The air quality predication based on machine learning methods

Maohao Ran

Schools of Engineering, Arizona State University, Tempe, 85281 Arizona, United States

mran2@asu.edu

Abstract. Air quality is the focus of attention all over the world. As an important index to measure air quality, accurate prediction of $PM_{2.5}$ plays an essential role in regulating air quality. This paper uses different machine learning algorithms and neural networks to infer air quality (AQ) in the study area and observe the impact of these methods on accuracy. The results indicate that shuffling the data can enhance the model's performance. In addition, the neural network is the most affected by the data shuffle operation compared to other models. In the case of a shuffle operation, the performance of the neural networks is the lowest among all models. However, in the case of non-shuffle data, the neural network performance is the best among all models. Therefore, in the absence of large-scale data sets, the traditional machine learning method with a relatively small scale should be selected to model the air quality prediction problem because the traditional machine learning method performs better in the small sample data scene.

Keywords: neural networks, shuffle data, environment predication.

1. Introduction

In the past two decades, China's economy has overgrown. A large part of the industrial power source has been obtained by consuming fossil fuels and other energy sources and destroying the environment. In addition, the rapid development of highly polluted industrial activities inevitably leads to severe air pollution problems. Industrialization and urbanization have aggravated environmental health risks and pollution and accelerated China's continuous decline in air quality. The high economic development has increased people's health risks and paid a heavy price. With the increasing improvement of living standards, people pay more and more attention to health and sustainable development, which will be a hot topic in the future and need to be deeply studied. If people want to be healthy and society wants to achieve sustainable development, the first thing to do is to solve the problem of environmental pollution. Among them, air pollution is a kind of environmental pollution with a large degree of influence and a wide range of influence. It can not only cause global warming, water pollution, acid rain, smog, and other environmental problems but also seriously threaten people's lives and health. A few days ago, air pollution greatly impacted some big cities' visibility and ecological environment, among which $PM_{2.5}$ is a necessary standard to measure air quality. $PM_{2.5}$, also known as inhalable lung particles, has a diameter of less than or equal to 2.5 microns. Although the content of $PM_{2.5}$ in the atmosphere is not high, it seriously affects human health. Compared with large particles of air pollutants, $PM_{2.5}$ has a smaller diameter, is easy to carry toxic chemicals, and can carry diseases into human organs to cause cardiovascular diseases and respiratory diseases, which pose a great threat to human health.

With the rise of artificial intelligence, traditional machine learning was first applied to the prediction of $PM_{2.5}$ Extreme Learning Machine (ELM) to predict the concentration of air pollutants in Hong Kong [1]. ELM outperforms other established statistical techniques in terms of performance prediction of $PM_{2.5}$, generalization ability, and learning speed. The emergence of deep learning makes machine learning a breakthrough in predicting $PM_{2.5}$ concentration. Previous studies have proposed that based on long short-term memory networks (LSTM), the problems of gradient explosion and gradient dissipation in the prediction of recurrent neural networks (RNN) were solved. In terms of parameter update, convergence time, and generalization, GRU outperforms LSTM and is more appropriate for $PM_{2.5}$ prediction [2]. In pursuit of more accurate and reliable prediction results, more and more people apply the integrated model to $PM_{2.5}$ prediction. An integrated model of convolutional neural network (CNN) and LSTM is proposed to predict $PM_{2.5}$ [3]. Compared with SVM, RD, DT, MLP, CNN, and LSTM, it shows the feasibility and practicability of the model. ANN, CNN, LSTM, and other neural network combination models are used to extract the temporal-spatial relationship and predict the air quality for up to 48 hours [4]. The proposed Spatial-temporal model needs complicated Euclidean distance before extracting spatially related features, which is complicated and inefficient. Mutual information is used to analyze various characteristic factors of $PM_{2.5}$, including spatial relationships [5]. The research shows that by comparing the air pollutant data of different areas, the correlation between other air pollutant data in the same area and $PM_{2.5}$ data is greater, so other air pollutants in the same area should be the first characteristic to be considered. But how to effectively extract the correlation between other features and $PM_{2.5}$ is still a problem to be solved. StemGNN, a multivariate time series prediction model without prior knowledge, is put forward [6]. This model automatically uses a self-attention mechanism to construct the graph structure between multivariate time series and then uses graph convolution to extract the graph structure features. The model provides a new idea to consider the correlation between other air pollutants and $PM_{2.5}$ in the same area.

In this paper, the $PM_{2.5}$ concentration in Chengdu, China, was inferred. The study area is 70km×70km square in Chengdu and is divided into 4,900 1km×1km spatial grids. The goal of this article is to deduce the study area's hourly $PM_{2.5}$ concentrations. This paper uses different machine learning algorithms and neural networks to infer air quality (AQ) in the study area and observe the impact of these methods on accuracy.

2. Methods

In this paper, the accuracy of $PM_{2.5}$ concentration prediction by various models is verified based on the experimental data, including XGBoost, CatBoot, Linear Regression, SVR and Neural Network (3 layer). This section mainly introduces the principles of five models involved in the experiment in detail.

2.1. XGBoost

The integrated method creates many weak evaluators from the data and aggregates their modeling outputs to produce regression performance that is superior to that of a single model. The XGBoost model is a typical method of boosting in integrated algorithms and is a machine learning algorithm that is implemented within the Gradient Boosting framework. The XGBoost model's concept is to incrementally add trees (Figure 2), each tree addition requiring the learning of a new function F to account for the residual error of the previous forecast. K trees are created after training, and each tree will eventually fall to a matching leaf node, with each leaf node corresponding to a score. The projected value of the sample may be obtained by adding the scores assigned to each tree. The projected outcome of the entire model on sample i is represented in formula under the assumption that there are K trees in the model (1):

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{1}$$

where f is the leaf scoring map of the kth tree, y is the model's final prediction score, x is the feature vector corresponding to sample i, and K is all the trees established.

The benefits of the XGBoost algorithm are as follows: The objective function is given a regularization term by XGBoost, which lowers the variance of the model, simplifies the trained model, and effectively prevents over-fitting. Additionally, the loss function is given a second-order Taylor expansion by XGBoost, which improves the model's accuracy. XGBoost enables column sampling, allows parallelization, learns from RF method, and has quick training time.

2.2. CatBoost

Yandex presented CatBoost, an open-source machine learning package, in 2017. It is still an upgraded method inside the confines of the GBDT algorithm, much as the earlier mentioned XGBoost and LightGBM. Based on the symmetric decision trees technique, it is a GBDT framework. It handles categorical variables and also has fewer parameters and good accuracy. Its ability to effectively and appropriately handle categorical characteristics is likewise a key feature and benefit. As implied by its name, CatBoost combines category and boost to enhance the algorithm's generalization and accuracy by reducing gradient bias and prediction shift [7].

In order to decode the classification values into numbers and address the exponential growth problem of feature combinations by the greedy technique at the new split of the existing tree, CatBoost combines a variety of statistical classification characteristics and numerical features. The emphasis is on the following actions to prevent over-fitting, similar to average coding: (1) Subsets of the records are created at random. (2) The mathematical formula discovered via research is displayed in formula after the label is turned into an integer and the classification characteristics are translated into digital features (2):

$$avgTarget = \frac{countinClass + prior}{totalCount + 1} \tag{2}$$

where countinClass is the quantity of the target's specified classification features. Total is the number of previous objects. Prior by initial Parameter specification [8,9].

The following are CatBoost's innovations: 1) It incorporates a cutting-edge algorithm that converts category information into numerical features automatically. To create new numerical features, first create some data on the categorical characteristics, then figure out how frequently each category feature occurs. 2) Catboost employs combined category features, which are features that may be connected, considerably enhancing the dimensions of features. 3) Using the ranking promotion approach to deal with the noise points in the training set, which avoids the gradient estimation deviation and fixes the prediction deviation issue. 4) The basis model is the perfectly symmetric tree.

2.3. Linear regression

In statistics, linear regression is a kind of regression analysis that simulates the relationship between a number of independent and dependent variables. A linear combination and parameters of the model make up this function. Linear regression can be split into simple and multiple linear regression[10]. The former has only one independent variable. The latter has two or more independent variables.

In linear regression, the unobserved parameters of the model are also estimated using the data after the data are modelled using a linear prediction function. They're referred to as linear models. The affine function of X is defined as the conditional mean of y for a given value of x in the most common type of linear regression modeling. In extreme cases, the conditional distribution of y given x, where y is a linear function of x, can be the median or another quantile of the linear regression model.

In general, the least square approach may be used to find the equation of linear regression and calculate the straight line for $\mathrm{y} = \mathrm{bx} + \mathrm{a}$. Assuming that there are $x_1, x_2, \ldots, x_k$, components, the linear relationship as indicated in formula (3) may often be taken into consideration. In general, there are frequently several factors that effect y.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon \tag{3}$$

Make n independent observations on y and $x_1, x_2, \cdots, x_k$, in the meantime, get n sets of observations $(\mathrm{x_{t1}}, \mathrm{x}_{t2}, \cdots, \mathrm{x_{tk}})$, $\mathrm{t} = 1,2,\ldots,\mathrm{n}(\mathrm{n} > \mathrm{k} + 1)$, which meet the relation (4).

To get n sets of observations $(\mathrm{x_{t1}}, \mathrm{x}_{t2}, \cdots, \mathrm{x_{tk}})$, $\mathrm{t} = 1,2,\ldots,\mathrm{n}(\mathrm{n} > \mathrm{k} + 1)$, that meet the relation (4), make n independent observations on y and $x_1, x_2, \cdots, x_k$ simultaneously.

$$y = \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \cdots + \beta_k x_{tk} + \varepsilon_t \tag{4}$$

2.4. Support vector regression (SVR)

Support vector regression (SVR) is developed from the concept of support vector machine (SVM), and is used in many scenarios such as prediction under nonlinear conditions. Support vector The regression problem can be described as seeking the mapping expression of nonlinear space as shown in formula (5):

$$y = f(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{m} \omega_i x_i + b \tag{5}$$

where x_i represents the value of each dimension in the training set, i represents the variable of each dimension, ω represents the variable coefficient, and b represents the offset.

For (x, y) in the sample, a prediction model f is obtained, which makes the prediction result closest to the actual value. Assuming that the regression model is almost the real value can be fully expressed, and there is only negligible error ε. The SVR problem can be transformed into formula (6):

$$min_{\omega,b} \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{m} \ell_\varepsilon(f(x_i) - y_i) \tag{6}$$

C is the regularization constant, and ℓ_ε is the insensitive loss function, and the expression of ℓ_ε is shown in formula (7).

$$\ell_\varepsilon(z) = \begin{cases} 0, |z| \leq \varepsilon \\ |z| - \varepsilon, |z| < \varepsilon \end{cases} \tag{7}$$

2.5. Neural network

Neural networks may be classified into two groups based on their model structures: feedforward networks (sometimes referred to as multilayer perceptron networks) and feedback networks. From a mathematical perspective, the former may be viewed as a large-scale non-linear mapping system, whereas the latter is a large-scale non-linear dynamic system. Artificial neural networks contain supervised, unsupervised, and semi-supervised learning following the preferred learning method. It may be classified into two types based on the operating mode: certainty and unpredictability; It may be classified into two categories based on the features of the time: continuous versus discrete.

Massive parallel processing, distributed storage, variable topology, high redundancy, and nonlinear operation are traits shared by all artificial neural networks, regardless of type. It therefore possesses quick computation, powerful association, great adaptability, strong fault tolerance, and strong self-organization abilities. Artificial neural networks, which have been used extensively to imitate intelligent behaviors, are built on these qualities and skills. Artificial neural networks, for instance, may be applied in object detection, image generation, image and voice recognition, control and optimization.

3. Experimental results and analysis

3.1. Data description

The study area is 70km×70km square in Chengdu and is divided into 4,900 1km×1km spatial grids. The goal of this article is to deduce the study area's hourly $\mathrm{PM}_{2.5}$ concentrations. This paper uses different machine learning algorithms and neural networks to infer air quality (AQ) in the study area and observe the impact of these methods on accuracy. The government-installed fixed air sensors yielded a total of 159 476 labels.

Feature in each grid contain: 1) Traffic: road congestion levels, vehicle speed, road length. 2) Meteorology: relative humidity, water vapor pressure, rainfall, solar radiation, temperature. 3)

Geography: points of interest (POI), such as eateries, shops, schools, hotels, and commercial areas, as well as land use type (such as a building, factory, or commercial area). Figure 1 to Figure 2 shows the visualizations of several features.

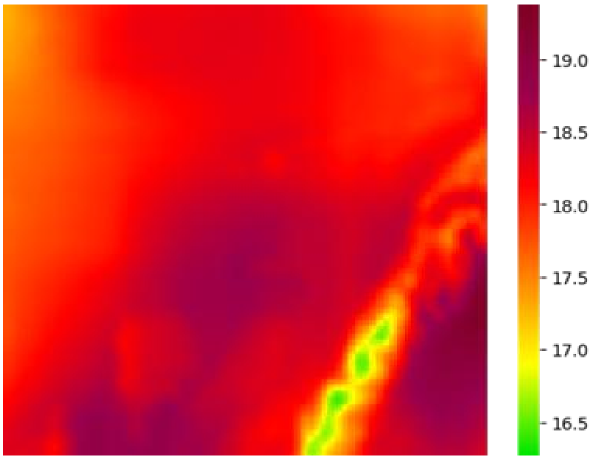

Figure 1. Temperature of an hour.

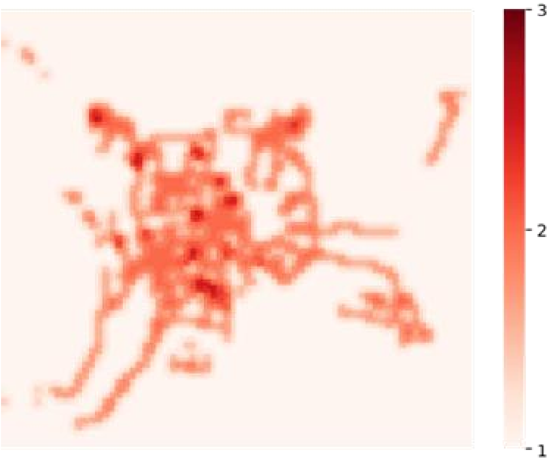

Figure 2. Road congestion levels (1-3) of an hour.

3.2. Evaluation matrix

The accuracy *of models can be evaluated using Mean Absolute Error (MAE), Symmetric Mean Absolute Percentage Error (SMAPE), Root Mean Square Error (RMSE) and R-squared (R2)*. The specific definition of each metrics is as follows.

The definition of RMSE, which is used measure of difference between the observed and actual value, is provided in the formula (8).

$$RMSE(X,h)=\sqrt{\frac{1}{m}\sum_{i=1}^{m}(h(x_i)-y_i)^2} \tag{8}$$

SMAPE is the average absolute percentage error, which fixes the shortcomings of the original MAPE, and its value ranges from 0% to 200%. Its definition is shown in formula (9).

$$SMAPE=\frac{100\%}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i-y_i|}{(|\hat{y}_i|+|y_i|)/2} \tag{9}$$

R^2 is Coefficient of Determination, also known as R-Squared. Its definition is that for a certain variable, there are a series of observed values y_i, and the corresponding predicted value $\hat{y}_i$. The definition of R square is shown as formula (10).

$$R^2=1-\frac{(\hat{y}_i-y_i)^2}{\sum_i(y_i-\bar{y})^2} \tag{10}$$

The average of absolute errors is known as MAE. Its definition is given in the formula and can more accurately represent the anticipated value error's real context (11).

$$MAE(X,h)=\frac{1}{m}\sum_{i=1}^{m}|h(x_i)-y_i| \tag{11}$$

3.3. Results and analysis

This study used two different types of experimental settings: 1) shuffle the data set during training; 2) Do not shuffle. Tables 1 and 2 show the experimental results under different settings, respectively.

Table 1. The results of five methods (dataset not shuffled).

Methods	RMSE	SMAPE	R^2	MAE
XGBoost	30.42	56.94	-0.49	20.65
CatBoost	31.92	68.15	-0.64	22.28
Linear Regression	32.96	66.51	-0.75	23.31
SVR	31.03	63.34	-0.55	21.87
Neural Networks (3 layer)	26	38.1	-0.09	15

Table 2. The results of five methods (dataset shuffled).

Methods	RMSE	SMAPE	R^2	MAE
XGBoost	11.51	36.31	0.56	0.2
CatBoost	11.42	39.56	0.57	8.21
Linear Regression	16.66	47.08	0.09	12.83
SVR	15.40	51.49	0.22	11.18
Neural Networks (3 layer)	17.43	54.42	0	13.45

Contrasting Table 1 and Table 2 to show the experimental outcomes, this paper thinks that shuffling the data can effectively improve the model's performance. Moreover, it is worth noting that the neural network is most affected by data shuffle operation compared with other models. In the case of a shuffle operation, the neural network's performance is the lowest among all models. Still, in the case of data without shuffle, the neural network's performance is the best among all models. In this paper, because of its solid fitting ability, the neural network takes the time series information contained in the training set without shuffle operation as a bias. However, by comparing the results of all models, the neural network model is not dominant. This paper thinks that the reason is that this paper does not provide high-density and large-scale data support for the neural network model.

4. Conclusion

The purpose of this work is to calculate the study area's hourly $PM_{2.5}$ concentrations. This paper uses various machine learning algorithms and neural networks to infer air quality (AQ) in the study area and observe the impact of these methods on accuracy. The results indicate that shuffling the data can effectively improve the model's performance. Moreover it should be noted that the neural network is the most affected by the data shuffle operation compared to other models. When using a shuffle operation, the performance of the neural networks is the lowest among all models. However, in the case of non-shuffle data, the neural network performance is the best among all models. However, when comparing the results of these models, the neural network model is not dominant. The next step of this study is to collect data sets related to air quality and develop an air pollution prediction model using more urban features.

References

[1] Zhang, J., & Ding, W. (2017). Prediction of air pollutants concentration based on an extreme learning machine: the case of Hong Kong. International journal of environmental research and public health, 14(2), 114.

[2] Zhou, X., Xu, J., Zeng, P., & Meng, X. (2019, February). Air pollutant concentration prediction based on GRU method. In Journal of Physics: Conference Series (Vol. 1168, No. 3, p. 032058).IOP Publishing.

[3] Huang, C. J., & Kuo, P. H. (2018). A deep CNN-LSTM model for particulate matter (PM2. 5) forecasting in smart cities. Sensors, 18(7), 2220.

[4] Soh, P. W., Chang, J. W., & Huang, J. W. (2018). Adaptive deep learning-based air quality prediction model using the most relevant spatial-temporal relations. Ieee Access, 6, 38186-38199.

[5] Pak, U., Ma, J., Ryu, U., Ryom, K., Juhyok, U., Pak, K., & Pak, C. (2020). Deep learning-based PM2. 5 prediction considering the spatiotemporal correlations: A case study of Beijing, China. Science of The Total Environment, 699, 133561.

[6] Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., ... & Zhang, Q. (2020). Spectral temporal graph neural network for multivariate time-series forecasting. Advances in neural information processing systems, 33, 17766-17778.

[7] Meng, Q., Ke, G., Wang, T., Chen, W., Ye, Q., Ma, Z. M., & Liu, T. Y. (2016). A communication-efficient parallel algorithm for decision tree. Advances in Neural Information Processing Systems, 29.

[8] Klein, A., Falkner, S., Bartels, S., Hennig, P., & Hutter, F. (2017, April). Fast bayesian optimization of machine learning hyperparameters on large datasets. In Artificial intelligence and statistics (pp. 528-536). PMLR.

[9] Hamid, A. J., & Ahmed, T. M. (2016). Developing prediction model of loan risk in banks using data mining. Machine Learning and Applications: An International Journal (MLAIJ), 3(1), 1-9.

[10] Gross, J., & Groß, J. (2003). Linear regression (Vol. 175). Springer Science & Business Media.