

Article

Air Pollution Prediction Using Long Short-Term Memory (LSTM) and Deep Autoencoder (DAE) Models

Thanongsak Xayasouk ^{1,†} , HwaMin Lee ^{2,†} and Giyeol Lee ^{3,*}¹ Department of Computer Science, Soonchunhyang University, Asan 31538, Korea; trafalga.ttt@gmail.com² Department of Computer Software & Engineering, Soonchunhyang University, Asan 31538, Korea; leehm@sch.ac.kr³ Department of Landscape Architecture, Chonnam National University, Gwangju 61186, Korea

* Correspondence: gylee@jnu.ac.kr; Tel.: +82-62-530-2108

† These authors contributed equally to this work.

Received: 2 February 2020; Accepted: 19 March 2020; Published: 24 March 2020



Abstract: Many countries worldwide have poor air quality due to the emission of particulate matter (i.e., PM₁₀ and PM_{2.5}), which has led to concerns about human health impacts in urban areas. In this study, we developed models to predict fine PM concentrations using long short-term memory (LSTM) and deep autoencoder (DAE) methods, and compared the model results in terms of root mean square error (RMSE). We applied the models to hourly air quality data from 25 stations in Seoul, South Korea, for the period from 1 January 2015, to 31 December 2018. Fine PM concentrations were predicted for the 10 days following this period, at an optimal learning rate of 0.01 for 100 epochs with batch sizes of 32 for LSTM model, and DAEs model performed best with batch size 64. The proposed models effectively predicted fine PM concentrations, with the LSTM model showing slightly better performance. With our forecasting model, it is possible to give reliable fine dust prediction information for the area where the user is located.

Keywords: air pollution; deep autoencoder (DAE); deep learning; long short-term memory (LSTM); fine particulate matter; PM₁₀; PM_{2.5}

1. Introduction

As industry and population expand rapidly in South Korea, air pollution is increasingly becoming problematic for human health in the country. In 2017, South Korea ranked 173rd among the 180 countries with the greatest air pollution impact [1]. Air pollution in urban areas consists of carbon dioxide (CO₂), carbon monoxide (CO), nitrogen oxide (NO₂), nitrogen monoxide (NO), ozone (O₃), and fine particulate matter (PM), the last of which is of greatest concern in South Korea. Fine PM is classified into PM₁₀ and PM_{2.5} based on particle diameter, where PM₁₀ and PM_{2.5} are particles with diameters <10 and <2.5 μm, respectively (Figure 1). PM includes dust, pollen, soot, smoke, and liquid droplets that harm the respiratory system [2,3], causing respiratory symptoms including irregular heart rate, coughing, airway irritation, abnormal lung function, breathing difficulty, heart attack, stroke-associated diseases, and asthma. Despite increasing air pollutant concentrations in South Korea, the Korean government has reported difficulties in gaining accurate air pollution data due to insufficient air pollution measurement stations for reasonable nationwide coverage.

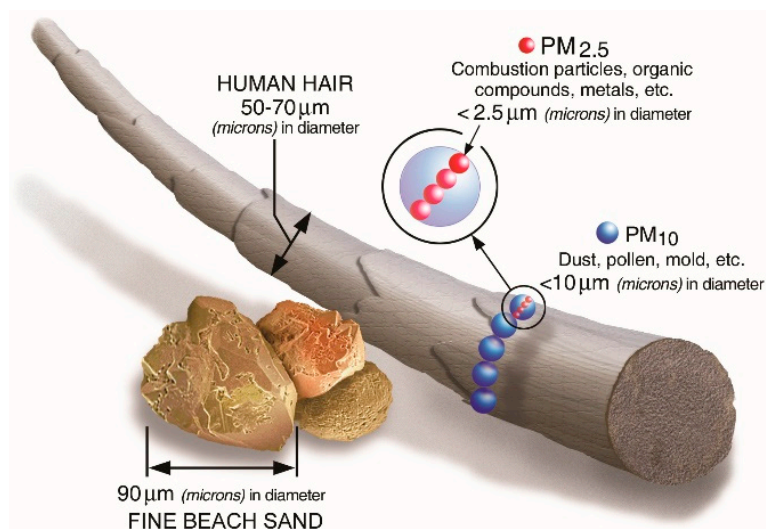


Figure 1. Particulate matter (PM) size comparison.

Thus, many studies have been conducted to determine and analyze air quality. The recent development of machine learning techniques, especially deep learning, have provided new opportunities to improve air quality research. Deep learning consists of an artificial intelligence (AI) system that can obtain data unsupervised, in unstructured or unlabeled learning approaches such as deep neural network or deep neural learning methods [4]. Deep learning requires three essential elements: the graphics processing unit (GPU), which controls operation processing speed; vast quantities of data for experiments; and signal information processing. Deep learning has been widely adopted in academic and practical applications such as translation, speech recognition, language processing, and image classification [5,6] (Figure 2). Several studies of air quality prediction have also adopted AI and deep learning techniques [7–14]; many of these studies have used deep neural networks to obtain short-term air quality forecasts. Using these approaches, current fine PM concentrations have been found to be strongly correlated with pollution emissions from power plants, factory chimneys, and various other sources.

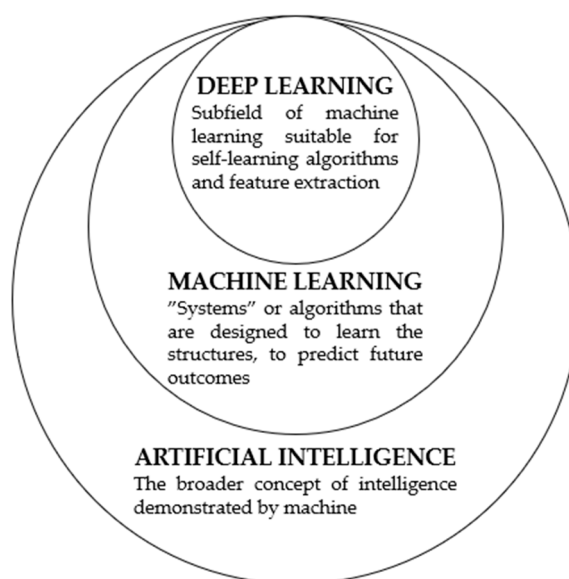


Figure 2. Relationships among artificial intelligence (AI) approaches.

In the current study, we used hourly PM₁₀ and PM_{2.5} measurement data collected in Seoul, South Korea during 2015–2018, as well as data on meteorological features such as humidity, rain, wind speed

and direction, temperature, and atmospheric conditions. These air pollution data attributes were learned by long short-term memory (LSTM) and deep autoencoder (DAE) models. The models were then used to predict fine PM concentrations in Seoul, and the performance of the two models was compared in terms of root mean square error (RMSE).

2. Related Research

Kalapnidas et al. [15] reported detailed air pollution effects using ordinal air pollution data (low, medium, high, and alarm levels) with a case-based reasoning (CBR) system and the lazy learning method. Similarly, Athanasiadis et al. [16] predicted air pollution based on O₃ concentrations classified as low, medium, and high levels of pollutants including SO₂, NO, and NO₂ using a σ -fuzzy lattice neurocomputing (FLN) model. Land-use regression was also applied to estimate NO_x and NO₂ concentrations [17], and O₃ concentrations [18]. Hoek et al. [19] concluded that land-use regression methods are able to model annual mean PM_{2.5} concentrations. The LUR model is considered to be suitable for PM_{2.5} prediction due to the linear relationship between PM_{2.5} and explanatory variables, while the ANN based model designed to handle non-linearity may perform better in general as well [20]. Kunwar et al. [21] applied an ensemble learning method and a principal components analysis (PCA) algorithm to integrate air quality data to forecast air quality index (AQI) values. However, these approaches involving regression of categorical variables can produce ambiguous results because some data are ignored.

Various studies have predicted air pollutant concentrations under different circumstances. Corani [22] forecasted hourly O₃ and PM₁₀ concentrations from previous-day air pollution data using a neural network algorithm to train pruned neural network (PNN) and feed-forward neural network (FFNN) models. Fu et al. [23] also applied an FFNN model with a undulating scheme and the gray method. Jiang et al. [24] predicted air pollution using traditional chemical and physical models in combination with regression and multiple-layer perceptron models. Ni et al. [25] found that a linear regression model performed better than several other models for predicting fine PM concentrations in Beijing, China.

Detailed air pollution predictions have been obtained by combining various model designs based on LSTM and convolutional neural network (CNN) approaches. One such study proposed an experimental model to forecast fine PM concentrations [25]; another used LSTM and RNN models as a framework to obtain long-term PM_{2.5} trends from time-series data for use in government policy making and resource allocation [26]. Fully connected LSTM (LSTM-FC) has been applied with a neural network to forecast and visualize PM concentrations at urban meteorological stations [27]. LSTM and RNN have also been used as a framework for large-scale, long-term time series data for PM forecasting [28]. Another study proposed an LSTM-based model to predict hourly fine PM concentrations at 25 target locations in Seoul [29].

Deep spatiotemporal learning based on an air quality forecast method has been applied to discuss spatial and temporal correlations in PM concentration based on a stacked autoencoder (SAE) model for training air pollution data with the greedy layer-wise technique [30]. These techniques have also been used to predict local traffic flow [31]. Another study applied multitask learning (MTL) approaches involving homogeneous and deep belief network (DBN) methods using unsupervised learning for predictive models [32].

A back-propagation (BP) neural network was combined with an integrated development environment (IDE) model to predict fine PM concentrations using meteorological and fine PM data for Chengdu, China [33]; model results were improved in the IDE-BPNN combination model. Another study applied a support vector machine (SVM) method using fine PM data, meteorological elements, and geographical information to predict air quality at pollution measurement stations by incorporating nonlinear PM characteristics [34].

3. Materials

3.1. Study Areas

Seoul, South Korea, contains 25 air pollution measurement stations (one station per district) separated by from one another by 5 km along the transverse Mercator (TM) link system (Figure 3). The stations are mainly situated far from major roadways and at the tops of public buildings.



Figure 3. Major air pollution monitoring stations (red dots) in Seoul, South Korea.

These stations automatically collect hourly air quality data 24 h per day; the data are then uploaded to a website that is open to the general public. Seoul also contains several special monitoring stations including the Namsan Mountain high-altitude station, the Gwanak Mountain station, which measures levels of air pollution that has traveled long distances, and the Bukhan Mountain station, which is located in a clean zone; there are also 14 roadside measurement stations and 12 measurement stations located on highway bus line medians.

3.2. Dataset

3.2.1. PM Data

PM concentration ($\mu\text{g}/\text{m}^3$) data used in this study were derived from hourly measurements at the 25 monitoring stations in Seoul, South Korea, from 1 January 2015, to 31 December 2018 [35]. Trends in the PM_{10} and $\text{PM}_{2.5}$ data are shown in Figures 4 and 5, respectively.

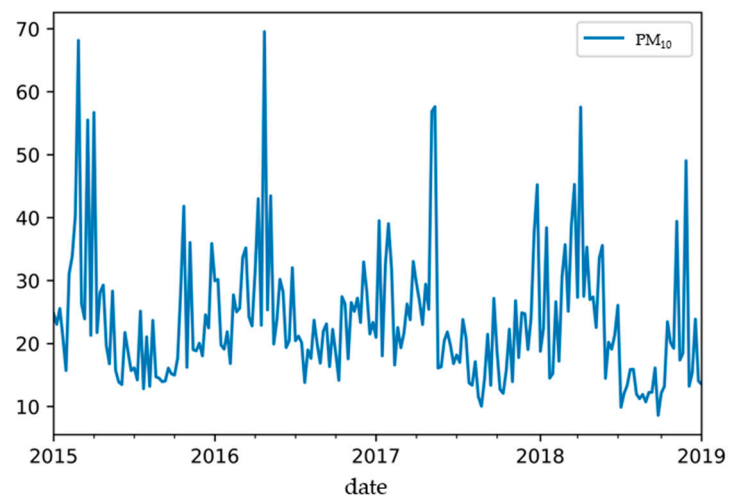


Figure 4. Hourly PM_{10} concentration data.

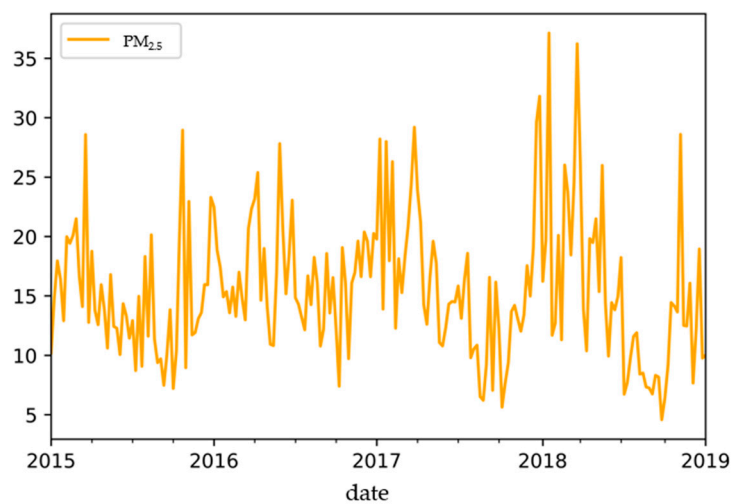


Figure 5. Hourly $PM_{2.5}$ concentration data.

3.2.2. Meteorological Data

Meteorological data for the study period were obtained from the Korea Meteorological Agency website [36]. The dataset contained preprocessed hourly values of wind speed, wind direction, temperature, sky condition, and rainfall (Figure 6).

Korean government agencies use an air quality index (AQI) to quantify air quality concentration effects for communication with the general public. This AQI has five categories (Table 1), which indicate relative health risks due to air pollution.

Table 1. Air quality index (AQI) classification.

AQI		Description
PM_{10}	$PM_{2.5}$	
0–30	0–15	Good
31–50	16–25	Moderate
51–100	26–50	Unhealthy
100+	50+	Hazardous

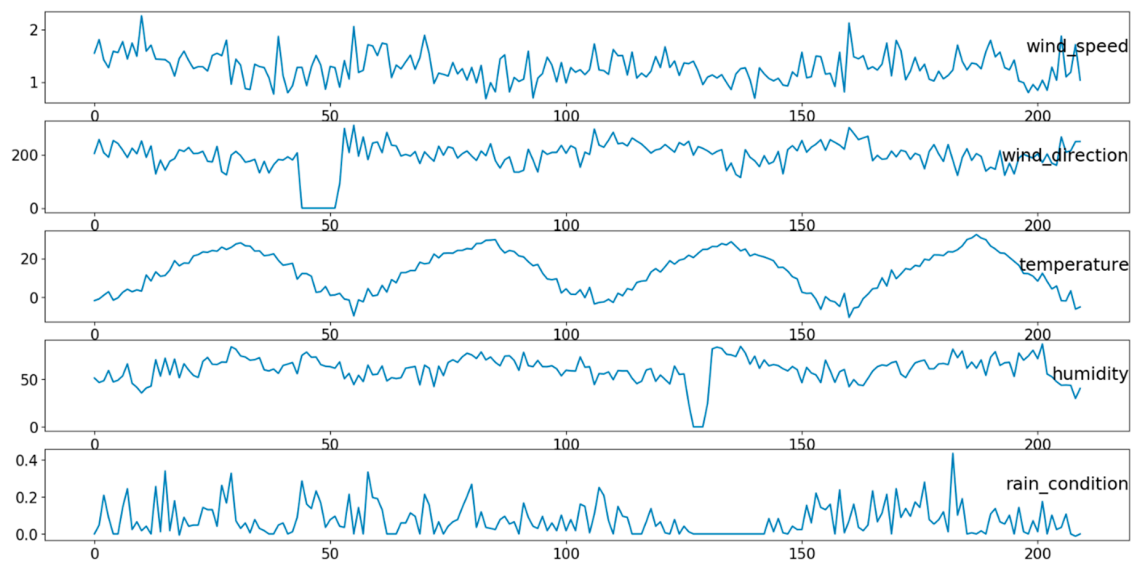


Figure 6. Hourly meteorological data.

4. Proposed Methods

4.1. LSTM Models

RNNs have been used to create sequential information data for many deep learning applications including translation [37], image classification [38], voice recognition [39], and object tracking [40]. Two types of RNNs are LSTM [41] and gated recurrent units (GRUs) [42].

The RNN architecture is unrolled or unfolded to show the entire network as a complete sequence, with one layer per word (Figure 7).

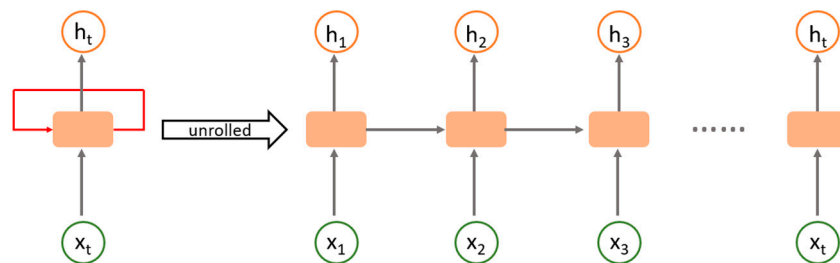


Figure 7. Recurrent neural network (RNN) architecture.

The recursive RNN formulas are as follows:

$$h_t = \tanh(W_h h_{t-1} + W_x x_t) \quad (1)$$

$$y_t = W_y h_t \quad (2)$$

where x_t is the input vector, h_t is the hidden layer, y_t is the experiment output vector, and W_h is a weighted matrix. The RNN is applied to LSTM to create an environment for the computation process, obtain input, and create output [43]. During this process, long-term memory is created from short-term memory. The LSTM system consists of an input gate, a forget gate, and an output gate.

LSTM calculates the hidden state as follows:

$$i_t = \sigma(W_f[h_{t-1}, x_t] + b_i) \quad (3)$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (4)$$

$$\text{Sigmoid} = \frac{1}{1 + e^{-1}} \quad (5)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (6)$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (7)$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \quad (8)$$

$$h_t = o_t * \tanh(c_t) \quad (9)$$

where σ is the logistic sigmoid function; i , f , and o are the input, forget, and output gates, respectively; h is a hidden vector that is the same size in each layer; W is a weight matrix for the transformation of information from cell to gate vectors; and m is a vector-only feature in every gate that obtains input from feature m of the cell vector. In Equation (7), \tilde{c}_t is a hidden element that is tasked with the current input layer; c_t is the internal memory computed in this unit; and h_t is the output of a hidden state, derived through memory multiplication (Figure 8).

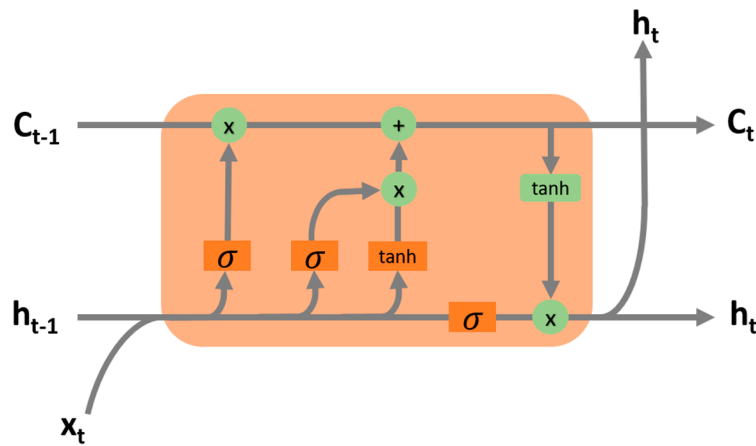


Figure 8. Long short-term memory (LSTM) architecture.

The forget gate (Figure 9a) is responsible for removing information from the cell state; it receives two inputs: the hidden state output from the previous time step (h_{t-1}) and the input for the current time step (x_t). These inputs are multiplied by weight matrices, and a bias is added. A sigmoid function is then applied to obtain an output vector with values ranging from 0 to 1, which is used to decide which values to keep and which to discard.

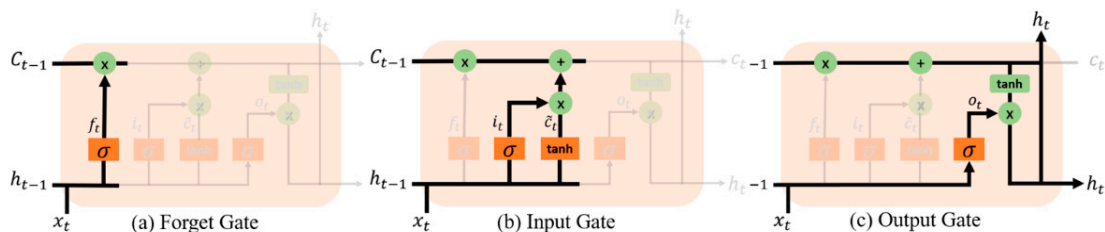


Figure 9. Long short-term memory (LSTM) gates.

Next, the input gate transfers information to the cell state in a two-step method (Figure 9b). Similar to the input gate, a sigmoid function is applied as a filter for h_{t-1} and x_t to build a vector of suitable values for the cell state ranging from -1 to 1 . This vector then provides values that can be added to the cell state.

The output gate (Figure 9c) decides which information to output from the cell state. In LSTM, the output gate function is performed in three steps. First, the vector is built and the hyperbolic tangent

function \tanh is applied to the cell state to scale the values from -1 to 1 . The sigmoid function is then applied to the previous hidden state to create a filter for values of h_{t-1} and x_t . Finally, the filtered values are multiplied by the vector created in step 1 to produce LSTM output information.

The LSTM algorithm used in our prediction system is described in detail in Table 2.

Table 2. Training the LSTM algorithm.

Step	Description
1	Preprocessing of all fine particulate matter and meteorological data LSTM pre-training
2	<ul style="list-style-type: none"> Denote $x_{(t)}$ as an element-wise input, ignoring bias Create weight matrix W to transform information from cell to gate vectors Define m for each element of the gate vector and obtain input from the cell state Set c as an internal memory of the cell state
3	Fine tuning <ul style="list-style-type: none"> Build a vector by applying the \tanh function to the cell state Apply the sigmoid function to create a filter for values of h_{t-1} and x_t
4	Obtain prediction results

4.2. Deep Autoencoders (DAEs)

4.2.1. Autoencoder

An autoencoder is a type of neural network that encodes input data for reconstruction as output data [44]. To begin this process, the autoencoder must learn to capture the significant features of the input. An example of an autoencoder with a single input layer, single hidden layer, and single output layer is shown in Figure 10. To train set $\{x(1), x(2), \dots, x(n)\}$ such that $x(i) \in \mathbb{R}^d$, the first step of the autoencoder model is to encode the single input $x(i)$ to hidden layer $y(x(i))$ according to Equation (10); this layer is then decoded as output layer $z(x(i))$ according to Equation (11), as follows:

$$y(x) = f(W_1x + b) \quad (10)$$

$$z(x) = g(W_2x + c) \quad (11)$$

where W_1 is a weight matrix for the optimization process, b is an encode bias vector, W_2 is a decoding matrix of the output layer, and c is a decoding bias vector. In this study, we also applied the logistic sigmoid function $1/(1 + \exp(-x))$ to $f(x)$ and $g(x)$.

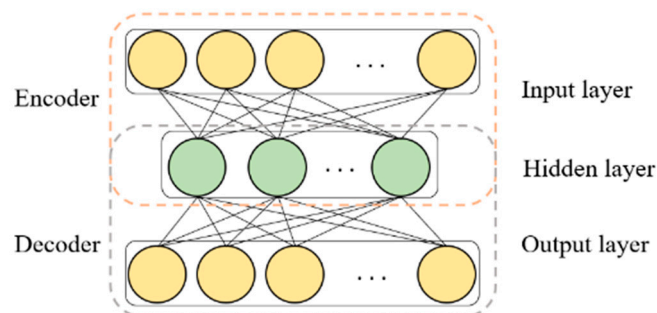


Figure 10. Processes of the autoencoder model.

The autoencoder model uses a vector input layer (x) and encoding function (f) to approximate another vector (y); during reconstruction, a decoder function (g) is applied to vector y to recreate vector x ; the resulting output layer from the application of (g) is vector z . Reconstruction error is determined by scaling with the loss function $LH(x,z)$; this function is minimized as $L(X, Z)$ to obtain optimal parameter values as follows:

$$\theta = \arg_{\theta} \min L(X, Z) = \arg_{\theta} \min \frac{1}{2} \sum_{i=1}^N \|x^{(i)} - z(x^{(i)})\|^2. \quad (12)$$

One urgent problem in the application of autoencoder models is the size of the hidden layer, which is set as equal to or larger than the output layer. This problem is generally addressed by the design of the model functions. In the present study, we used a nonlinear autoencoder with a hidden layer that is one unit larger than the input layer by applying the sparsity constraint method, such that the autoencoder model was transformed into a sparse autoencoder. To obtain sparse representation, we imposed a sparsity constraint to minimize reconstruction error as follows:

$$SAO = L(X, Z) + \gamma \sum_{i=1}^{H_D} KL(\rho \parallel \hat{\rho}_j) \quad (13)$$

$$\hat{\rho}_j = (1/N) \sum_{i=1}^N y_j(x^{(i)}) \quad (14)$$

where γ is the weight, H_D is the number of hidden units, ρ is the sparsity parameter, and H_D is the number of hidden units. In Equation (14), the average value of the activation function for hidden unit j in the training set is the Kullback–Leibler (KL) divergence for machine learning, $KL(\rho \parallel \hat{\rho}_j)$, which is calculated as follows:

$$KL(\rho \parallel \hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j}. \quad (15)$$

KL divergence defines the parameter $KL(\rho \parallel \hat{\rho}_j) = 0$ if $\rho = \hat{\rho}_j$. The sparsity constraint on the input process and back propagation (BP) method are applied to modify this problem.

4.2.2. The DAE Model

Deep or stacked autoencoder models are among the most powerful types of neural network architecture [45]. The DAE model begins by pre-training a single input layer, followed by hidden layers, such that the output of the k th hidden layer is used as input for the $(k + 1)$ th hidden layer. Thus, hidden layers are stacked hierarchically within the DAE, so the final hidden layer is a higher-level representation of all layers of input, and may be used in forecasting.

In this study, we applied a DAE model for fine PM forecasting by adding a standard forecaster at the top of the model layer. Layer-wise training of the resulting DAE is shown in Figure 11. Figure 12 shows the structure of a DAE, including stacked autoencoder nodes.

We applied a DAE model to represent fine PM features; the prediction was then applied to a logistic regression model. In the proposed method, the DAE model was combined with a dropout process to handle multiple faults. The workflow of the DAE model is shown in Figure 13, and the algorithm is described in detail in Table 3.

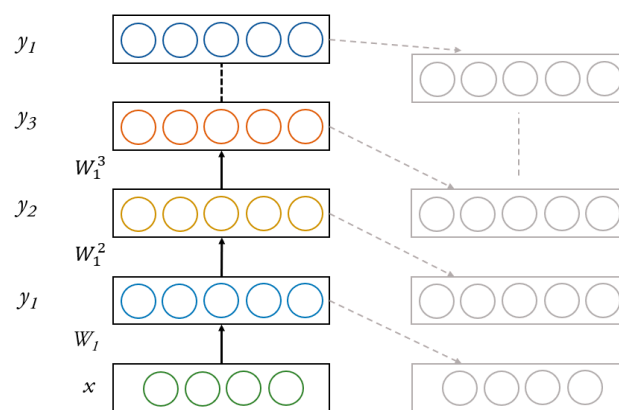


Figure 11. Layer-wise training of a deep autoencoder (DAE) model.

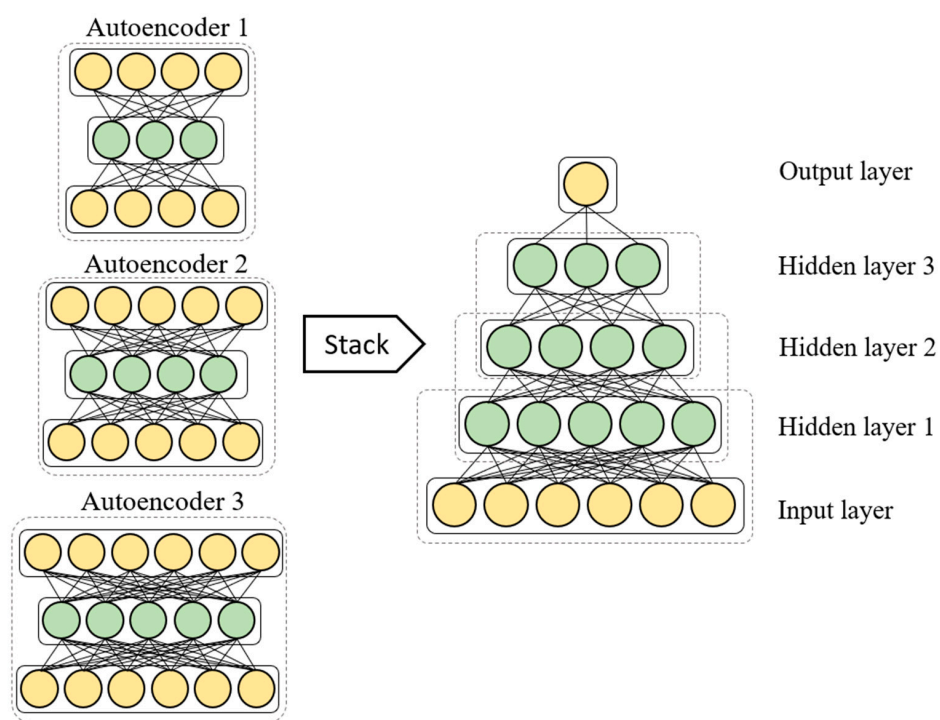


Figure 12. Stacked autoencoders within a DAE.

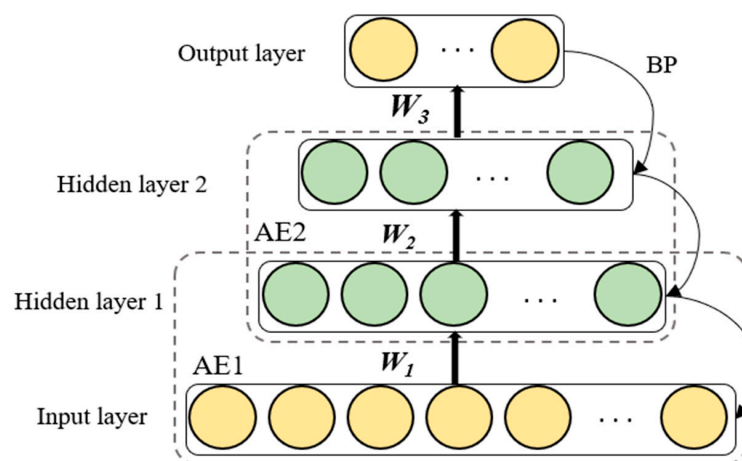


Figure 13. DAE workflow.

Table 3. Training the deep autoencoder (DAE) algorithm.

Step	Description
1	Preprocessing of all fine particulate matter and meteorological data Preparation of the DAE framework <ul style="list-style-type: none"> • Define a weight γ and parameter ρ to randomize the matrices • Apply the greedy layer-wise method to the hidden layers
2	<ul style="list-style-type: none"> • Assign the output of the kth hidden layer $\{W_1^{k+1}, b_1^{k+1}\}_{k=0}^{l-1}$, as the input of the $(k+1)$ th hidden layer • Determine reasonable encoding for the $(k+1)$ th hidden layer Fine tuning <ul style="list-style-type: none"> • Apply supervised training to define weight and bias terms $\{W_1^{k+1}, b_1^{k+1}\}$ to obtain random values
3	<ul style="list-style-type: none"> • Apply the gradient descent optimization method and back propagation to adjust hyperparameters throughout the network
4	Obtain prediction results

4.3. Model Performance Evaluation

We evaluated the performance of the proposed model in terms of the root mean square error (RMSE) between measured air pollution values and predicted values. RMSE was calculated as follows:

$$RMSE = \sqrt{\sum_{i=1}^N (P_m - P_r)^2 / N}, \quad (16)$$

where P_m and P_r are the measured and predicted PM concentrations, respectively, and N is the number of measured values.

5. Results

5.1. Fine PM Prediction

In this study, we obtained PM₁₀ and PM_{2.5} concentration data and meteorological data consisting of rainfall, wind speed and direction, temperature, humidity, and sky condition for use as input nodes. The output variable was predicted PM₁₀ or PM_{2.5} concentration. All data were partitioned into two sets, with 85% used for training and 15% for testing.

We combined all raw data obtained from the open data website and performed preprocessing to check for missing values and categorical values within the dataset. We then split the data into training and test datasets and applied the LSTM and DAE models to predict PM₁₀ and PM_{2.5} concentrations for the 10 days following the study period. Figure 14 shows the workflow for predicting PM concentrations using the LSTM and DAE model.

Finally, we evaluated the accuracy of the proposed method using the RMSE between observed and predicted values. We adjusting the learning rate, epoch, and batch size of the model to obtain optimal results.

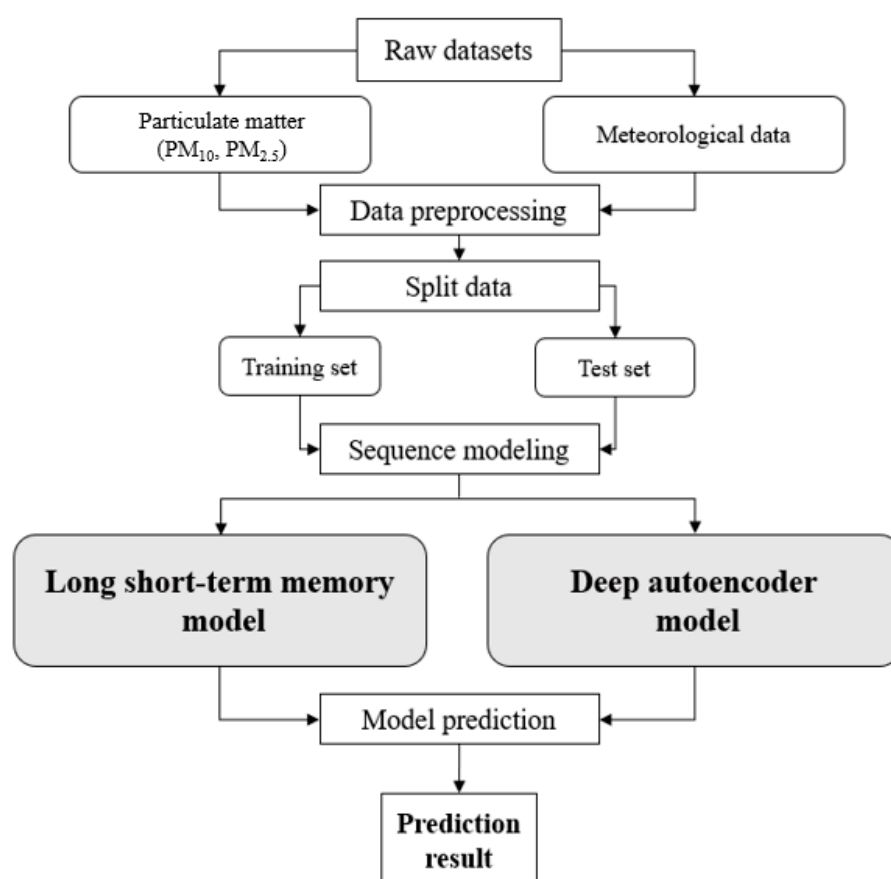


Figure 14. Workflow for predicting fine PM concentrations using the LSTM and DAE models.

5.2. LSTM Model Performance

The optimal settings for the LSTM model for both PM₁₀ and PM_{2.5} prediction were a learning rate of 0.01, epoch of 100, and batch size of 32. For a batch size of 32, the RMSE values were 11.113 for PM₁₀ and 12.174 for PM_{2.5}, with a processing time of 11:18 min (Figure 15, Table 4).

Table 4. LSTM model performance for predicting PM₁₀ and PM_{2.5} concentrations. Optimal root mean square error (RMSE) values are indicated in bold.

Batch Size	Learning Rate	Epoch	RMSE		Processing Time (Min)
			PM ₁₀	PM _{2.5}	
32	0.01	100	11.113	12.174	11:18
64	0.01	100	11.163	12.237	17:05
128	0.01	100	11.139	12.243	23:57
256	0.01	100	11.228	11.642	38:18

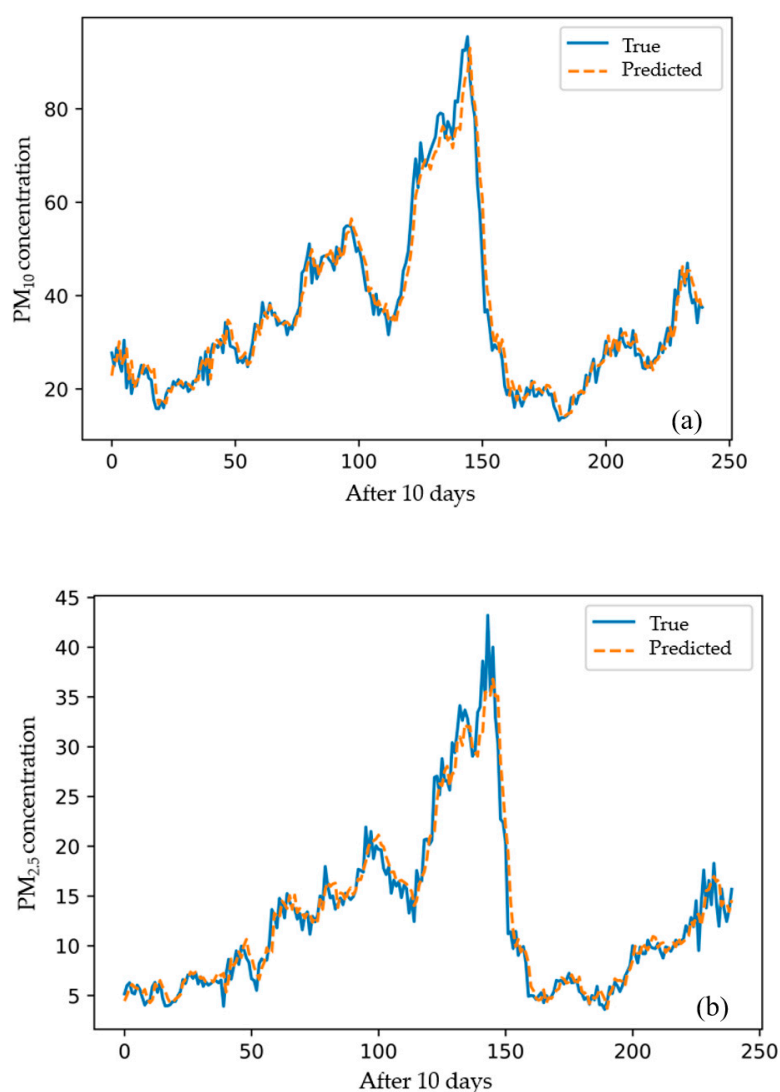


Figure 15. (a) Comparison of observed and 10-day predicted concentrations of (a) PM_{10} and (b) $PM_{2.5}$ obtained using the LSTM model.

5.3. DAE Model Performance

The optimal settings for the DAE model for both PM_{10} and $PM_{2.5}$ prediction were a learning rate of 0.01, epoch of 100, and batch size of 64. For a batch size of 64, the RMSE values were 15.038 for PM_{10} and 15.437 for $PM_{2.5}$, with a processing time of 15:40 min (Figure 16, Table 5).

Table 5. DAE model performance for predicting PM_{10} and $PM_{2.5}$ concentrations. Optimal RMSE values are indicated in bold.

Batch Size	Learning Rate	Epoch	RMSE		Processing Time (Min)
			PM_{10}	$PM_{2.5}$	
32	0.01	100	15.644	17.493	11:50
64	0.01	100	15.038	15.437	15:40
128	0.01	100	16.024	15.711	24:05
256	0.01	100	16.825	17.473	35:58

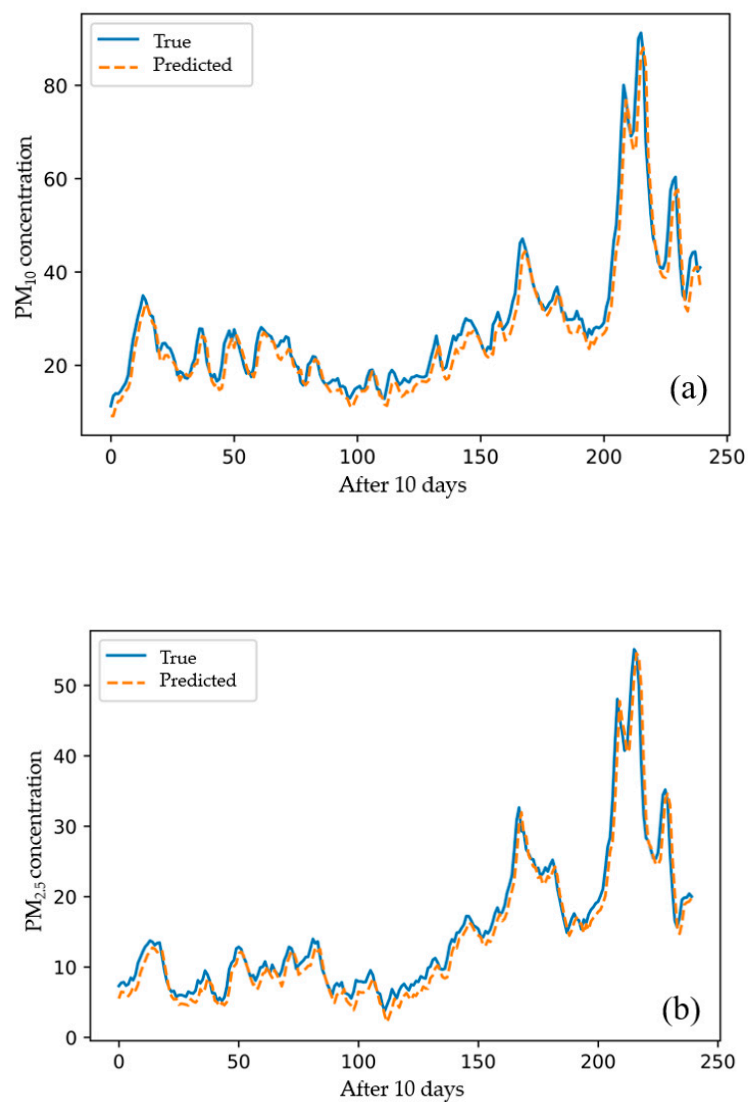


Figure 16. (a) Comparison of observed and 10-day predicted concentrations of (a) PM₁₀ and (b) PM_{2.5} obtained using the DAE model.

We used total average RMSE values to compare the results obtained using the LSTM and DAE models. Although both proposed algorithms effectively predicted PM₁₀ (Figure 17) and PM_{2.5} (Figure 18) concentrations, the LSTM model showed slightly better performance.

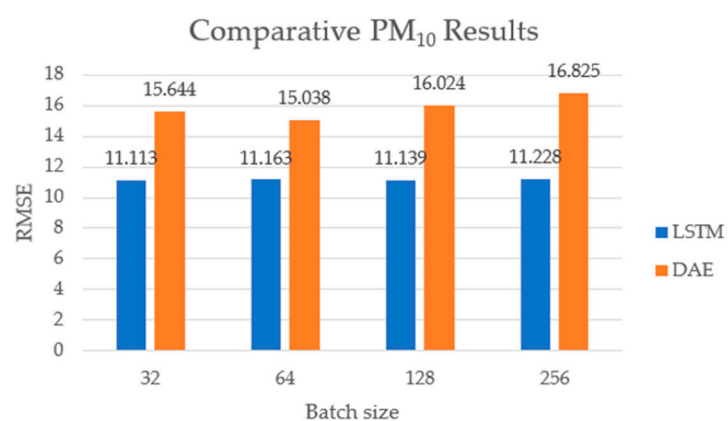


Figure 17. Comparison of LSTM and DAE model predictions of PM₁₀ concentration.

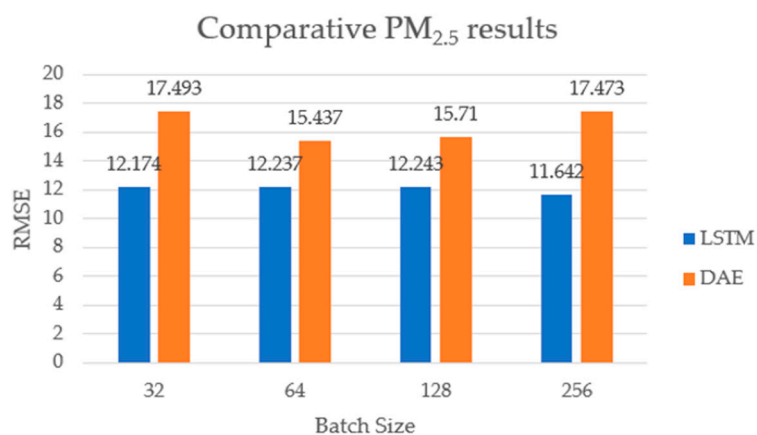


Figure 18. Comparison of LSTM and DAE model predictions of PM_{2.5} concentration.

6. Conclusions

Recent advances in the development of deep learning models have led to a rapid increase in their application in academic and industrial settings. In South Korea, the greatest environmental concern is air pollution in the form of fine PM, which consists of liquid and solid particle compounds that are dangerous to human health. Despite increasing levels of air pollutants in South Korea, the number of measurement stations remains insufficient to obtain accurate PM levels throughout the country. In this study, we proposed predictive models of fine PM concentration using LSTM and DAE approaches, and compared their RMSE values for 10-day PM₁₀ and PM_{2.5} concentration prediction results for Seoul. The principal contributions of this study are as follows: (1) According the experimental results, we have optimized the LSTM and DAE model with a learning rate of 0.01, epoch of 100, and batch sizes of 32, 64, 128, and 256. For the prediction result, the LSTM model had minimum RMSE values of 11.113 for PM₁₀ and 12.174 for PM_{2.5} at a batch size of 32. At the same time, the DAEs model had minimum RMSE values of 15.038 for PM₁₀ and 15.431 for PM_{2.5} at a batch size of 64. (2) We also compared the total average RMSE of prediction of PM₁₀ and PM_{2.5}, the LSTM prediction model were more accurate than the DAE model. The comparison showed that our proposed algorithm can predict and receive the appropriate accuracy between LSTM and DAE model. In the future, we will design alternative deep learning models to obtain more accurate results with larger data sets. We will also improve our model's performance by considering GIS-based spatial data.

Author Contributions: T.X. and H.L. conceived and designed the experiments, analyzed the data and wrote the paper. G.L. supervised the work and helped with designing the conceptual framework, and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2020-2015-0-00403) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation) and by Soonchunhyang Research Fund.

Acknowledgments: We appreciate the air quality indices data provide by the Korean Ministry of Environment (<http://www.airkorea.or.kr/>).

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

References

1. Jung, W. South Korea's Air Pollution: Gasping for Solutions. Available online: <http://isdsp.eu/publication/south-koreas-air-pollution-gasping-solutions/> (accessed on 6 April 2019).
2. Jin, L.; Luo, X.; Fu, P.; Li, X.-D. Airborne particulate matter pollution in urban China: a chemical mixture perspective from sources to impacts. *Natl. Sci. Rev.* **2016**, *4*, 593–610. [CrossRef]
3. Xing, Y.-F.; Xu, Y.-H.; Shi, M.-H.; Lian, Y.-X. The impact of PM_{2.5} on the human respiratory system. *J. Thorac. Dis.* **2016**, *8*, E69–E74. [PubMed]

4. Torrisi, M.; Pollastri, G.; Le, Q. Deep learning methods in protein structure prediction. *Comput. Struct. Biotechnol. J.* **2020**, *521*, 436–444. [\[CrossRef\]](#)
5. Heaton, J. *Deep Learning and Neural Networks*; Heaton Research Inc: Washington, DC, USA, 2015.
6. Deng, L.; Yu, D. Deep Learning: Methods and Applications. *Found. Trends Signal Process.* **2014**, *7*, 197–387. [\[CrossRef\]](#)
7. Ordieres-Meré, J.; Vergara, E.; Capuz-Rizo, S.F.; Salazar, R. 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.* **2005**, *20*, 547–559. [\[CrossRef\]](#)
8. Barai, S.V.; Dikshit, A.K.; Sharma, S. Neural Network Models for Air Quality Prediction: A Comparative Study. In *Computational Intelligence in Security for Information Systems*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2007; Volume 39, pp. 290–305.
9. Zhou, Q.; Jiang, H.; Wang, J.; Zhou, J. A hybrid model for PM_{2.5} forecasting based on ensemble empirical mode decomposition and a general regression neural network. *Sci. Total. Environ.* **2014**, *496*, 264–274. [\[CrossRef\]](#)
10. Elangasinghe, M.; Singhal, N.; Dirks, K.; Salmond, J.; Samarasinghe, S. Complex time series analysis of PM₁₀ and PM_{2.5} for a coastal site using artificial neural network modelling and k-means clustering. *Atmospheric Environ.* **2014**, *94*, 106–116. [\[CrossRef\]](#)
11. Russo, A.; Raischel, F.; Lind, P.G. Air quality prediction using optimal neural networks with stochastic variables. *Atmos. Environ.* **2013**, *79*, 822–830. [\[CrossRef\]](#)
12. Hu, X.; Waller, L.A.; Lyapustin, A.; Wang, Y.; Al-Hamdan, M.Z.; Crosson, W.L.; Estes, M.G., Jr.; Estes, S.M.; Quattrochi, D.; Puttaswamy, S.J.; et al. Estimating ground-level PM_{2.5} concentrations in the Southeastern United States using MAIAC AOD retrievals and a two-stage model. *Remote. Sens. Environ.* **2014**, *140*, 220–232. [\[CrossRef\]](#)
13. Chang, Y.-S.; Lin, K.-M.; Tsai, Y.-T.; Zeng, Y.-R.; Hung, C.-X. Big data platform for air quality analysis and prediction. In Proceedings of the 2018 27th Wireless and Optical Communication Conference (WOCC), Hualien, Taiwan, 30 April–1 May 2018; pp. 1–3.
14. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks* **2015**, *61*, 85–117. [\[CrossRef\]](#)
15. Kalapanidas, E.; Avouris, N. Short-term air quality prediction using a case-based classifier. *Environ. Model. Softw.* **2001**, *16*, 263–272. [\[CrossRef\]](#)
16. Athanasiadis, I.N.; Kaburlasos, V.G.; Mitkas, P.A.; Petridis, V. Applying machine learning techniques on air quality data for real-time decision support. In Proceedings of the First international NAISO symposium on information technologies in environmental engineering (ITEE'2003), Gdansk, Poland, 24–27 June 2003.
17. Famoso, F.; Wilson, J.; Monforte, P.; Lanzafame, R.; Brusca, S.; Lulla, V. Measurement and modeling of ground-level ozone concentration in Catania, Italy using biophysical remote sensing and GIS. *Int. J. Appl. Eng. Res.* **2017**, *12*, 10551–10562.
18. Hoek, G.; Beelen, R.; de Hoogh, K.; Vienneau, D.; Gulliver, J.; Fischer, P.; Briggs, D. A review of land-use regression models to assess spatial variation of outdoor air pollution. *Atmos. Environ.* **2008**, *42*, 7561–7578. [\[CrossRef\]](#)
19. Lee, J.-H.; Wu, C.-F.; Hoek, G.; De Hoogh, K.; Beelen, R.; Brunekreef, B.; Chan, C.-C. Land use regression models for estimating individual NO_x and NO₂ exposures in a metropolis with a high density of traffic roads and population. *Sci. Total. Environ.* **2014**, *472*, 1163–1171. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Qi, Y.; Li, Q.; Karimian, H.; Liu, D. A hybrid model for spatiotemporal forecasting of PM_{2.5} based on graph convolutional neural network and long short-term memory. *Sci. Total. Environ.* **2019**, *664*, 1–10. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Singh, K.P.; Gupta, S.; Rai, P. Identifying pollution sources and predicting urban air quality using ensemble learning methods. *Atmos. Environ.* **2013**, *80*, 426–437. [\[CrossRef\]](#)
22. Corani, G. Air quality prediction in Milan: feed-forward neural networks, pruned neural networks and lazy learning. *Ecol. Model.* **2005**, *185*, 513–529. [\[CrossRef\]](#)
23. Fu, M.; Wang, W.; Le, Z.; Khorram, M.S. Prediction of particular matter concentrations by developed feed-forward neural network with rolling mechanism and gray model. *Neural Comput. Appl.* **2015**, *26*, 1789–1797. [\[CrossRef\]](#)
24. Jiang, D.; Zhang, Y.; Hu, X.; Zeng, Y.; Tan, J.; Shao, D. Progress in developing an ANN model for air pollution index forecast. *Atmos. Environ.* **2004**, *38*, 7055–7064. [\[CrossRef\]](#)

25. Qin, D.; Yu, J.; Zou, G.; Yong, R.; Zhao, Q.; Zhang, B. A Novel Combined Prediction Scheme Based on CNN and LSTM for Urban PM2.5 Concentration. *IEEE Access* **2019**, *7*, 20050–20059. [CrossRef]
26. Bui, T.-C.; Le, V.-D.; Cha, S.-K. A Deep Learning Approach for Forecasting Air Pollution in South Korea Using LSTM 2018. *arXiv* **2018**, arXiv:1804.07891.
27. Zhao, J.; Deng, F.; Cai, Y.; Chen, J. Long short-term memory—Fully connected (LSTM-FC) neural network for PM2.5 concentration prediction. *Chemosphere* **2019**, *220*, 486–492. [CrossRef] [PubMed]
28. Reddy, V.; Yedavalli, P.; Mohanty, S.; Nakhat, U. Deep Air: Forecasting Air Pollution in Beijing, China. *arXiv* **2018**.
29. Kim, S.; Lee, J.M.; Lee, J.; Seo, J. Deep-dust: Predicting concentrations of fine dust in Seoul using LSTM 2019. *arXiv* **2019**.
30. 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]
31. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F.-Y. Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE Trans. Intell. Transp. Syst.* **2014**, *16*, 1–9. [CrossRef]
32. Huang, W.; Song, G.; Hong, H.; Xie, K. Deep Architecture for Traffic Flow Prediction: Deep Belief Networks With Multitask Learning. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2191–2201. [CrossRef]
33. Teng, Y.; Huang, X.; Ye, S.; Li, Y. Prediction of particulate matter concentration in Chengdu based on improved differential evolution algorithm and BP neural network model. In Proceedings of the 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Institute of Electrical and Electronics Engineers (IEEE), Chengdu, China, 20–22 April 2018; pp. 100–106.
34. Dong, Y.; Wang, H.; Zhang, L.; Zhang, K. An improved model for PM2.5 inference based on support vector machine. In Proceedings of the 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Institute of Electrical and Electronics Engineers (IEEE), Shanghai, China, 30 May–1 June 2016; pp. 27–31.
35. Air Korea. Available online: <http://www.airkorea.or.kr/web> (accessed on 6 April 2019).
36. Korea Meteorological Agency. Available online: <https://data.kma.go.kr/cmmn/main.do> (accessed on 6 April 2019).
37. Mahata, S.K.; Das, D.; Bandyopadhyay, S. MTIL2017: Machine Translation Using Recurrent Neural Network on Statistical Machine Translation. *J. Intell. Syst.* **2019**, *28*, 447–453. [CrossRef]
38. Wang, Q.; Lin, J.; Yuan, Y. Salient Band Selection for Hyperspectral Image Classification via Manifold Ranking. *IEEE Trans. Neural Networks Learn. Syst.* **2016**, *27*, 1279–1289. [CrossRef]
39. Graves, A.; Mohamed, A.-R.; Hinton, G.; Graves, A. Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Institute of Electrical and Electronics Engineers (IEEE), Vancouver, BC, Canada, 26–31 May 2013; pp. 6645–6649.
40. Milan, A.; Rezatofghi, S.H.; Dick, A.; Reid, I.; Schindler, K. Online Multi-Target Tracking Using Recurrent Neural Networks. *arXiv* **2016**, *1604*, 03635.
41. Liu, T.; Wu, T.; Wang, M.; Fu, M.; Kang, J.; Zhang, H. Recurrent Neural Networks based on LSTM for Predicting Geomagnetic Field. In Proceedings of the 2018 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES), Institute of Electrical and Electronics Engineers (IEEE), Bali, Indonesia, 20–21 September 2018; pp. 1–5.
42. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, *1412*, 3555.
43. Fan, J.; Li, Q.; Hou, J.; Feng, X.; Karimian, H.; Lin, S. A Spatiotemporal Prediction Framework for Air Pollution Based on Deep RNN. *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2017**, *4*, 15–22. [CrossRef]
44. Xu, G.; Fang, W. Shape retrieval using deep autoencoder learning representation. In Proceedings of the 2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Institute of Electrical and Electronics Engineers (IEEE), Chengdu, China, 16–18 December 2016; pp. 227–230.
45. Zhao, X.; Nutter, B. Content Based Image Retrieval system using Wavelet Transformation and multiple input multiple task Deep Autoencoder. In Proceedings of the 2016 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI), Santa Fe, NM, USA, 6–8 March 2016; pp. 97–100.

