Article

Short-Term Prediction of PM_{2.5} Using LSTM Deep Learning Methods

Endah Kristiani ^{1,2} , Hao Lin ¹, Jwu-Rong Lin ³ , Yen-Hsun Chuang ³ , Chin-Yin Huang ⁴
and Chao-Tung Yang ^{1,5,*}

¹ Department of Computer Science, Tunghai University, Taichung City 407224, Taiwan; endahkristi@thu.edu.tw (E.K.); mike840724@gmail.com (H.L.)

² Department of Informatics, Krida Wacana Christian University, Jakarta 11470, Indonesia

³ Department of International Business, Tunghai University, Taichung City 407224, Taiwan; jrlin@thu.edu.tw (J.-R.L.); yhchuang@thu.edu.tw (Y.-H.C.)

⁴ Department of Industrial Engineering and Enterprise Information, Tunghai University, Taichung City 407224, Taiwan; huangcy@thu.edu.tw

⁵ Research Center for Smart Sustainable Circular Economy, Tunghai University, No. 1727, Sec. 4, Taiwan Boulevard, Taichung City 407224, Taiwan

* Correspondence: ctyang@thu.edu.tw

Abstract: This paper implements deep learning methods of recurrent neural networks and short-term memory models. Two kinds of time-series data were used: air pollutant factors, such as O₃, SO₂, and CO₂ from 2017 to 2019, and meteorological factors such as temperature, humidity, wind direction, and wind speed. A trained model was used to predict air pollution within an eight-hour period. Correlation analysis was applied using Pearson and Spearman correlation coefficients. The KNN method was used to fill in the missing values to improve the generated model's accuracy. The average absolute error percentage value was used in the experiments to evaluate the model's performance. LSTM had the lowest RMSE value at 1.9 than the other models from the experiments. CNN had a significant RMSE value at 3.5, followed by Bi-LSTM at 2.5 and Bi-GRU at 2.7. In comparison, the RNN was slightly higher than LSTM at a 2.4 value.

Keywords: PM_{2.5} prediction; deep learning; air pollution; particle pollution; particulate matter forecasting; fine aerosol

Citation: Kristiani, E.; Lin, H.; Lin, J.-R.; Chuang, Y.-H.; Huang, C.-Y.; Yang, C.-T. Short-Term Prediction of PM_{2.5} Using LSTM Deep Learning Methods. *Sustainability* **2022**, *14*, 2068. <https://doi.org/10.3390/su14042068>

Academic Editor: Diego Pablo Ruiz Padillo

Received: 21 December 2021

Accepted: 31 January 2022

Published: 11 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

There are many reasons for air pollution, such as factory exhaust, steam locomotives, forest fires, and household polluting fuels. In this case, air pollution has a substantial health impact on humans. Although there are many observation instruments on the market, future air quality estimation is needed so that preventive actions can be taken before it becomes more harmful to physical health. Throughout Taiwan, the government already has hundreds of observation instruments. Every hour, the air information obtained from these observation instruments returns to a database for public usage. Here we used PM_{2.5} as a focus of observation to judge the impact of air pollution. Many air pollution factors affect PM_{2.5}. The air quality information collected in this experiment includes TEMP, CO, NO, NO₂, NO_x, O₃, PM₁₀, PM_{2.5}, and SO₂. We selected relevant air pollutant factors in the dataset. In addition, it is not only chemical reactions that affect the PM_{2.5} concentration; the environmental impact also needs to be considered. Therefore, in the experiment, we also added relevant meteorological factors in the experiments to improve prediction accuracy and generalization ability [1].

In past experiments, we used RNN to predict PM_{2.5} [2]. However, during the training process for RNN, gradient explosion and gradient disappearance occurred. As a result, the accuracy of the model dropped significantly [3]. Therefore, in this study, LSTM was

implemented to train the machine [4]. LSTM can improve the gradient disappearance and gradient explosion problems encountered by an RNN. We expect that multiple air pollution factors can be used to predict the air quality of PM_{2.5} for many hours in the future. If the accuracy of PM_{2.5} prediction can improve significantly, the prediction results could be combined with an early warning system. In this way, the air quality data can be provided as a reference. In addition, along with LSTM, a CNN was also applied for machine learning building and training. Both methods have been compared for their performance [5–7]. In this case, we found that LSTM is more suitable for processing time-series datasets and that a CNN has more advantages in feature captures [8,9]. However, as mentioned above, there is a need to increase air pollution factors. The selection of these factors will be crucial, as it significantly affects the accuracy of machine learning models. Therefore, correlation measurement methods were used to provide a strong reference for the selection of air pollution factors [10,11].

The obtained dataset is also an essential part of improving accuracy. In the collected dataset, there are missing values. There are many methods to handle missing values, such as using zero paddings, average value, and linearity [12]. In this experiment, we used two methods: KNN and linear interpolation. The results were compared to determine which method is more suitable for this air pollution prediction model. This paper proposes a short-term prediction of PM_{2.5} using LSTM deep learning methods. The contribution of this paper is that we implemented optimization and comparison methods in each machine learning process: data preprocessing, training, and visualization. Therefore, the objectives of this paper are listed as follows:

- to compare CNN, RNN, LSTM, GRU, bi-LSTM, and bi-GRU models to predict PM_{2.5};
- to compare linear interpolation and KNN to handle missing values;
- to compare Pearson and Spearman correlations and principal component analysis;
- to compare two datasets of chemical versus chemical and meteorological factors;
- to identify the best correlation period of air pollutant and PM_{2.5} in each time lag.

2. Background Review

Before conducting this study, we consulted other related research results, including their theoretical basis, experimental ideas, and practical processes. These results provided a more definite concept, which helped us to avoid difficulties and obtain improved results.

2.1. Data Preprocessing

Xingjian Shi et al. [13] used two methods to complement values, namely KNN and SVM. These two methods are suitable for different situations, and the performance is excellent. The article shows the importance of missing value filling. Regardless of the type of data and research, the impact of missing values on the experiment is enormous. That paper shows that the principle of KNN used for the missing value complement is a triable method. Therefore, in this study, experiments on missing value complementation were implemented, and KNN was used as a supplementary value method.

Wu and Zhang [14] built an automatic encoder in the model, mainly to reduce the dimensionality and reduce the noise of the Jiang data. Moreover, it can compress the original data learned from the input layer, and the decoding matches the original data. This method has proved to be immensely helpful. Therefore, this data processing method was applied to our research. In addition to providing us with an idea of data preprocessing, we used bidirectional LSTM for modeling and obtained excellent results in the evaluation of RMSE.

Zhang et al. [15] proposed a PM_{2.5} prediction model based on auto-encoder bidirectional LSTM to highlight the link between PM_{2.5} and climate variables. The model included preprocessing data, an auto-encoder, and a Bi-LSTM layer. A real-world dataset was used to verify the suggested model's performance, which may increase the predictive accuracy in an experimental situation.

Salcedo-Sanz et al. [16] generated a thorough review of the latest information fusion work performed for Earth observation. The practical intention was to describe the most relevant works in the field and the essential applications for Earth observation, where ML information fusion has achieved significant results. They also evaluated some of the most commonly used datasets, models, and sources for problems in Earth observation, describing their importance and how the data can be obtained when needed. Therefore, in our experiments, we conducted more research on data preprocessing and used different kinds of neural networks to build a model.

2.2. Correlation Analysis

Jing Chen et al. [17] used historical data of Changsha in 2018. They used the method of correlation coefficient measurement. Correlation experiments were conducted on chemical factors related to $PM_{2.5}$, including AQI, $PM_{2.5}$, PM_{10} , SO_2 , and NO_2 . The experimental results of the correlation coefficient show that the correlation between $PM_{2.5}$ and PM_{10} , SO_2 , and CO is strong. In contrast, its correlation with O_3 is weak. After using the selected factors as training variables, the prediction results obtained were quite good. Therefore, in our study, the model training selects the variable part. We used two methods of correlation measurement to measure the correlation between the variables in our dataset and, for subsequent experiments, to understand the correlation between $PM_{2.5}$ and other factors. It is hoped that this will improve the accuracy of the prediction model.

Wen Yan Yuan et al. [18] proposed multi-factor and multi-scale analysis and correlation analysis for different variable factors. The analysis methods used in this method include the Pearson correlation coefficient. The multi-scale analysis will show the impact of meteorological factors on $PM_{2.5}$ at different times.

Therefore, this paper provides more ideas about the correlation between related factors and multi-scale analysis. However, it also reminds us that the relationship between time and factors will change, which also needs attention and discussion.

2.3. Deep Learning Model

Xiang Li et al. [19] used LSTM to predict $PM_{2.5}$ and discussed the prediction effect comparison between LSTM and other regression methods. In this prediction, it not only used chemical factors as models, but the training variables also included meteorological factors as auxiliary data, such as temperature, humidity, and wind direction. On the other hand, this paper's prediction model uses MAPE, whose evaluation result value is 11, a more accurate prediction performance. Therefore, in our research, we conducted modeling and an evaluation of whether or not to add meteorological factors and adjust the subsequent model construction according to the experimental results.

Yi Ting Tsai et al. [20] used two neural network construction models to predict $PM_{2.5}$. These two neural networks are RNN and LSTM, which the Taiwan Environmental Protection Agency provides. The historical data were from 2012 to 2016. The prediction time was four hours in the future. The evaluation model method uses RMSE. In this paper's experimental results, it can be seen that the RMSE performance of LSTM is better than that of the RNN. The prediction performance of various regions is also quite good. Therefore, in our experiments, we implemented different neural network models for comparison, including CNN, RNN, LSTM, and GRU, to understand what kind of neural network model is more suitable for this kind of issue.

Jun Ma et al. [21] used a new deep learning hybrid model. This model is a hybrid model of BLSTM and IDW. BLSTM can effectively capture the time series of data features of air pollution. At the same time, with the IDW layer, it is possible to consider the spatial correlation of air pollution and interpolate the spatial distribution. According to the paper's experimental results, the evaluation values presented by their proposed models perform exceptionally well, and the MAPE evaluation value is 9. Therefore, in our study, the IDW mechanism was used to perform diffusion calculations on $PM_{2.5}$. Based

on its characteristics, we estimated the areas without air quality monitoring stations to understand the air quality status in the area.

Kim et al. [22] deployed daily PM_{10} and $PM_{2.5}$ predictions using LSTM in South Korea. They compared the performance with ground PM observations from a chemistry-transport model (CTM) simulations. The results indicated that the index of agreements (IOA) was increased from 0.36 to 0.78 and, with the 3-D CTM simulations, from 0.62 to 0.79 with the LSTM-based model.

Ayturan et al. [23] modeled a $PM_{2.5}$ short-term prediction of 1, 2, and 3 h using three deep learning methods, LSTM, RNN, and GRU, in Turkey. The combination of RNN and GRU achieved the best result with an R^2 of 0.83, 0.7, and 0.63 for 1, 2, and 3 h predictions.

Li et al. [24] proposed the 24 h forecasting of $PM_{2.5}$ based on an attention-based CNN-LSTM model. They compared seven models: SVR, RFR, MLP, Simple RNN, LSTM, CNN-LSTM, and AC LSTM. The results indicate that the AC-LSTM model improved the performance in the multi-scale prediction tasks. The RMSE value was the lowest at 14.83 in the 13–24 h measurement compared to the other models.

Wu et al. [25] demonstrated the prediction of PM_{10} and $PM_{2.5}$ using composite methods. The dataset was extracted based on the Moderate Resolution Imaging Spectroradiometer (MODIS) and the dense dark vegetation (DDV) methods. The aerosol optical depth (AOD) was optimized using the calculation of the planetary boundary layer height (PBLH) and relative humidity (RH). The highest correlation with PM_{10} and $PM_{2.5}$ was selected. The LSTM model showed high performance, based on the average, maximum, and minimum accuracy and the stability of $PM_{10}/PM_{2.5}$ prediction.

Yang et al. [26] presented a hybrid model of CNN-LSTM and CNN-GRU in predicting PM_{10} and $PM_{2.5}$ in Seoul, South Korea, for seven days. They compared single and hybrid models. Their experimental results show that hybrid models outperform single models.

Xayasoux et al. [27] built models for predicting PM concentrations using LSTM and deep autoencoder (DAE) methods and compared the RMSE model outputs. From 1 January 2015 to 31 December 2018, they applied the models to hourly air quality data from 25 stations in Seoul, South Korea. For the 10 days that followed this time, PM concentrations were predicted at an optimal learning rate of 0.01, 100 epochs, and batch sizes of 32 for the LSTM model. The DAE model performed best with a batch size of 64. The proposed models effectively predicted concentrations of fine PM, with the LSTM model showing much better results.

Kow et al. [28] proposed a hybrid model (CNN-BP) involving a CNN and a back propagation neural network (BPNN) that would simultaneously make accurate $PM_{2.5}$ predictions for multiple stations at different horizons. The case study was formed by the hourly dataset of six air quality factors and two meteorological factors gathered during 2017 from 73 air quality monitoring stations in Taiwan. The results show that the model proposed for CNN-BP results in a higher forecast precision and is excellent at generating accurate regional multi-step forward $PM_{2.5}$ predictions. The BPNN, random forest, and long-term memory neural network models are remarkably superior.

Lin et al. [29] discussed a new method of data fusion called the space–time multi-sensor data fusion framework. It is based on the optimum linear data fusion theory. It integrates the spatial–temporal estimation with a multi-time step Kriging method. Findings showed that by integrating $PM_{2.5}$ concentration data from 1176 low-cost AirBoxes as additional information in their model, estimating the concentration of $PM_{2.5}$ spatiotemporally becomes more accessible and more realistic. The regression model r^2 for the validation was 0.89.

Chen et al. [30] proposed the EPLS model-based method of feature extraction. Their models contain dimension reduction, mode decomposition, and least squares projection. For a synthetic dataset and air quality data collection, it uses various clustering and similarity measurement methods. Experimental findings show that EPLS successfully deals with high noise and outer air quality clustering problems and can also be adjusted to various clustering strategies and distance measurements.

2.4. The State of the Art (SOTA)

Training models of PM_{2.5} have been studied. By analyzing the processing ability and analytical ability of several predictors, Liu et al. [31] presented the Q-learning method of the GCN-LSTM-GRU-Q model. The best RMSE model performance was 17.6366. Yan et al. [32] built EntityDenseNet, a new deep learning model for retrieving ground-level PM_{2.5} concentrations from Himawari-8 data. The model has the capacity to extract PM_{2.5} spatiotemporal features automatically. Validation on all of Mainland China showed that the root-mean-square errors of hourly, daily, and monthly PM_{2.5} retrievals were 26.85, 25.3, and 15.34 g/m³, respectively. For forecasting 24 h of PM_{2.5}, Li et al. [33] used a hybrid CNN-LSTM model that combined the CNN with an LSTM neural network. With a score of 17.9306, the RMSE of the multivariate CNN-LSTM model was the lowest. Xiao et al. [34] suggested a weighted LSTM neural network extended model (WLSTME) to solve the problem of how to account for the effect of site density and wind conditions on the spatiotemporal correlation of air pollution concentrations. Their proposed model had the lowest RMSE (40.67) and MAE (26.10) values, as well as the highest *p*-value (0.59). Table 1 shows the state-of-the-art (SOTA) approaches related to the PM_{2.5} model's performance based on RMSE.

Table 1. State-of-the-art (SOTA) approach comparison.

Author	Model	RMSE
Liu et al., 2021	GCN-LSTM-GRU-Q	17.6366
Yan et al., 2020	EntityDenseNet	15.34
Li et al., 2020	multivariate CNN-LSTM	17.9306
Xiao et al., 2020	WLSTME	40.67
Our model	empirical study of LSTM	1.9

3. Material and Methods

First, this system collects various air pollution indicators and stores them in the database. The data are then adjusted through various preprocessing methods, and the data are adapted to be appropriate for machine learning models. Different machine learning models are deployed for experiments based on multiple neural networks. Finally, the data is adjusted to the proper air-pollution prediction model in each region and added to the system for prediction. Next, prediction results are presented in graphs and maps to indicate the future air pollution level over the short term.

3.1. Research Procedures

The air pollutant and meteorological factor data were stored in the database from the air quality monitoring station to facilitate subsequent experiments. Data preprocessing first conducts correlation experiments on each variable as a reference for training the model. Missing values are then found by filling, standardization, data cutting, and one-hot encoding the data required in our experiment. For model training, the prediction target set in this study was PM_{2.5}, and the prediction time was 8 h. The algorithm used was a long short-term memory method, and the evaluation model was MAPE and RMSE. After the data prediction was completed, the data were stored in the database for diffusion calculation and displayed on the web page visually, such as by a line chart, to easily monitor and refer to it. Figure 1 describes the research procedure of this paper.

Figure 1. Research process.

3.2. Taiwan Demographics

The total population in persons was 23,603,121 in 2019, and the population density was 652.07 persons per square kilometer [35]. The mean temperature range in 1991–2020 was 4.4 to 25.5 °C according to their areas. The relative humidity range was 73.6% to 89.6%. The precipitation range was 1097.1 to 4697.1 mm. The sunshine duration range was 956.4 to 2281.8 h [36]. Taiwan's gross domestic product per capita reached USD 25,909 in 2019 [37]. In 2019, there were 8,118,885 cars and 13,992,922 motorcycles, with 610.87 per square kilometer. There were 90,424 factories, with 2.50 in every square kilometer.

3.3. Dataset

The dataset in this study was extracted from a government, open data platform from the Taiwan Environmental Protection Administration (EPA) [35]. There are two options: JSON and CSV; in this case, we chose the JSON format, as shown in Figure 2.

AIR QUALITY INDICATOR (AQI)

The Air Quality Index (AQI) of each station is provided every hour, and the original data version is published on the <https://airtw.epa.gov.tw> of the Air Quality Monitoring Network

score this dataset:
3.82 (93 votes)
average 3.82 (93 votes)

hits: 111286 download times: 63491 number of views: 45

description of the main field

SiteName (station name), County (county), AQI (air quality indicator), Pollutant, Status, SO2 (sulfur dioxide (ppb)), CO (carbon monoxide (ppm)), CO_8hr (carbon monoxide 8-hour moving average (ppm)), O3 (Ozone (ppb)), O3_8hr (Ozone 8-hour Moving Average (ppb)), PM10 (Suspended Particles (µg/m3)), PM2.5 (fine suspension particles (µg/m3)), NO2 (nitrogen dioxide (ppb)), NOx (nitrogen oxides (ppb)), NO (nitric oxide (ppb)), WIND_SPEED (wind speed (m/sec)), WIND_DIRC (degrees), PublishTime, PM2.5_AVG (moving average of finely suspended particles (µg/m3)), PM10_AVG (Moving Average of Suspended Particles (µg/m3)), SO2_AVG (Moving Average of Sulfur Dioxide (ppb)), Longitude (Longitude), Latitude (Latitude), SiteId (station number).

information resource download url

[JSON](#) view information AIR QUALITY INDICATOR (AQI)-JSON
[CSV](#) view information AIR QUALITY INDEX (AQI)-CSV

Figure 2. Data extraction from EPA.

For further experiments on machine learning, the data were stored in a MySQL database. Figure 3 describes the raw data in this database.

SiteName	County	AQI	Pollutant	Status	SO2	CO	CO_8hr	O3	O3_8hr	PM10	PM25	NO2	NOx	NO	WindSpeed	WindDirC	PublishTime	PM10_AVG	PM25_AVG
Xitun	Taichung	146	Fine Aerosol	Very Unhealthy	2.5	1.09	1.1	3.2	5	87	56	29	71	41	0.5	200	2017-01-08 03:00	98	53
Xitun	Taichung	145	Fine Aerosol	Very Unhealthy	1.7	0.8	1.1	3.1	4	73	48	24	39	15	0.5	193	2017-01-08 04:00	94	53
Xitun	Taichung	140	Fine Aerosol	Very Unhealthy	1.7	0.8	1.1	2.9	3	65	38	23	49	26	0.5	13	2017-01-08 05:00	89	50
Xitun	Taichung	130	Fine Aerosol	Very Unhealthy	1.6	0.64	1.1	3.1	3	47	32	21	42	21	0.5	14	2017-01-08 06:00	80	47
Xitun	Taichung	120	Fine Aerosol	Very Unhealthy	1.2	0.69	1	3.5	3	44	30	19	41	23	0.5	120	2017-01-08 07:00	73	43
Xitun	Taichung	111	Fine Aerosol	Very Unhealthy	1.3	0.49	9	6.4	4	36	27	16	38	20	1.8	47	2017-01-08 08:00	65	39
Xitun	Taichung	104	Fine Aerosol	Very Unhealthy	1.8	0.32	8	22	6	38	26	13	20	6.9	2.2	36	2017-01-08 09:00	59	37
Xitun	Taichung	99	Fine Aerosol	Very Unhealthy	2	0.28	6	27	9	41	26	12	19	7.2	3.2	327	2017-01-08 10:00	55	35

Figure 3. Raw AQI data in the MySQL database.

3.4. Dataset Processing

Before conducting training, an appropriate dataset is needed. When the machine performs feature learning, it relies on the existing dataset information to learn, so proper handling of the dataset is essential. Table 2 describes the distribution of stations in Taiwan. This study builds a model based on station distribution and provides a prediction function.

Table 2. Distribution of stations in Taiwan.

County	Station
North District	Keelung, Xizhi, Sanchong, Tucheng, Yonghe, Banqiao, Linkou, Tamsui, Cailiao, Xindian, Xinzhuang, Wanli, Guting, Songshan, Yangming, Wanhua, Shilin, Datong, Zhongshan, Dayuan, Zhongli, Pingzhen, Taoyuan, Longtan, Guanyin, Hsinchu, Zhudong, Dongshan, Yilan
Central District	Sanyi, Miaoli, Toufen, Fengyuan, Shalu, Dali, Zhongming, Xitun, Erlin, Changhua, Xianxi, Zhushan, Nantou, Puli, Douliu, Lunbei, Mailiao, Taixi
South District	Chiayi, Puzi, Xingang, Annan, Shanhua, Xinying, Tainan, Daliao, Xiaogang, Renwu, Zuoying, Linyuan, Qianjin, Qianzhen, Mino, Fuxing, Nanzi, Fengshan, Qiaotou, Pingtung, Hengchun, Chaozhou, Magong
Eastern District	Hualien, Taitung, Guanshan
Islands	Kinmen, Matsu

3.4.1. Data Imputation

Filling the missing values is essential. The original data might have some missing values due to such unavoidable factors as sensor damage, an unstable network connection, and electricity problems. Suppose these missing values are left in the dataset to continue the subsequent experiments. In that case, the machine learning model training's effectiveness will be inferior. It will also cause errors in the learning process. In order to avoid this situation, in this study, two different missing value imputation methods were selected: KNN and linear interpolation. The results from these two methods were then compared.

3.4.2. Extreme Value Processing

The processing of extreme values is also crucial. For example, when collecting data from sensors, some natural disasters and human factors might cause errors in the data sensors' collection. Suppose these extreme values are put into the neural network for training together with the dataset. In that case, it will render the machine learning process with the wrong features, thus resulting in poor prediction performance. Therefore, to avoid the adverse effects of extreme values on machine learning training, the extreme values of the dataset were deleted by observing each data in the dataset and setting the maximum and minimum values. If the value in the dataset exceeds the setting value, the record will be flagged.

There are some approaching methods for managing extreme values. They can be based on the general PM_{2.5} index or the sensor error of the PM_{2.5} measuring station. In this case, we refer to the sensor threshold values in certain areas or stations [38]. Therefore, the error threshold value of PM_{2.5} depends on the dataset history instead of the threshold of the PM_{2.5} index. The method used in this study is the 3σ method, which retains the data distributed in $(\mu - 3\sigma, \mu + 3\sigma)$. μ represents the average value, and σ is the standard deviation. The 3σ extreme value method is used to calculate a value that is too far from the average value, such as an extreme value. Generally, the value exceeding the standard deviation threefold is judged as an extreme value. The lower limit value is set to 0 here. As shown in Figure 4 as an example, the x-axis is the number of data, and the y-axis is the value of

PM_{2.5}. The blue line is the upper limit of the numerical value set. Here the lower limit is set to 0. Suppose the upper and lower limits are exceeded. In that case, the program will classify the data as extreme, and the green polyline will perform the lower limit data.

Figure 4. Extreme value processing of PM_{2.5} in $\mu\text{g}/\text{m}^3$.

The RMSE values of RNN, CNN, LSTM, Bi-LSTM, GRU, and Bi-GRU are 2.4702, 3.4984, 1.8643, 2.8379, 2.6398, and 2.6695, respectively. It can be seen that the LSTM model has the lowest error value among these models. Figure 5 shows a comparison of RMSE values in various models.

Figure 5. Comparison of RMSE values.

3.4.3. One-Hot Encoding

In this experiment, the data used include air chemical and meteorological factors, and the wind direction variables in the meteorological factors needed to be processed. In particular, because the wind direction has a total of 360 degrees, in order to facilitate the machine learning in the subsequent model training, we simplified the data into eight suggestions for the experiments.

In data processing and feature engineering, if this type of data is encountered, it will be converted into data and brought into the model. The model will treat these data as continuous parameters that affect the model's performance. Therefore, this experiment uses one-hot encoding to encode these values and put them into the model for training and subsequent investigations. Figure 6 shows the data type after one-hot encoding. The wind directions are the eight compass bearings: east, southeast, south, southwest, west, northwest, north, and northeast.

	East	South East	South	South West	West	North West	North	North East
0	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0
9	0	1	0	0	0	0	0	0
10	1	0	0	0	0	0	0	0

Figure 6. One-hot encoding of eight wind directions.

3.4.4. Data Segment

There are several methods for evaluating machine-learning models. First, the dataset is divided into three kinds: training, test, and validation. The procedure is to then train the model and evaluate the training model with the validation dataset. Finally, the test dataset examines the final test when the model has been trained and verified. In this way, the accuracy of the machine-learning model can be more accurate.

In this study, the dataset was divided into a training set, test set, and validation set at a ratio of 6:2:2. The test dataset that was not used in training was put into the model for prediction. The error between the actual $PM_{2.5}$ and the prediction result during this period was compared and used as one of the references for the accurate evaluation of the prediction model.

The prediction time units used the multiple air-pollution factor values of the first 8, 16, and 24 h periods to predict the model's input data. The output results were the $PM_{2.5}$ concentration data for the next 8 h of the predicted period.

3.4.5. Normalization Data

Before putting the dataset into the neural network for operation, in addition to vectorizing the dataset, the dataset must be normalized. The so-called vectorization means that all input data and labels of the neural network must be in a floating-point tensor, so regardless of what data are processed, it must be converted to a tensor first. This step is called vectorization, and the reason for normalization is to input a relatively large value into the neural network. The difference in the numerical range of the two features is too large, and the impact on the neural network is not good. It is easy for the convergence effect of the neural network to be weak, so the dataset needs to be normalized. Normalization is mainly performed to reduce the length of the data and improve the data's consistency. Therefore, in this experiment, we normalized the various air pollution factors in our dataset to accelerate the machine learning of the correct features more easily and quickly.

Table 3 represents the air quality data after it was normalized. The value range of each factor is substantially reduced, so when training the neural network, it was easier to converge. The values also had to be normalized in the subsequent prediction actions to obtain the correct prediction results.

Table 3. Normalization data.

SO ₂	O ₃	PM ₁₀	Nox	WIND SPEED	RH	PM _{2.5}
0.077273	0.303279	0.079692	0.018676	0.313953	0.645233	0.192308
0.077277	0.311475	0.075835	0.018261	0.348837	0.656319	0.192308
0.070455	0.327869	0.080977	0.014526	0.267442	0.656319	0.144231
0.065909	0.327869	0.078406	0.014526	0.337209	0.667406	0.153846

3.5. Data Correlation

Suppose we put variables with too low a correlation in the model and let the machine learn the features. In this case, it will cause the machine to capture the features less efficiently and cause the machine to learn the wrong features, resulting in a low accuracy in the entire machine learning prediction model. Therefore, it is essential to find air pollution factors that have a high correlation with $PM_{2.5}$.

This study carried out Pearson correlation coefficient and Spearman correlation coefficient experiments on the air pollution factors in the air-pollution quality dataset and $PM_{2.5}$. We then screened out air pollution factors that showed a high correlation. Simultaneously, we also conducted a principal component analysis of each air pollution factor, observed the correlation between each element, and subsequently found the appropriate factor as the input data for training the machine learning prediction model.

Figure 7 is a dotted distribution diagram of the correlation between some air pollution factors and $PM_{2.5}$. It can be seen that the correlation between $PM_{2.5}$ and SO_2 , CO, and PM_{10} is improved. Therefore, the correlation between them is high. However, the dot plot is only used as a preliminary reference and judgment. There will be more correlations of pollution factors and more detailed factors in the experimental results.

3.6. Lag Time

In the air quality dataset, there might be factors and mutual changes between $PM_{2.5}$ that are not happening immediately. It may take some delay time for the chemical reaction to change. In order to arrange this possibility, we adjusted the time delay of each air pollution factor in the dataset and conducted a correlation experiment with $PM_{2.5}$, so that air pollution factors with a high correlation performance would not miss due to the long chemical reaction time requirement.

Therefore, after adjusting the time delay of each air pollution factor, we used the Pearson correlation coefficient and Spearman correlation coefficient to conduct correlation measurement experiments on the variables of each period and $PM_{2.5}$. First, we identified the period with the best correlation with $PM_{2.5}$ in each time lag. We then adjusted the data of this factor according to the amount of time retention. After the adjustment was completed, the data were then put into the training model as input data for improved prediction results.

Figure 7. Correlation analysis of $PM_{2.5}$ and air pollutants.

3.7. Building and Training Prediction Models

The machine learning workflows need to define the problem we want to solve, create a dataset, and select the appropriate verification program, such as simple split and cross-verification methods. Next, it needs to perform necessary preprocessing actions on the data, normalization, vectorization, and missing value handling, and these actions must be completed before the training process. The machine-learning model can then begin to be built.

3.7.1. Model Building

First, we needed to select the neural layer's activation function, the model's loss function, and the optimizer settings for this machine-learning model. These settings are not random choices but need to be adjusted according to the neural network and dataset we want to use.

After setting the activation function, loss function, and optimizer, we can adjust the number of neural layers of the machine-learning model and each neural network layer's neurons. Generally, if more neural layers and neurons are added for model training, more features can be learned, but overfitting will be more likely. Therefore, it is necessary to use the conventional processing mentioned later, including the discarding method and the L1 and L2 traditional methods. This situation often occurs, and special treatment is required.

Finally, after stacking this machine learning model architecture, we need to select hyper-parameters. Each hyper-parameter represents a different meaning. Epochs represent how many times the training dataset participated in the training process. In Keras, the parameter update is carried out in batches, which is the so-called small-batch reduction algorithm. It divides the data into several groups, called batches, and batch size refers to the number of samples in each data group.

In addition, LSTM solves the problem of gradient disappearance in an RNN. However, it did not solve the problem of gradient explosion, so it must be continuously tested when writing the architecture and selecting the activation function and various hyper-parameters suitable for this data and model. The hyper-parameter will be set in the subsequent experimental process to avoid the occurrence of gradient explosion without a learning effect.

3.7.2. The Normalized Model

As mentioned in the previous section, training machine learning models will first train the model in an overfitting state and then use conventional methods to adjust the model to a satisfactory state. Overfitting occurs when the training model cannot remember the general characteristics of the training data and the local characteristics of the training data. Therefore, the prediction effect of the model is not as good.

In order to solve overfitting problems, we performed the following:

- We reduced the size of the neural network. By reducing the size of the neural network model, the number of parameters that the model uses in order to learn can reduce the number of correct features.
- We added weight normalization: This method mainly adopts a smaller weight value to limit this model's complexity and to normalize the distribution of weights. The main methods are the L1 conventional method and the L2 conventional method.
- We added a dropout method: This method mainly randomly drops some features of the neural layer during training, and the drop rate setting will generally be between 0.2 and 0.5.

3.7.3. Model Evaluation

After completing the machine-learning model's construction and training, it is required to evaluate the model to understand its effectiveness. There are many ways to assess it. The simple split method mentioned in the previous section is one of the evaluation methods. However, there is no way of knowing this model's performance only by understanding the

predicted and actual values. Therefore, several model evaluation indicators are needed as a basis for comparison. In the previous section, some model evaluation indicators were mentioned. In this study, we used MAE as a loss function. It will calculate the error value of the training model at each step and assist the machine learning. In the model's final evaluation, we compared MAE, MSE, MAPE, and RMSE as indicators for model evaluation. MAPE is one of the most commonly used evaluation indicators. It can be seen from the following formula that the MAPE is the error value divided by the actual value. Therefore, when the real value is low, and the error is significant, it will significantly impact the value of MAPE. The following is the formula for MAPE.

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{\hat{Y}_i - Y_i}{Y_i} \right| \quad (1)$$

Table 4 shows the evaluation criteria for the MAPE values.

Table 4. MAPE evaluation criteria.

MAPE Value	Ability to Predict
<10%	High accuracy
10~20%	Good accuracy
20~50%	Reasonable accuracy
>50%	Poor accuracy

MAE is a commonly used forecast evaluation index, but it does not take into account the average of the actual values. Although an evaluation value can be obtained after calculation, there is no way of knowing the model's pros and cons. We can only perform a comparison. The following is the formula for MAE.

$$\text{MAE} = \left(\frac{1}{n}\right) * \sum |y_i - x_i| \quad (2)$$

RMSE is used to measure the error between an observed value and a real value. The calculation method does not consider the actual value. Therefore, as long as there is a large error in the prediction result, the RMSE value will be inferior. The following is the formula for RMSE.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2} \quad (3)$$

MSE is the most common indicator in regression problems because it is calculated faster. It is used to measure the average squared difference between the predicted and actual values. The following is the MSE formula.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2 \quad (4)$$

3.8. Visualization

This study also presents the results of data analysis on a webpage. The webpage has two functions. Python's folium package is used to draw, and the diffused PM_{2.5} prediction results currently on the map are displayed in different colors to show the PM_{2.5} values. The prediction results of each station are also shown. It was created in HTML+JAVA+CSS3 and employs the responsive web page concept to facilitate users to watch on various devices, including smart mobile devices, tablets, and computers, to provide a good user experience.

3.9. IDW

In this study, we applied IDW to calculate the characteristics of unknown points, calculate the PM_{2.5} concentration value of the area around the entire air quality monitoring station, and establish PM_{2.5} concentration diffusion data. However, if it is too far away, this method cannot be used for estimation. The IDW formula is as follows:

$$f(x, y) = \frac{\sum_{i=1}^N w(d_i) z_i}{\sum_{i=1}^N w(d_i)} \quad (5)$$

where $w(d_i)$ is the weight equation, z_i is the value of the i th known point, and d_i is the distance between point i and the unknown point. The size of $w(d_i)$ is determined by the inverse ratio of d_i to the center.

This study first used monitoring stations within 3 km around the unknown point as a reference in the design algorithm. If there were no stations in this range, the range was expanded to 5 km. If there was still no monitoring station, the PM_{2.5} concentration at that point would be expressed as a concentration value of 0. This is because the distance estimation error would be too large due to the distance. However, if too many stations were in the range, it would be calculated as an average.

4. Experimental Results

In this section, the results based on the designed research were demonstrated. First, the correlation experiments were implemented using the Pearson correlation coefficient, the Spearman correlation coefficient, and principal component analysis to understand the correlation between air pollution factor variables. Second, the time lag was implemented to understand which period correlation was higher between PM_{2.5} and variables. Third, KNN and linear interpolation were applied to handle missing values and were compared to understand which method is more suitable in this experiment. The different neural networks were conducted to implement prediction models and compare the predictions' differences.

4.1. Data Preprocessing Experiment

4.1.1. Correlation Experiment of Input Variables

Table 5 shows the corresponding performance of the values obtained from the correlation coefficient experiment. The correlation analysis calculates the absolute value to indicate how the correlation performs. Values less than 0.3 indicate a low correlation, values above 0.3 and less than 0.7 indicate a moderate correlation, and values above 0.7 indicate a high correlation.

Table 5. Relevance evaluation criteria.

Correlation Performs	Correlation Value
low correlation	$0.0 < \text{value} < 0.3$
moderate correlation	$0.3 < \text{value} < 0.7$
high correlation	$0.7 < \text{value} < 1.0$

4.1.2. Pearson Correlation

Table 6 shows the experimental results of using the Pearson correlation coefficient to test the air quality factor. From this table, it can be seen that the correlation between PM_{2.5} and SO₂, CO, PM₁₀, NO₂, THC, NMHC, and CH₄ is higher than 0.3. This shows that the correlation performance of these factors is satisfactory. Therefore, in this study, these air pollution factors were used as a reference for this deep learning prediction model.

Table 6. Pearson correlation coefficient evaluation.

Parameters	Pearson Correlation
$\rho(\text{PM}_{2.5}, \text{SO}_2)$	0.369
$\rho(\text{PM}_{2.5}, \text{CO})$	0.453
$\rho(\text{PM}_{2.5}, \text{O}_3)$	0.059
$\rho(\text{PM}_{2.5}, \text{PM}_{10})$	0.765
$\rho(\text{PM}_{2.5}, \text{NO}_x)$	0.299
$\rho(\text{PM}_{2.5}, \text{NO})$	0.174
$\rho(\text{PM}_{2.5}, \text{NO}_2)$	0.349
$\rho(\text{PM}_{2.5}, \text{THC})$	0.379
$\rho(\text{PM}_{2.5}, \text{NMHC})$	0.379
$\rho(\text{PM}_{2.5}, \text{WS})$	−0.200
$\rho(\text{PM}_{2.5}, \text{WD})$	0.132
$\rho(\text{PM}_{2.5}, \text{TEMP})$	0.197
$\rho(\text{PM}_{2.5}, \text{RAINFALL})$	−0.143
$\rho(\text{PM}_{2.5}, \text{CH}_4)$	0.324

4.1.3. Spearman Correlation

Table 7 shows the experimental results of using the Spearman correlation coefficient to conduct correlation experiments on air quality factors. From this table, it can be seen that the correlation between $\text{PM}_{2.5}$ and SO_2 , CO , PM_{10} , NO_x , NO_2 , THC , NMHC , and CH_4 is higher than 0.3. Therefore, these air pollution factors were also used as a reference for the deep learning prediction model in this study.

Table 7. Spearman correlation coefficient evaluation.

Parameters	Spearman Correlation
$\rho(\text{PM}_{2.5}, \text{SO}_2)$	0.337
$\rho(\text{PM}_{2.5}, \text{CO})$	0.449
$\rho(\text{PM}_{2.5}, \text{O}_3)$	0.063
$\rho(\text{PM}_{2.5}, \text{PM}_{10})$	0.673
$\rho(\text{PM}_{2.5}, \text{NO}_x)$	0.320
$\rho(\text{PM}_{2.5}, \text{NO})$	0.186
$\rho(\text{PM}_{2.5}, \text{NO}_2)$	0.338
$\rho(\text{PM}_{2.5}, \text{THC})$	0.380
$\rho(\text{PM}_{2.5}, \text{NMHC})$	0.373
$\rho(\text{PM}_{2.5}, \text{WS})$	−0.171
$\rho(\text{PM}_{2.5}, \text{WD})$	0.096
$\rho(\text{PM}_{2.5}, \text{TEMP})$	0.174
$\rho(\text{PM}_{2.5}, \text{RAINFALL})$	−0.221
$\rho(\text{PM}_{2.5}, \text{CH}_4)$	0.314

In these experiments, we integrated the chemical and meteorological factors. We used the Pearson correlation coefficient and the Spearman correlation coefficient to conduct correlation measurement experiments on the variables of each period and $\text{PM}_{2.5}$. However,

meteorological factors such as humidity, wind speed, and wind direction have a low correlation with $PM_{2.5}$. Results show that the correlation factors obtained from both correlations are similar. The main difference is that the Spearman-related experiments have NOx. However, the Spearman correlation coefficient is the primary evaluation criterion for measuring non-linear correlation values. Therefore, we included NOx in the air factor reference. Consequently, we found that, based on Pearson and Spearman correlation, TEMP, CO, NO, NO₂, NO_x, O₃, PM₁₀, PM_{2.5}, and SO₂ are the parameters suggested for the model's training.

4.1.4. Principal Component Analysis

Table 8 shows that the air pollution factors in this data-sheet can be divided into three groups, because the main change is to reduce the dimension of the data. Therefore, this table shows that the correlation between those factors is high and shows which elements can be ignored to join the prediction model. The table serves as a reference in the experiment's factor selection.

Table 8. Principal component analysis results.

Ingredient	1	2	3
NOX	0.934		
NO ₂	0.904		
CO	0.898		
NO	0.655	−0.322	
WS	−0.563		−0.444
O ₃	−0.374	0.765	
PM ₁₀	0.608	0.682	
PM ₂₅	0.613	0.667	
WD			0.831
SO ₂	0.353		0.398

4.2. Time Lag Experiment

Table 9 shows the experimental results of using the Pearson correlation coefficient to correlate different time stays. The table shows that, except for the period of O₃ after the 3 h delay and the period of NO after the 6 h delay, the maximum correlation time retention of the remaining air pollution factors is 0 h (no lag time). This means that PM_{2.5} has the highest correlation with these current air factor values.

Table 9. Lag time evaluation (Pearson correlation coefficient).

	O ₃	SO ₂	CO	PM ₁₀	NO ₂	NO
0 h	0.059	0.389	0.435	0.869	0.279	0.020
1 h	0.097	0.371	0.379	0.842	0.255	0.006
2 h	0.132	0.359	0.315	0.792	0.220	−0.017
3 h	0.149	0.355	0.255	0.737	0.193	−0.051
4 h	0.140	0.345	0.207	0.686	0.184	−0.083
5 h	0.121	0.315	0.167	0.642	0.186	−0.111
6 h	0.094	0.284	0.146	0.605	0.186	−0.125
7 h	0.065	0.257	0.144	0.571	0.190	−0.120
8 h	0.041	0.236	0.132	0.547	0.181	−0.122

Table 10 shows the experimental results of using Spearman correlation coefficients to correlate the different time stays. The table shows that, except for the period of O₃ after the 3 h delay and the period of NO after the 8 h delay, the maximum correlation time retention of the remaining air pollution factors is 0 h (no lag time). This means that PM_{2.5} has the highest correlation with these current air factor values.

Table 10. Lag time evaluation (Spearman correlation coefficient).

	O ₃	SO ₂	CO	PM ₁₀	NO ₂	NO
0 h	0.063	0.238	0.364	0.750	0.229	0.068
1 h	0.109	0.219	0.303	0.748	0.181	0.031
2 h	0.148	0.212	0.235	0.726	0.134	−0.026
3 h	0.164	0.193	0.179	0.711	0.094	−0.086
4 h	0.158	0.172	0.140	0.696	0.078	−0.130
5 h	0.144	0.150	0.102	0.679	0.066	−0.151
6 h	0.135	0.135	0.083	0.658	0.059	−0.163
7 h	0.130	0.123	0.080	0.626	0.058	−0.168
8 h	0.130	0.120	0.075	0.600	0.042	−0.185

In fact, based on the correlation experiment results of the above two tables, the correlation performance of the air pollution factors other than NO is best. The best time lag is similar. The optimal time retentions of NO measured by these two methods are 6 and 8 h. However, the Spearman correlation coefficient is the primary evaluation criterion for measuring non-linearly related values. Therefore, the experiment was based on the Spearman correlation coefficient.

4.3. Missing Value Imputation Experiment

In order to determine a more suitable method of complementing the missing value, the KNN method and the linear interpolation method were implemented. In the experiment, the error value of the missing value of KNN in a short time was more extensive, and the error value of the missing value of the linear interpolation method was higher over the long term.

Therefore, Table 11 indicates that KNN is suitable for a long-term missing value complement. At the same time, linear interpolation is ideal for a short-term missing value complement.

Table 11. Missing value imputation comparison.

Average Error	KNN	Linear
random 50 h	4.7 ($\mu\text{g}/\text{m}^3$)	3.3 ($\mu\text{g}/\text{m}^3$)
6 consecutive hours	3.3 ($\mu\text{g}/\text{m}^3$)	3.0 ($\mu\text{g}/\text{m}^3$)
12 consecutive hours	2.5 ($\mu\text{g}/\text{m}^3$)	4.2 ($\mu\text{g}/\text{m}^3$)
24 consecutive hours	2.8 ($\mu\text{g}/\text{m}^3$)	3.3 ($\mu\text{g}/\text{m}^3$)

4.4. Model Training and Experiment

This study aimed to train a prediction model to predict the $\text{PM}_{2.5}$ concentration of each air quality monitoring station and the surrounding area over the following 8 h. Training deep learning models leads to many problems with parameters, prediction time adjustment, neural network selection, and overfitting. These are discussed in this section.

4.4.1. Overfitting Experiment

There are three different methods to avoid overfitting. In order to understand whether these methods are useful, we adjusted the model architecture: We generalized, reduced the network size and the L1 and L2 normalization, and added a dropout layer for comparison. Figure 8 shows the loss training and visualization of the prediction results in four different architecture models. There is overfitting in the training process of Figure 8a. Figure 8b–d show improved training with no overfitting. In the prediction visualization, we did k-fold cross-validation (in 8 folds). These results might be random in neural network prediction, and the estimation could be slightly different every time we execute the prediction. Therefore, k-fold cross-validation was initially used to determine whether our models can demonstrate stable prediction.

Figure 8. The training loss and model prediction results of PM_{2.5} in µg/m³ in graphs. (a) Loss and prediction results of PM_{2.5} in µg/m³. (b) Loss and prediction results of PM_{2.5} in µg/m³. (c) Loss and prediction results for PM_{2.5} in µg/m³. (d) Loss and prediction results of PM_{2.5} in µg/m³.

We can see that these three adjustment methods have improved the model's accuracy by evaluating the prediction results. Therefore, these methods will be used as the model of this study. Table 12 presents a comparison table of MAPE to evaluate the model's performance.

Table 12. Comparison of overfitting processing methods.

	Normally	Reduce the Network Size	L1 and L2 Normalized	Dropout Layer
1st hour	22%	16%	21%	22%
2nd hour	31%	26%	29%	29%
3rd hour	38%	32%	35%	34%
4th hour	44%	37%	39%	38%
5th hour	49%	42%	44%	43%
6th hour	53%	48%	48%	47%
7th hour	54%	52%	50%	49%
8th hour	54%	56%	53%	51%
Average	43%	38%	39%	39%

4.4.2. Time Unit Adjustment Experiment

The experiments on how many time unit data need to be predicted are demonstrated in this section. The forecast times were 8, 16, 24, and 48 h. Table 13 shows the experimental results evaluated by MAPE. From this prediction result, it can be seen that the prediction accuracy in less time is higher than it is with more time. Therefore, the model in this study used the 8 h time unit as the primary input time unit.

Table 13. Time unit adjustment comparison.

	8 h Unit	16 h Unit	24 h Unit	48 h Unit
1st hour	9%	12%	13%	15%
2nd hour	16%	20%	21%	22%
3rd hour	21%	23%	25%	24%
4th hour	25%	25%	27%	26%
5th hour	26%	25%	29%	29%
6th hour	27%	25%	30%	30%
7th hour	26%	25%	30%	28%
8th hour	26%	24%	31%	32%

4.4.3. Meteorological Factor Experiment

In this study, though chemical factors were added as variables to the model training, we also added meteorological factors to the model training, including temperature, humidity, wind direction, and wind speed. We used LSTM as the neural network architecture for this experiment and determined whether the addition of meteorological factors would provide more accurate prediction results for the PM_{2.5} prediction model. It can be seen in Figure 9 that, after adding meteorological factors as variables for model training, the performance of its evaluation value is higher than that when only chemical factors are added. Therefore, meteorological and chemical factors were used as training variables in the training of subsequent models.

4.5. Comparison of Neural Network Prediction Models

In the experiments, the model had six kinds of neural networks: CNN, RNN, LSTM, GRU, bidirectional LSTM, and bidirectional GRU. Multiple variables were input into the model to train and obtain training results and evaluation values of the prediction data. The following subsections show each neural network's training results and evaluation comparison. Table 14 shows the neural network architecture of each model.

Figure 9. Meteorological factor experiment comparison.

Table 14. The layer architecture of each model.

Model	Layer Architecture
CNN	Convolution1D (128)
	Convolution1D (128)
	MaxPooling1D (1)
	Dense(128) activation='relu'
	Dense(128) activation='relu'
RNN	Dense(8) activation='linear'
	RNN(128)
	RNN(128)
	Dense(128) activation='relu'
	Dense(128) activation='relu'
LSTM	Dense(8) activation='linear'
	LSTM(128)
	LSTM(128)
	Dense(128) activation='relu'
	Dropout(0.1)
Bi-LSTM	Dense(128) activation='relu'
	Dense(8) activation='linear'
	Bi-LSTM(128)
	Timdistributed(Dense(1))
GRU	Dense(128) activation='relu'
	Dense(128) activation='relu'
	Dense(8) activation='linear'
	GRU(128)
	GRU(128)
Bi-GRU	Bi-GRU(128)
	Bi-GRU(128)
	Dense(128) activation='relu'
	Dense(8) activation='linear'

4.5.1. CNN

Figure 10 demonstrates the evaluation of the prediction results using the CNN model, with MAPE, RMSE, MSE, and MAE as evaluation methods.

Figure 10. Prediction result evaluation of CNN model.

4.5.2. RNN

Figure 11 presents the evaluation of the prediction results using the RNN model, with MAPE, RMSE, MSE, and MAE as evaluation methods.

Figure 11. Prediction result evaluation of RNN model.

4.5.3. LSTM

Figure 12 shows the evaluation of the prediction results using the LSTM model, with MAPE, RMSE, MSE, and MAE as evaluation methods.

Figure 12. Prediction result evaluation of LSTM model.

4.5.4. Bi-LSTM

Figure 13 describes the evaluation of the prediction results using the Bi-LSTM model, with MAPE, RMSE, MSE, and MAE as evaluation methods.

Figure 13. Prediction result evaluation of Bi-LSTM model.

4.5.5. GRU

Figure 14 describes the evaluation of the prediction results using the GRU model, with MAPE, RMSE, MSE, and MAE as evaluation methods.

Figure 14. Prediction result evaluation of GRU model.

4.5.6. Bi-GRU

Figure 15 presents the evaluation of the prediction results using the Bi-GRU model, with MAPE, RMSE, MSE, and MAE as evaluation methods.

Figure 15. Prediction result evaluation of Bi-GRU model.

4.5.7. Comparison of Various Neural Networks

The following tables present a comparison of the predicted $PM_{2.5}$ concentration of CNN, RNN, LSTM, GRU, bidirectional LSTM, and bidirectional GRU over the following 8 h, including the real value, predicted value, and error value for each hour. In terms of hourly prediction, the first hour had lower values with an average of 12. In the following hour, the $PM_{2.5}$ prediction values were increased gradually from 1 to 6. Figure 16 shows that the average error value of the LSTM prediction result is smaller than the other neural network methods. Therefore, LSTM is more suitable than other neural networks to construct deep learning prediction models for the air pollution data type used in this study.

Figure 16. PM_{2.5} real and prediction values in various neural networks.

The RMSE values of RNN, CNN, LSTM, Bi-LSTM, GRU, and Bi-GRU are 2.4702, 3.4984, 1.8643, 2.8379, 2.6398, and 2.6695, respectively. It can be seen that the LSTM model has the lowest error among the other models. Figure 17 shows the comparison of RMSE values of various models.

Figure 17. RMSE comparison of various neural networks.

4.6. Visualization

Diffusion Map

The website visualization applies IDW to perform diffusion calculations on the current and forecast data of the entire Taiwan Air Quality Monitoring Stations. The value of PM_{2.5} in areas around stations can be estimated.

The Taiwan diffusion map is shown in Figure 18. The map shows the status of the PM_{2.5} value in each sensor area. The majority of green points indicates that PM_{2.5} values

were at a good level based on the $PM_{2.5}$ index from Figure 18. The color in the bar reflects the status of the $PM_{2.5}$ value, as described in Table 15.

Figure 18. Taiwan $PM_{2.5}$ diffusion map (a).

Table 15. $PM_{2.5}$ index.

$PM_{2.5}$ Index	Color	Status
0 to <15.5	Green	Good
15.5 to <35.5	Orange	Moderate
35.5 to <54.5	Red	Unhealthy
54.5 to <150.5	Purple	Very Unhealthy
150.5 to <250.5	Dark Grey	Severe
250.5 to >300	Black	Hazardous

The $PM_{2.5}$ index value we used is described in Figure 19.

Figure 19. $PM_{2.5}$ index value.

Figure 20 shows the data at a diffusion point. The latitude and longitude of the point are also displayed based on the area that the user selects. In the map, we can see that the real-time $PM_{2.5}$ is 60. Therefore, the point color in that area was purple. We used a different color to represent the status of the $PM_{2.5}$ index.

Figure 21a–d show the actual value of $PM_{2.5}$ and the predicted value of the following 8 h at each station. The solid line is the actual value, and the dotted line is the predicted value. The change of $PM_{2.5}$ over the following 8 h is represented by a line chart. The following figures are the line chart of Songshan (Figure 21a), Xitun (Figure 21b), Zuoying (Figure 21c), and Hualien (Figure 21d) stations.

Figure 20. Taiwan PM_{2.5} diffusion map (b).

Figure 21. PM_{2.5} predicted value. (a) PM_{2.5} predicted value of Songshan. (b) PM_{2.5} predicted value of Xitun. (c) PM_{2.5} predicted value of Zuoying. (d) PM_{2.5} predicted value of Hualien.

4.7. Discussion

This study experimented with several machine learning procedures to find an optimum model. The datasets used were collected from 77 air quality monitoring stations throughout Taiwan. For the handling of missing values, we used linear interpolation and KNN, and these two methods were compared to determine the best handling of missing values. The missing value imputation measures the average error comparison in four time categories: 6, 2, 24, and 50 h. KNN was suitable for a long-term missing value complement, while linear interpolation was suitable for a short-term missing value complement. For overfitting experiments, there are three methods: generalize, reduce the network size and the L1 and L2 normalization, and add a dropout layer. The general method led to overfitting, while we found no overfitting using the other three methods. We determined 8, 16, 24, and 48 h as the times used for forecasting. The experimental results were evaluated based on MAPE values. From this prediction result, it could be seen that prediction accuracy with less time was higher than it was with more time. Therefore, the model in this study used an 8 h time unit as the primary input of time. Two correlation analyses, using Pearson and Spearman correlation coefficients, were implemented to provide a clear, quantitative standard of air pollutant feature selection. Combined with the correlation calculation method and lag time mechanism, it was possible to find the time points with

a high correlation between pollution factors and provide more accurate data to facilitate model training. In this case, the optimal time lag was between 6 and 8 h. In terms of hourly prediction, the first hour had more low RMSE values with an average of 12. In the following hour, the RMSE values increased gradually from 1 to 6. The average RMSE values of RNN, CNN, LSTM, Bi-LSTM, GRU, and Bi-GRU were 2.4702, 3.4984, 1.8643, 2.8379, 2.6398, and 2.6695, respectively. It can be seen that the average error value of the LSTM prediction result is smaller than that of the other neural network methods.

5. Conclusions and Future Study

This study implemented the short-term prediction of PM_{2.5} using LSTM deep learning methods. In this case, we empirically studied the time series model in terms of PM_{2.5} data by performing comparison methods from data preprocessing, correlation analysis, and machine learning training to find an optimum model. In terms of data preprocessing and handling missing values, we compared KNN and linear interpolation. The finding was that KNN is suitable for a long-term missing value complement, while linear interpolation is ideal for a short-term missing value complement. We also used the Pearson and Spearman correlation coefficient methods for lag time experiments. It was found that the best lag time is similar in both methods. We combined data from chemical and meteorological variables in terms of correlation analysis by using Pearson and Spearman correlation coefficients. Meteorological parameters such as humidity, wind speed, and wind direction have a weak relationship with PM_{2.5}. According to the results, the correlation factors obtained from both correlation methods are almost the same. The primary distinction is that NO_x is present in the Spearman experiments. The Pearson and Spearman correlation's recommended parameters are TEMP, CO, NO, NO₂, NO_x, O₃, PM₁₀, PM_{2.5}, and SO₂. In terms of machine learning models, it can be seen that the average error value of LSTM is smaller than that of the other neural network methods. Therefore, LSTM is more suitable than other neural networks for constructing deep learning prediction models of the air pollution data type used in this study. In the future, more advanced models, such as ensemble learning, could be performed for further comparison.

Author Contributions: Conceptualization, C.-T.Y. and C.-Y.H.; methodology, J.-R.L.; software, E.K. and H.L.; validation, J.-R.L. and Y.-H.C.; formal analysis, J.-R.L. and Y.-H.C.; investigation, C.-T.Y. and J.-R.L.; resources, C.-T.Y. and J.-R.L.; data curation, E.K. and H.L.; writing—original draft preparation, E.K. and H.L.; writing—review and editing, C.-T.Y. and C.-Y.H.; visualization, Y.-H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was sponsored by the Ministry of Science and Technology (MOST), Taiwan, under Grant No. 110-2221-E-029-020-MY3, 110-2621-M-029-003, 110-2622-E-029-003, 110-2811-E-029-003, TCVGH-T1097801, TCVGH-T1107803, and TCVGH-1107201C.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that there is no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RMSE	Root Means Square Error
MAPE	Mean Absolute Percentage Error
MAE	Mean Absolute Error
MSE	Mean Square Error
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit

RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
PM	Particulate Matter
TEMP	Temperature
CO	Carbon Monoxide
NO	Nitrogen Monoxide
NOx	Nitrogen Oxide
SO	Sulfur Oxide
KNN	K-Nearest Neighbors
JSON	JavaScript Object Notation
CSV	Comma-Separated Values
XML	Extensible Markup Language
MySQL	My Structured Query Language
IDW	Inverse Distance Weighting

References

1. Zhou, Y.; Chang, F.J.; Chang, L.C.; Kao, I.F.; Wang, Y.S. Explore a deep learning multi-output neural network for regional multi-step-ahead air quality forecasts. *J. Clean. Prod.* **2019**, *209*, 134–145. [\[CrossRef\]](#)
2. Lee, C.F.; Yang, C.T.; Kristiani, E.; Tsan, Y.T.; Chan, W.C.; Huang, C.Y. Recurrent Neural Networks for Analysis and Automated Air Pollution Forecasting. *Int. Conf. Front. Comput.* **2019**, *542*, 50–59. [\[CrossRef\]](#)
3. Wen, C.; Liu, S.; Yao, X.; Peng, L.; Li, X.; Hu, Y.; Chi, T. A novel spatiotemporal convolutional long short-term neural network for air pollution prediction. *Sci. Total Environ.* **2019**, *654*, 1091–1099. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Wu, Q.; Lin, H. Daily urban air quality index forecasting based on variational mode decomposition, sample entropy and LSTM neural network. *Sustain. Cities Soc.* **2019**, *50*, 101657. [\[CrossRef\]](#)
5. Ma, J.; Li, Z.; Cheng, J.C.; Ding, Y.; Lin, C.; Xu, Z. Air quality prediction at new stations using spatially transferred bi-directional long short-term memory network. *Sci. Total Environ.* **2020**, *705*, 135771. [\[CrossRef\]](#)
6. Li, Y.; Zhu, Z.; Kong, D.; Han, H.; Zhao, Y. EA-LSTM: Evolutionary attention-based LSTM for time series prediction. *Knowl.-Based Syst.* **2019**, *181*, 104785. [\[CrossRef\]](#)
7. Zhao, J.; Deng, F.; Cai, Y.; Chen, J. Long short-term memory-Fully connected (LSTM-FC) neural network for PM_{2.5} concentration prediction. *Chemosphere* **2019**, *220*, 486–492. [\[CrossRef\]](#)
8. Kim, T.Y.; Cho, S.B. Web traffic anomaly detection using C-LSTM neural networks. *Expert Syst. Appl.* **2018**, *106*, 66–76. [\[CrossRef\]](#)
9. Cinar, Y.G.; Mirisaee, H.; Goswami, P.; Gaussier, E.; Ait-Bachir, A. Period-aware content attention RNNs for time series forecasting with missing values. *Neurocomputing* **2018**, *312*, 177–186. [\[CrossRef\]](#)
10. Zhang, Y.; Zhang, R.; Ma, Q.; Wang, Y.; Wang, Q.; Huang, Z.; Huang, L. A feature selection and multi-model fusion-based approach of predicting air quality. *ISA Trans.* **2020**, *100*, 210–220. [\[CrossRef\]](#)
11. 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*. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Demirhan, H.; Renwick, Z. Missing value imputation for short to mid-term horizontal solar irradiance data. *Appl. Energy* **2018**, *225*, 998–1012. [\[CrossRef\]](#)
13. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting*; ACM: New York, NY, USA, 2015; pp. 802–810. Available online: <https://proceedings.neurips.cc/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf> (accessed on 15 December 2021).
14. Wu, H.; Zhang, B. A deep convolutional encoder-decoder neural network in assisting seismic horizon tracking. *arXiv* **2018**, arXiv:1804.06814.
15. Zhang, B.; Zhang, H.; Zhao, G.; Lian, J. Constructing a PM_{2.5} concentration prediction model by combining auto-encoder with Bi-LSTM neural networks. *Environ. Model. Softw.* **2020**, *124*, 104600. [\[CrossRef\]](#)
16. Salcedo-Sanz, S.; Ghamisi, P.; Piles, M.; Werner, M.; Cuadra, L.; Moreno-Martínez, A.; Izquierdo-Verdiguier, E.; Muñoz-Marí, J.; Mosavi, A.; Camps-Valls, G. Machine Learning Information Fusion in Earth Observation: A Comprehensive Review of Methods, Applications and Data Sources. *Inf. Fusion* **2020**, *63*, 256–272. [\[CrossRef\]](#)
17. Greco, L.; Ritrovato, P.; Xhafa, F. Prediction of PM_{2.5} Concentration Based on Multiple Linear Regression. In Proceedings of the 2019 International Conference on Smart Grid and Electrical Automation (ICSGEA), Xiangtan, China, 10–11 August 2019; pp. 457–460. [\[CrossRef\]](#)
18. Yuan, W.; Wang, K.; Bo, X.; Tang, L.; Wu, J. A novel multi-factor & multi-scale method for PM_{2.5} concentration forecasting. *Environ. Pollut.* **2019**, *255*, 113187. [\[CrossRef\]](#)
19. 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. [\[CrossRef\]](#)

20. Tsai, Y.-T.; Zeng, Y.-R.; Chang, Y.-S. Air Pollution Forecasting Using RNN with LSTM. In Proceedings of the 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), Athens, Greece, 12–15 August 2018; pp. 1074–1079. [\[CrossRef\]](#)
21. Ma, J.; Ding, Y.; J, L.G.V.; Lin, C.; Wan, Z. Spatiotemporal Prediction of PM_{2.5} Concentrations at Different Time Granularities Using IDW-BLSTM. *IEEE Access* **2019**, *7*, 107897–107907. [\[CrossRef\]](#)
22. Kim, H.S.; Park, I.; Song, C.H.; Lee, K.; Yun, J.W.; Kim, H.K.; Jeon, M.; Lee, J.; Han, K.M. Development of a daily PM₁₀ and PM_{2.5} prediction system using a deep long short-term memory neural network model. *Atmos. Chem. Phys* **2019**, *19*, 12935–12951. [\[CrossRef\]](#)
23. Ayturan, Y.A.; Ayturan, Z.C.; Altun, H.O.; Kongoli, C.; Tuncel, F.D.; Dursun, S.; Ozturk, A. Short-term prediction of PM_{2.5} pollution with deep learning methods. *Glob. Nest J.* **2020**, *22*, 126–131.
24. Li, S.; Xie, G.; Ren, J.; Guo, L.; Yang, Y.; Xu, X. Urban PM_{2.5} Concentration Prediction via Attention-Based CNN–LSTM. *Appl. Sci.* **2020**, *10*, 1953. [\[CrossRef\]](#)
25. Wu, X.; Wang, Y.; He, S.; Wu, Z. PM_{2.5}/PM₁₀ ratio prediction based on a long short-term memory neural network in Wuhan, China. *Geosci. Model Dev.* **2020**, *13*, 1499–1511. [\[CrossRef\]](#)
26. Yang, G.; Lee, H.; Lee, G. A Hybrid Deep Learning Model to Forecast Particulate Matter Concentration Levels in Seoul, South Korea. *Atmosphere* **2020**, *11*, 348. [\[CrossRef\]](#)
27. Xayasouk, T.; Lee, H.; Lee, G. Air Pollution Prediction Using Long Short-Term Memory (LSTM) and Deep Autoencoder (DAE) Models. *Sustainability* **2020**, *12*, 2570. [\[CrossRef\]](#)
28. Kow, P.Y.; Wang, Y.S.; Zhou, Y.; Kao, I.F.; Issermann, M.; Chang, L.C.; Chang, F.J. Seamless integration of convolutional and back-propagation neural networks for regional multi-step-ahead PM_{2.5} forecasting. *J. Clean. Prod.* **2020**, *261*, 121285. [\[CrossRef\]](#)
29. Lin, Y.C.; Chi, W.J.; Lin, Y.Q. The improvement of spatial-temporal resolution of PM_{2.5} estimation based on micro-air quality sensors by using data fusion technique. *Environ. Int.* **2020**, *134*, 105305. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Chen, Y.; Wang, L.; Li, F.; Du, B.; Choo, K.K.R.; Hassan, H.; Qin, W. Air quality data clustering using EPLS method. *Inf. Fusion* **2017**, *36*, 225–232. [\[CrossRef\]](#)
31. Liu, X.; Qin, M.; He, Y.; Mi, X.; Yu, C. A new multi-data-driven spatiotemporal PM_{2.5} forecasting model based on an ensemble graph reinforcement learning convolutional network. *Atmos. Pollut. Res.* **2021**, *12*, 101197. [\[CrossRef\]](#)
32. Yan, X.; Zang, Z.; Luo, N.; Jiang, Y.; Li, Z. New interpretable deep learning model to monitor real-time PM_{2.5} concentrations from satellite data. *Environ. Int.* **2020**, *144*, 106060. [\[CrossRef\]](#)
33. Li, T.; Hua, M.; Wu, X.U. A hybrid CNN-LSTM model for forecasting particulate matter (PM_{2.5}). *IEEE Access* **2020**, *8*, 26933–26940. [\[CrossRef\]](#)
34. Xiao, F.; Yang, M.; Fan, H.; Fan, G.; Al-Qaness, M.A. An improved deep learning model for predicting daily PM_{2.5} concentration. *Sci. Rep.* **2020**, *10*, 20988. [\[CrossRef\]](#)
35. Environmental Loading. Taiwan Environmental Protection Administration. Available online: <https://www.epa.gov.tw/eng/312737106A724758> (accessed on 15 December 2021).
36. Climate Statistics: Monthly Mean, Taiwan Central Weather Bureau. Available online: <https://www.cwb.gov.tw/V8/E/C/Statistics/monthlymean.html> (accessed on 15 December 2021).
37. Economy: Fast Focus, Government Portal of the Republic of China (Taiwan). Available online: https://www.taiwan.gov.tw/content_7.php (accessed on 15 December 2021).
38. Veselík, P.; Sejkorová, M.; Nieoczym, A.; Caban, J. Outlier identification of concentrations of pollutants in environmental data using modern statistical methods. *Pol. J. Environ. Stud.* **2020**, *29*, 853–860. [\[CrossRef\]](#)