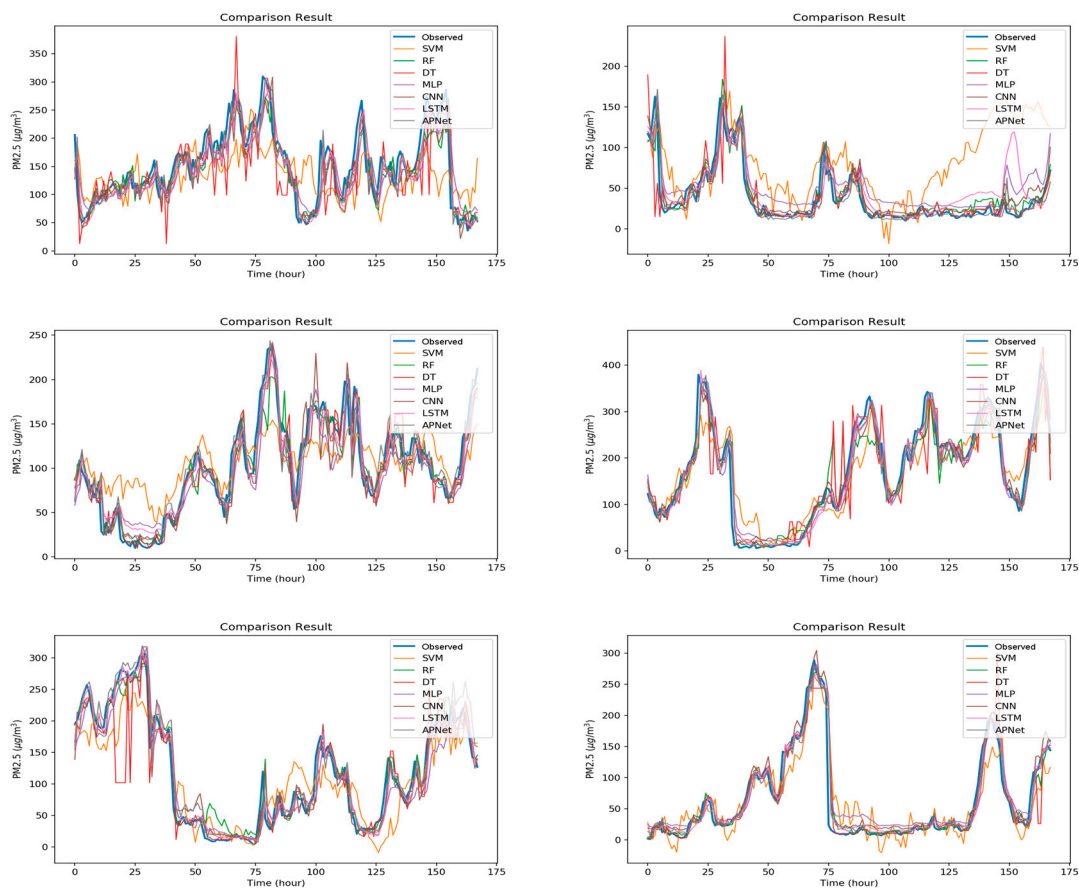


Figure A8. The comparisons of all the forecasting results.

References

1. International Energy Agency. Available online: <https://www.iea.org/> (accessed on 22 February 2018).
2. World Energy Outlook Special Report 2016. Available online: <https://www.iea.org/publications/freepublications/publication/WorldEnergyOutlookSpecialReport2016EnergyandAirPollution.pdf> (accessed on 22 February 2018).
3. Chen, L.-J.; Ho, Y.-H.; Lee, H.-C.; Wu, H.-C.; Liu, H.-M.; Hsieh, H.-H.; Huang, Y.-T.; Lung, S.-C.C. An Open Framework for Participatory PM_{2.5} Monitoring in Smart Cities. *IEEE Access* **2017**, *5*, 14441–14454. [CrossRef]
4. Han, L.; Zhou, W.; Li, W. City as a major source area of fine particulate (PM_{2.5}) in China. *Environ. Pollut.* **2015**, *206*, 183–187. [CrossRef] [PubMed]
5. Kioumourtzoglou, M.-A.; Schwartz, J.; James, P.; Dominici, F.; Zanobetti, A. PM_{2.5} and mortality in 207 US cities. *Epidemiology* **2015**, *27*, 221–227. [CrossRef]
6. Walsh, M.P. PM_{2.5}: Global progress in controlling the motor vehicle contribution. *Front. Environ. Sci. Eng.* **2014**, *8*, 1–17. [CrossRef]
7. Liu, J.; Li, Y.; Chen, M.; Dong, W.; Jin, D. Software-defined internet of things for smart urban sensing. *IEEE Commun. Mag.* **2015**, *53*, 55–63. [CrossRef]
8. Zhang, N.; Chen, H.; Chen, X.; Chen, J. Semantic framework of internet of things for smart cities: Case studies. *Sensors* **2016**, *16*, 1501. [CrossRef] [PubMed]
9. Zeng, Y.; Xiang, K. Adaptive Sampling for Urban Air Quality through Participatory Sensing. *Sensors* **2017**, *17*, 2531. [CrossRef] [PubMed]
10. Ghaffari, S.; Caron, W.-O.; Loubier, M.; Normandeau, C.-O.; Viens, J.; Lamhamedi, M.; Gosselin, B.; Messaddeq, Y. Electrochemical Impedance Sensors for Monitoring Trace Amounts of NO₃ in Selected Growing Media. *Sensors* **2015**, *15*, 17715–17727. [CrossRef] [PubMed]

11. Lary, D.J.; Lary, T.; Sattler, B. Using Machine Learning to Estimate Global PM_{2.5} for Environmental Health Studies. *Environ. Health Insights* **2015**, *9*, 41–52. [[CrossRef](#)]
12. Li, X.; Peng, L.; Hu, Y.; Shao, J.; Chi, T. Deep learning architecture for air quality predictions. *Environ. Sci. Pollut. Res.* **2016**, *23*, 22408–22417. [[CrossRef](#)] [[PubMed](#)]
13. Li, X.; Peng, L.; Yao, X.; Cui, S.; Hu, Y.; You, C.; Chi, T. Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation. *Environ. Pollut.* **2017**, *231*, 997–1004. [[CrossRef](#)] [[PubMed](#)]
14. Yu, S.; Mathur, R.; Schere, K.; Kang, D.; Pleim, J.; Young, J.; Tong, D.; Pouliot, G.; McKeen, S.A.; Rao, S.T. Evaluation of real-time PM_{2.5} forecasts and process analysis for PM_{2.5} formation over the eastern United States using the Eta-CMAQ forecast model during the 2004 ICARTT study. *J. Geophys. Res.* **2008**, *113*, D06204. [[CrossRef](#)]
15. Wang, Y.; Muth, J.F. An optical-fiber-based airborne particle sensor. *Sensors* **2017**, *17*, 2110. [[CrossRef](#)] [[PubMed](#)]
16. Shao, W.; Zhang, H.; Zhou, H. Fine particle sensor based on multi-angle light scattering and data fusion. *Sensors* **2017**, *17*, 1033. [[CrossRef](#)] [[PubMed](#)]
17. Feng, X.; Li, Q.; Zhu, Y.; Hou, J.; Jin, L.; Wang, J. Artificial neural networks forecasting of PM_{2.5} pollution using air mass trajectory based geographic model and wavelet transformation. *Atmos. Environ.* **2015**, *107*, 118–128. [[CrossRef](#)]
18. Dunea, D.; Pohoata, A.; Iordache, S. Using wavelet–feedforward neural networks to improve air pollution forecasting in urban environments. *Environ. Monit. Assess.* **2015**, *187*, 477. [[CrossRef](#)] [[PubMed](#)]
19. Kuo, P.-H.; Chen, H.-C.; Huang, C.-J. Solar Radiation Estimation Algorithm and Field Verification in Taiwan. *Energies* **2018**, *11*, 1374. [[CrossRef](#)]
20. Law Amendment Urged to Combat Air Pollution. Available online: http://www.china.org.cn/environment/2013-02/22/content_28031626_2.htm (accessed on 1 July 2018).
21. Orbach, J. Principles of Neurodynamics. Perceptrons and the Theory of Brain Mechanisms. *Arch. Gen. Psychiatry* **1962**, *7*, 218. [[CrossRef](#)]
22. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
23. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
24. Greff, K.; Srivastava, R.K.; Koutnik, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2222–2232. [[CrossRef](#)] [[PubMed](#)]
25. Why Are Deep Neural Networks Hard to Train? Available online: <http://neuralnetworksanddeeplearning.com/chap5.html> (accessed on 1 July 2018).
26. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 37, pp. 448–456.
27. Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. Self-Normalizing Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
28. Dan Foresee, F.; Hagan, M.T. Gauss-Newton approximation to bayesian learning. In Proceedings of the IEEE International Conference on Neural Networks, Houston, TX, USA, 2 June 1997; IEEE: Piscataway, NJ, USA, 1997; Volume 3, pp. 1930–1935.
29. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958. [[CrossRef](#)]
30. Wan, L.; Zeiler, M.; Zhang, S.; LeCun, Y.; Fergus, R. Regularization of neural networks using dropconnect. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 1058–1066.
31. Prechelt, L. Early Stopping | but when? In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 55–69, ISBN 978-3-642-35288-1, 978-3-642-35289-8.
32. Improving the Way Neural Networks Learn. Available online: <http://neuralnetworksanddeeplearning.com/chap3.html> (accessed on 1 July 2018).

33. Suykens, J.A.K.; Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* **1999**, *9*, 293–300. [[CrossRef](#)]
34. Wang, S.; Hae, H.; Kim, J. Development of easily accessible electricity consumption model using open data and GA-SVR. *Energies* **2018**, *11*, 373. [[CrossRef](#)]
35. Niu, D.; Li, Y.; Dai, S.; Kang, H.; Xue, Z.; Jin, X.; Song, Y. Sustainability Evaluation of Power Grid Construction Projects Using Improved TOPSIS and Least Square Support Vector Machine with Modified Fly Optimization Algorithm. *Sustainability* **2018**, *10*, 231. [[CrossRef](#)]
36. Liu, J.P.; Li, C.L. The short-term power load forecasting based on sperm whale algorithm and wavelet least square support vector machine with DWT-IR for feature selection. *Sustainability* **2017**, *9*, 1188. [[CrossRef](#)]
37. Das, M.; Akpınar, E. Investigation of Pear Drying Performance by Different Methods and Regression of Convective Heat Transfer Coefficient with Support Vector Machine. *Appl. Sci.* **2018**, *8*, 215. [[CrossRef](#)]
38. Wang, J.; Niu, T.; Wang, R. Research and application of an air quality early warning system based on a modified least squares support vector machine and a cloud model. *Int. J. Environ. Res. Public Health* **2017**, *14*, 249. [[CrossRef](#)] [[PubMed](#)]
39. Liaw, A.; Wiener, M. Classification and Regression by randomForest. *R News* **2002**, *2*, 18–22. [[CrossRef](#)]
40. Zhu, M.; Xia, J.; Jin, X.; Yan, M.; Cai, G.; Yan, J.; Ning, G. Class Weights Random Forest Algorithm for Processing Class Imbalanced Medical Data. *IEEE Access* **2018**, *6*, 4641–4652. [[CrossRef](#)]
41. Ma, J.; Qiao, Y.; Hu, G.; Huang, Y.; Sangaiyah, A.K.; Zhang, C.; Wang, Y.; Zhang, R. De-Anonymizing Social Networks With Random Forest Classifier. *IEEE Access* **2018**, *6*, 10139–10150. [[CrossRef](#)]
42. Huang, N.; Lu, G.; Xu, D. A Permutation Importance-Based Feature Selection Method for Short-Term Electricity Load Forecasting Using Random Forest. *Energies* **2016**, *9*, 767. [[CrossRef](#)]
43. Hassan, M.; Southworth, J. Analyzing Land Cover Change and Urban Growth Trajectories of the Mega-Urban Region of Dhaka Using Remotely Sensed Data and an Ensemble Classifier. *Sustainability* **2017**, *10*, 10. [[CrossRef](#)]
44. Quintana, D.; Sáez, Y.; Isasi, P. Random Forest Prediction of IPO Underpricing. *Appl. Sci.* **2017**, *7*, 636. [[CrossRef](#)]
45. Safavian, S.R.; Landgrebe, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man. Cybern.* **1991**, *21*, 660–674. [[CrossRef](#)]
46. Huang, N.; Peng, H.; Cai, G.; Chen, J. Power Quality Disturbances Feature Selection and Recognition Using Optimal Multi-Resolution Fast S-Transform and CART Algorithm. *Energies* **2016**, *9*, 927. [[CrossRef](#)]
47. Alani, A.Y.; Osunmakinde, I.O. Short-Term Multiple Forecasting of Electric Energy Loads for Sustainable Demand Planning in Smart Grids for Smart Homes. *Sustainability* **2017**, *9*, 1972. [[CrossRef](#)]
48. Rosli, N.; Rahman, M.; Balakrishnan, M.; Komeda, T.; Mazlan, S.; Zamzuri, H. Improved Gender Recognition during Stepping Activity for Rehab Application Using the Combinatorial Fusion Approach of EMG and HRV. *Appl. Sci.* **2017**, *7*, 348. [[CrossRef](#)]
49. Rau, C.-S.; Wu, S.-C.; Chien, P.-C.; Kuo, P.-J.; Chen, Y.-C.; Hsieh, H.-Y.; Hsieh, C.-H.; Liu, H.-T. Identification of Pancreatic Injury in Patients with Elevated Amylase or Lipase Level Using a Decision Tree Classifier: A Cross-Sectional Retrospective Analysis in a Level I Trauma Center. *Int. J. Environ. Res. Public Health* **2018**, *15*, 277. [[CrossRef](#)] [[PubMed](#)]
50. Rau, C.-S.; Wu, S.-C.; Chien, P.-C.; Kuo, P.-J.; Chen, Y.-C.; Hsieh, H.-Y.; Hsieh, C.-H. Prediction of Mortality in Patients with Isolated Traumatic Subarachnoid Hemorrhage Using a Decision Tree Classifier: A Retrospective Analysis Based on a Trauma Registry System. *Int. J. Environ. Res. Public Health* **2017**, *14*, 1420. [[CrossRef](#)] [[PubMed](#)]
51. Wang, J.-F.; Hu, M.-G.; Xu, C.-D.; Christakos, G.; Zhao, Y. Estimation of Citywide Air Pollution in Beijing. *PLoS ONE* **2013**, *8*, e53400. [[CrossRef](#)] [[PubMed](#)]
52. Study on PM_{2.5} Pollution in Beijing Urban District from 2010 to 2014. Available online: http://www.stat-center.pku.edu.cn/Stat/Index/research_show/id/169 (accessed on 1 July 2018).

53. Statistical Analysis of Air Pollution in Five Cities in China. Available online: http://www.stat-center.pku.edu.cn/Stat/Index/research_show/id/215 (accessed on 1 July 2018).
54. Hwang, H.J.; Yook, S.J.; Ahn, K.H. Experimental investigation of submicron and ultrafine soot particle removal by tree leaves. *Atmos. Environ.* **2011**, *45*, 6987–6994. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).