Communications for Statistical Applications and Methods
2023, Vol. 30, No. 2, 119–133

https://doi.org/10.29220/CSAM.2023.30.2.119
Print ISSN 2287-7843 / Online ISSN 2383-4757

Prediction of spatio-temporal AQI data

KyeongEun Kim[a], MiRu Ma[b], KyeongWon Lee[1,a]

[a]Department of Statistics, Seoul National University, Korea;
[b]Department of Statistics, Sungkyunkwan University, Korea

Abstract

With the rapid growth of the economy and fossil fuel consumption, the concentration of air pollutants has increased significantly and the air pollution problem is no longer limited to small areas. We conduct statistical analysis with the actual data related to air quality that covers the entire of South Korea using R and Python. Some factors such as SO_2, CO, O_3, NO_2, PM_{10}, precipitation, wind speed, wind direction, vapor pressure, local pressure, sea level pressure, temperature, humidity, and others are used as covariates. The main goal of this paper is to predict air quality index (AQI) spatio-temporal data. The observations of spatio-temporal big datasets like AQI data are correlated both spatially and temporally, and computation of the prediction or forecasting with dependence structure is often infeasible. As such, the likelihood function based on the spatio-temporal model may be complicated and some special modelings are useful for statistically reliable predictions. In this paper, we propose several methods for this big spatio-temporal AQI data. First, random effects with spatio-temporal basis functions model, a classical statistical analysis, is proposed. Next, neural networks model, a deep learning method based on artificial neural networks, is applied. Finally, random forest model, a machine learning method that is closer to computational science, will be introduced. Then we compare the forecasting performance of each other in terms of predictive diagnostics. As a result of the analysis, all three methods predicted the normal level of $PM_{2.5}$ well, but the performance seems to be poor at the extreme value.

Keywords: AQI data, descriptive model, neural networks model, random effect, random forest, spatio-temporal basis function, spatio-temporal big data

1. Introduction

The air pollution problem is frequently observed and can not be easily solved in most countries. In recent years, South Korea is facing a big problem with a high concentration of air pollutants, mostly the fine dust problem during the winter season. For simplifying public information about this air pollution problem, the air quality index (AQI) is defined as a standardized formula for how polluted the current air is and is used for data analysis. The main pollutants are fine particles, respirable particulate matter, sulfur dioxide, nitrogen dioxide, ozone, and carbon monoxide.

Many researchers have analyzed the AQI data. Baran (2019) predict the air quality index (AQI) by the extreme learning machines (ELM) algorithm. Yue and Li (2020) analyzed the weights of air pollutants on AQI and the interaction among them. Jiang (2021) calculated the correlation between six factors such as $PM_{2.5}$, PM_{10}, SO_2, NO_2, O_3, and CO concentration using the Pearson correlation coefficient,

This research was supported by the Basic Research Program through the National Research Foundation of Korea (NRF) funded by the MSIT (NRF-2020R1A4A1018207).

[1] Corresponding author: Department of Statistics, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Korea. E-mail: lkw1718@snu.ac.kr

Published 31 March 2023 / journal homepage: http://csam.or.kr

and analyzed them using multiple linear regression analysis. Also, Wang *et al.* (2022) presents an AQI prediction model based on convolution neural networks (CNN) and improved long short-term memory (ILSTM), named CNN-ILSTM.

In this paper, we apply three prediction models to the AQI data and compare their performances using various statistical measures. The goal of the three models is to predict or forecast $PM_{2.5}$ in South Korea using the covariates. The first model is the random effects model with spatio-temporal basis functions. This model is the most classical method of the three models for obtaining and inferring information from spatio-temporal data based on statistical theory. The second model is neural networks model which is applicable to spatio-temporal data. The neural networks model has the advantage of being able to learn complex patterns of the data. Finally, we consider the random forest model. The random forest model often used in the field of artificial intelligence. The random forest is a supervised learning algorithm often considered one of the best off-the-shelf machine learning algorithms for classification and regression. It is robust to outliers and known to be free from overfitting problems (Herrera *et al.*, 2019).

For analysis, 50 training datasets and 50 test datasets are provided. The three methods introduced above are then performed on each dataset to obtain the prediction values at the given spatio-temporal locations. The methods proposed in this paper are implemented in R and Python, and the source code is available from GitHub repository (`https://github.com/kke712/Prediction-of-ST-AQI-data`).

The paper is organized as follows. In Section 2, we describe the basic idea of the methods and illustrate the approaches. This is followed by explanations for three different models. Section 3 provides an AQI data description and analysis results. Some discussions are given in Section 4.

2. Methods

In this section, we describe three prediction models considered in this paper.

2.1. Random effects with spatio-temporal basis functions

In the AQI dataset, observations are spatio-temporal data that are closely located with each other and have a dependence structure in both space and time. We adopt a descriptive way to handle the spatio-temporal dependence model whose basic structure follows models in Wikle *et al.* (2019).

In the descriptive spatio-temporal statistical model, we assume the additive structure. This means the dependent variable can be decomposed into the latent spatio-temporal process and measurement error. The true process can be considered as the spatio-temporal fixed effect due to the covariates plus the random process which is spatio-temporally dependent. We call this a descriptive approach since this method describes the dependency of the random process.

The descriptive model has the advantage of a simple decomposing structure, but the exact likelihood of this descriptive method becomes unstable and infeasible in the big-n problem since it involves computing quadratic forms and determinants associated with a large covariance matrix (Bakar and Kokic, 2017). This problem can be alleviated by the basis function approach.

In the spatio-temporal random effect model, we assume a process term with dependence and an additional measurement error term. In particular, the process is decomposed into fixed effect terms and random process terms. Assume that at each time $t_i \in \{t_1, \ldots, t_T\}$ there are $n_i := n_{t_i}$ observations.

The spatial locations at time points t_i's are given as $\{s_{i1}, \ldots, s_{in_i}\}$. The observation is then represented by $\{Z(t, s) : t \in A_1, s \in A_2\}$, where the time index t is in the temoral index set A_1 of interest (in our case, one-dimensional real line) and s is location in the spatial area A_2 (in our case, a subset of two-dimensional Euclidean space). Let

$$\boldsymbol{Z} = \left(Z\left(t_1, s_{11}\right), Z\left(t_1, s_{12}\right), \ldots, Z\left(t_1, s_{1n_1}\right), \ldots, Z\left(t_T, s_{T1}\right), \ldots, Z\left(t_T, s_{Tn_T}\right)\right)'$$

be the vector of the dependent variable such as $PM_{2.5}$. Note that the number of irregular spatial observations can be different at each time. Recall that the main goal is to find a statistically optimal prediction value for a latent random spatio-temporal process at some new space-time location (t^*, s^*). If $t^* < t_T$, it becomes a smoothing problem, while if $t^* > t_T$ it is a forecasting problem.

Suppose that the data can be represented in terms of the potential spatio-temporal process and a measurement error:

$$Z\left(t_i, s_{ij}\right) = Y\left(t_i, s_{ij}\right) + \epsilon\left(t_i, s_{ij}\right), \quad j = 1, \ldots, n_i;\ i = 1, \ldots, T,$$

where the error $\{\epsilon(t_i, s_{ij})\}$ is *iid* mean-zero measurement error with the variance σ_ϵ^2. We assume that this error term is independent of Y. In other words, the data are noisy observations of the latent process Y in a finite set of locations.

We further assume that the latent process at the time t and the location s is given by

$$Y(t, s) = \mu(t, s) + \eta(t, s), \tag{2.1}$$

where $\mu(t, s)$ is the non-random mean process, and $\eta(t, s)$ is zero-mean spatio-temporal random process. We also assume $\mu(t, s) = x(t, s)'\boldsymbol{\beta}$, where $x(t, s)$ is a observed covariates at time t and location s, and $\boldsymbol{\beta}$ is an unknown p-dimensional vector of the corresponding coefficients.

The best linear unbiased predictor, $\hat{Y}(t^*, s^*)$, that minimizes the mean square prediction error (MSPE) $E(Y(t^*, s^*) - \hat{Y}(t^*, s^*))^2$ is called the kriging predictor. We further assume that the underlying process is a Gaussian process and the error term follows a Gaussian distribution. A stochastic process is called a Gaussian process if its finite dimensional distributions are multivariate normal distributions. It is denoted by $Y(\ell) \sim \mathrm{GP}(\mu(\ell), c(\cdot, \cdot))$, where $\mu(\ell) = E(Y(\ell))$ is a mean function and $c(\ell, \ell') = \mathrm{cov}(Y(\ell), Y(\ell'))$ is a covariance function, and ℓ, ℓ' are the locations in the area A (in our case, space-time locations).

In the spatio-temporal kriging, time is considered as an additional different dimension. The covariance function is the covariance between the two space-time locations. Let $\mathrm{cov}(\boldsymbol{Y}) \equiv \boldsymbol{C}_y = \boldsymbol{C}_\eta$, $\mathrm{cov}(\boldsymbol{\epsilon}) \equiv \boldsymbol{C}_\epsilon$, and $\mathrm{cov}(\boldsymbol{Z}) \equiv \boldsymbol{C}_z = \boldsymbol{C}_y + \boldsymbol{C}_\epsilon$. Then the joint distribution of $\boldsymbol{Z}$ and $Y(t^*, s^*)$ where (t^*, s^*) is a new space-time location is given by

$$\begin{bmatrix} Y(t^*, s^*) \\ \boldsymbol{Z} \end{bmatrix} \sim \mathrm{GP}\left(\begin{bmatrix} x(t^*, s^*)' \\ \boldsymbol{X} \end{bmatrix}\boldsymbol{\beta}, \begin{bmatrix} c^{**} & \boldsymbol{c}^{*'} \\ \boldsymbol{c}^* & \boldsymbol{C}_z \end{bmatrix}\right),$$

where $\boldsymbol{c}^{*'} \equiv \mathrm{cov}(Y(t^*, s^*), \boldsymbol{Z})$, $c^{**} \equiv \mathrm{var}(Y(t^*, s^*))$, $N \equiv \sum_{i=1}^{T} n_i$ and $\boldsymbol{X}$ is $N \times (p + 1)$ design matrix.

Since the joint distribution of $Y(t^*, s^*)$ and $\boldsymbol{Z}$ is a multivariate normal distribution, the conditional distribution of $Y(t^*, s^*)$ is given by Johnson and Wichern (2013)

$$Y(t^*, s^*) \mid \boldsymbol{Z} \sim \mathrm{GP}\left(x(t^*, s^*)'\boldsymbol{\beta} + \boldsymbol{c}^{*'}\boldsymbol{C}_z^{-1}(\boldsymbol{Z} - \boldsymbol{X}\boldsymbol{\beta}),\ c^{**} - \boldsymbol{c}^{*'}\boldsymbol{C}_z^{-1}\boldsymbol{c}^*\right).$$

The kriging predictor is the mean of the above conditional distribution when the processes are Gaussian. That is, $\hat{Y}(t^*, s^*) = x(t^*, s^*)'\boldsymbol{\beta} + \boldsymbol{c}^{*'}\boldsymbol{C}_z^{-1}(\boldsymbol{Z} - \boldsymbol{X}\boldsymbol{\beta})$. This means that the kriging predictor can be decomposed into the residual term and the marginal mean of the new location. These residuals are weighted as much as $\boldsymbol{c}^{*'}\boldsymbol{C}_z^{-1}$.

Since $\boldsymbol{\beta}$ is unknown, its estimator needs to be used in the kriging predictor and the generalized least square estimator can be used. Thus, the predictor becomes

$$\hat{Y}(t^*, s^*) = x(t^*, s^*)'\hat{\boldsymbol{\beta}}_{\mathrm{gls}} + \boldsymbol{c}^{*'}\boldsymbol{C}_z^{-1}\left(\boldsymbol{Z} - \boldsymbol{X}\hat{\boldsymbol{\beta}}_{\mathrm{gls}}\right),$$

where $\hat{\boldsymbol{\beta}}_{\mathrm{gls}} = (\boldsymbol{X}'\boldsymbol{C}_z^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{C}_z^{-1}\boldsymbol{Z}$. This is called the universal kriging predictor.

In practice, $\boldsymbol{C}_z$, $\boldsymbol{c}^*$ and c^{**} are also unknown. One simple approach is to parameterize the covariance function and use the maximum likelihood estimator to construct $\hat{\boldsymbol{C}}_z$, $\hat{\boldsymbol{c}}^*$ and $\hat{c}^{**}$. On the other hand, it is difficult to handle the spatio-temporal covariance matrix obtained from parametrization of the valid covariance function when the sample size is large.

This problem can be dealt with by introducing spatio-temporal basis functions. That is, the process model (2.1) can be rewritten in terms of fixed effects $\boldsymbol{\beta}$ and random effects $\{\gamma_k : k = 1, \ldots, n_\gamma\}$,

$$Y(t, s) = x(t, s)'\boldsymbol{\beta} + \eta(t, s) = x(t, s)'\boldsymbol{\beta} + \sum_{k=1}^{n_\gamma} \phi_k(t, s)\,\gamma_k + \nu(t, s).$$

Here, $\{\phi_k(t, s) : k{=}1, \ldots, n_\gamma\}$ is a specified spatio-temporal basis function corresponding to the location (t, s) and $\nu(t, s)$ captures a fine scale spatio-temporal random effect that can not be represented by the basis function.

Assuming $\boldsymbol{\gamma} \equiv (\gamma_1, \ldots, \gamma_{n_\gamma})' \sim \mathrm{GP}(\boldsymbol{0}, \boldsymbol{C}_\gamma)$ where $n_\gamma << N$, the N-dimensional vector $\boldsymbol{Y}$ is given by

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\Phi}\boldsymbol{\gamma} + \boldsymbol{\nu}, \tag{2.2}$$

where $\boldsymbol{\Phi}$ is the spatio-temporal basis function matrix whose k^{th} column is the k^{th} basis function, $\phi_k(\cdot, \cdot)$. Also, we assume $\boldsymbol{\nu} \sim \mathrm{GP}(\boldsymbol{0}, \boldsymbol{C}_\nu)$ and $\boldsymbol{\nu}$ has the same spatio-temporal order as $\boldsymbol{Y}$. The marginal distribution of $\boldsymbol{Y}$ is $\mathrm{GP}(\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{\Phi}\boldsymbol{C}_\gamma\boldsymbol{\Phi}' + \boldsymbol{C}_\nu)$, where $\boldsymbol{C}_y = \boldsymbol{\Phi}\boldsymbol{C}_\gamma\boldsymbol{\Phi}' + \boldsymbol{C}_\nu$. The spatio-temporal basis function explains the dependency of this model. The advantage of this method is that the high-dimensional problem now only focuses on n_γ of random effects, in that the inverse matrix is then n_γ- dimensional matrix. Assuming the model (2.2) and $\boldsymbol{V} \equiv \boldsymbol{C}_\nu + \boldsymbol{C}_\epsilon$, $\boldsymbol{C}_z = \boldsymbol{\Phi}\boldsymbol{C}_\gamma\boldsymbol{\Phi}' + \boldsymbol{V}$. By the well-known Sherman-Morrison-Woodbury matrix identity (Searle, 2017), the inverse matrix can be calculated by

$$\boldsymbol{C}_z^{-1} = \boldsymbol{V}^{-1} - \boldsymbol{V}^{-1}\boldsymbol{\Phi}\left(\boldsymbol{\Phi}'\boldsymbol{V}\boldsymbol{\Phi} + \boldsymbol{C}_\gamma^{-1}\right)^{-1}\boldsymbol{\Phi}'\boldsymbol{V}^{-1}.$$

If we assume that $\boldsymbol{V}$ is a diagonal matrix, this inverse matrix is easy to calculate. Since a calculation of a matrix $\boldsymbol{C}_\gamma^{-1}$ is low-dimensional, computational costs can be significantly reduced.

The choice of basis function is influenced by the type of the residual structure and random effects. The tensor product basis function is considered in our case, where the spatio-temporal basis function

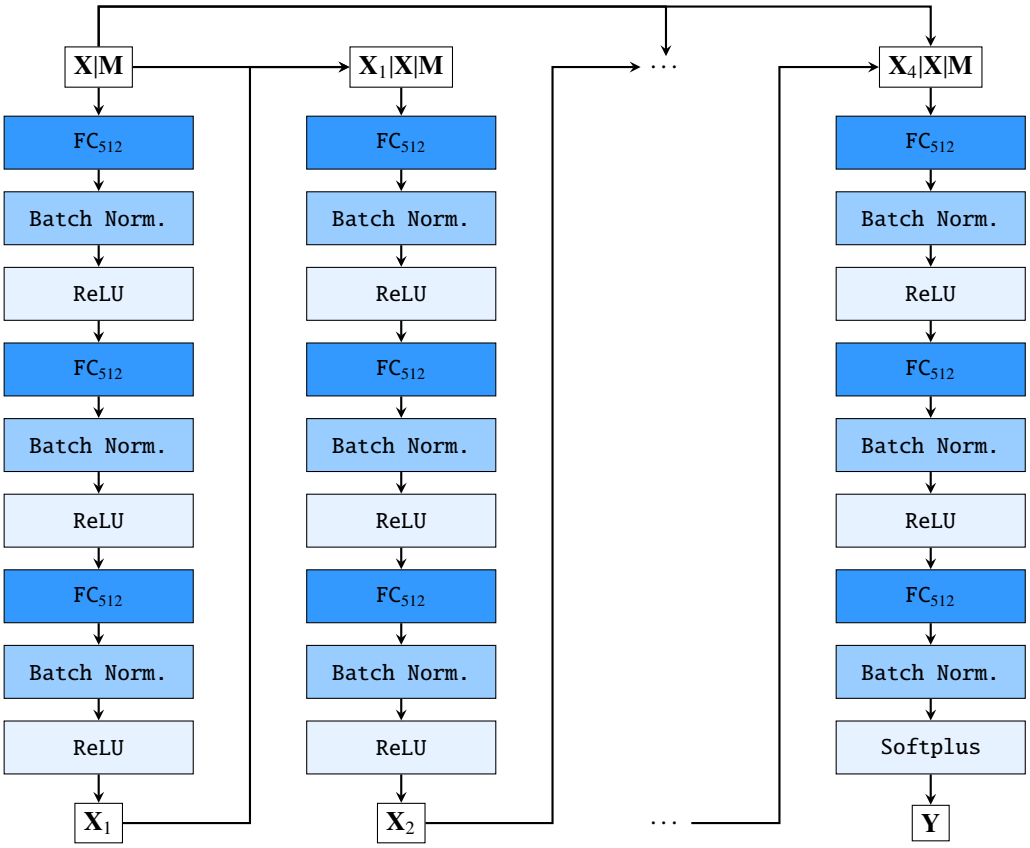


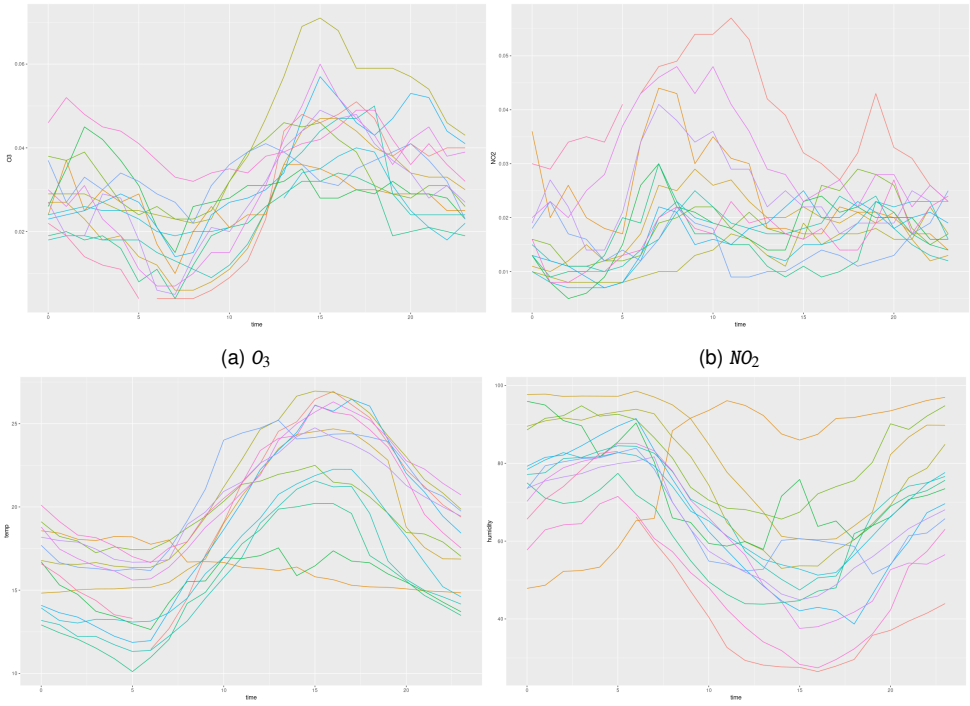
Figure 1: *Structure of the neural networks model. 'FC' means fully connected layer and 'Batch Norm.' means batch normalization layer. $X_1 \mid X_2$ is a concatenated matrix of two matrices X_1 and X_2.*

is the product of the spatial basis function and the temporal basis function. We let $n_\gamma = 2664$, where we use a bisquare basis function for the spatial basis function and a Gaussian basis function for the temporal basis function. In particular, the spatial basis function is generated by two resolutions. The coefficient γ of each resolution is assumed to be independent, and the covariance within the resolution decreases exponentially as the distance between the centers of the basis function increases. This model is implemented in R statistical software.

2.2. Neural networks model

The neural network is a machine learning technique that combines linear transformations and multiple nonlinear transformations to analyze the complex structure of data. The neural network is powerful especially when dealing with unstructured data, from visual data to natural language processing. With the neural network, the end-to-end learning is possible based on the input and output alone, without explicit intermediate steps. These properties have played important roles in the rapid development of numerous artificial intelligence applications.

We consider a simple fully connected network model. To deal with missing values in the dataset, we



(a) O_3 (b) NO_2

(c) `temp` (d) `humidity`

Figure 2: *Daily variation trend of* O_3, NO_2, `temp`, *and* `humidity`. *This shows the daily trend of variables at the particular point with latitude 35.8803 and longitude 128.5625 from* `2020-05-15` *to* `2020-05-29`. *We draw the lines in different colors according to the date. Since it is hourly data, 24 time points are used for each line. The x-axis is time and the y-axis is the value of each variable.*

introduce a mask matrix M as in Yoon *et al.* (2018). The matrix M has a value of 1 if the data is observed or 0 otherwise and we use the input matrix which is concatenated matrix of the data matrix and mask matrix.

A vanishing gradient problem arises when training neural networks model with gradient-based learning methods and backpropagation. The problem is that as the depth of the model increases, the gradient becomes smaller and it becomes difficult to change the value of the parameter. In the worst case, the model is no longer trained. To avoid the vanishing gradient problem, we adopt several techniques. We use a rectified linear unit (ReLU) function $\mathrm{ReLU}(x) = \max\{0, x\}$ as activation functions (Nair and Hinton, 2010) and the batch normalization layers (Ioffe and Szegedy, 2015), which include the process of adjusting the mean and variance in the network, to enable stable learning. In addition, we build a model with skip connection as in VGGNet (Simonyan and Zisserman, 2014). That is, we let the model use input data with the latest output periodically. For the last activation function, we use a softplus function $\mathrm{Softplus}(x) = \log(1 + \exp(x))$. Figure 1 shows the structure of the model.

2.3. Random forest

A decision tree introduced by Quinlan (1986) is a supervised learning method to solve classification and regression problems. The decision tree consists of nodes and all data points split into child nodes

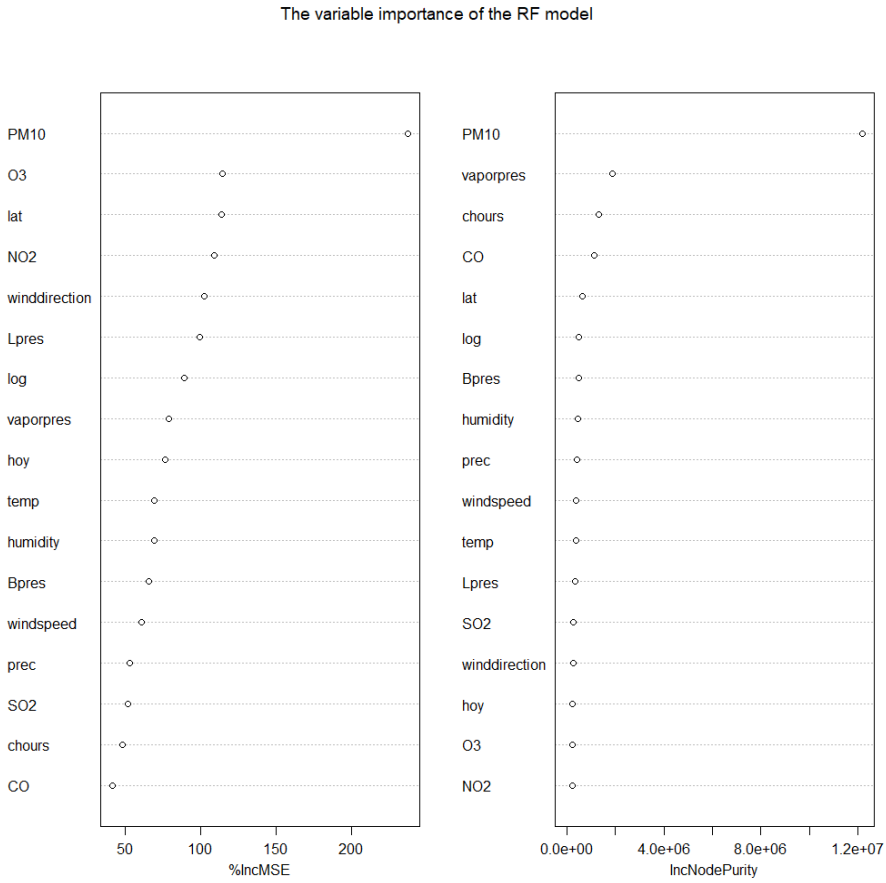


Figure 3: *The figure of the variable importance of the RF model. The left plot is computed using the mean decrease in accuracy and the right plot is computed using the mean decrease in node impurity.*

until arriving at terminal nodes through decision rules in the tree structure. This method is simple and convenient, but it is unable to generalize because the decision rules are set randomly and only use one tree to classify or predict.

Leo (2001) introduced random forest to solve this problem. The random forest is a combined method randomization skills with decision trees. More precisely, it makes an inference about the population using independent decision trees through randomization skills such as bagging or random node optimization. The random forest is more stable than a naive decision tree method and has more generalization power. The random forest is an ensemble model for the prediction or classification of response variables by combining many small trees obtained by applying the decision tree method to the bootstrapped sample. It has a good prediction performance despite its simplicity.

Variable importance is another strength of the random forest model (RF model). The RF model naturally provides variable importance measures when implemented to solve classification or regression problems. The RF model offers variable selection using variable importance moreover, can explain

Table 1: The table of the variables

Name	Description	Source	Unit
SO_2	Sulfur dioxide	AirKorea	ppm
CO	Carbon monoxide	AirKorea	ppm
O_3	Ozone	AirKorea	ppm
NO_2	Nitrogen dioxide	AirKorea	ppm
PM_{10}	Particulate Matter 10 (PM_{10})	AirKorea	$\mu g/m^2$
$PM_{2.5}$	Particulate Matter 2.5 ($PM_{2.5}$)	AirKorea	$\mu g/m^2$
prec	Precipitation	KMA	mm
windspeed	Wind speed	KMA	m/s
winddirection	Wind direction	KMA	arc°
vaporpres	Vapor pressure	KMA	hPa
Lpres	local pressure	KMA	hPa
Bpres	Sea level pressure	KMA	hPa
temp	Temperature	KMA	°C
humidity	Humidity	KMA	%
hoy	An hour of the year representing seasonality.		hours
chours	The cumulative hours representing long-term trends.		hours
lat	Latitude		°
log	Longitude		°

Names, brief descriptions, sources, and units of the variables are presented.

features, not like the other tree models. There are many ways to compute variable importance measures such as MDI (mean decrease in impurity) importance, permutation importance, and drop column importance. In this paper, the permutation importance is used which uses out-of-bag error. The detail of the method is given in in (Leo, 2001), and the result of variable importance is given in the Figure 3 of the Section 3.

The random forest is used in many fields and has an implementation in R, the `randomForest` package. The random forest doesn't have an implementation for the spatio-temporal data. We add two geographical variables(`lat`,`log`) and two temporal variables (`hoy` and `chours`) to reflect spatio-temporal characteristics of the data. Details are given in Section 3.1.

3. Experiments

3.1. Data

For the dataset for prediction, we combine two datasets: The air pollution data provided by AirKorea(`https://airkorea.or.kr`) and the meteorological data such as temperature and humidity from the Korea meteorological administration (KMA, `https://www.kma.go.kr/kma`). Both datasets were collected hourly.

A difficulty in combining the two datasets is that the observatories for air pollution data and meteorological data are different. To merge the two datasets from different sources, meteorological data observation values at air pollution data stations were interpolated through the inverse distance weighting (IDW) interpolation technique. We introduce some variables to supplement spatial and temporal elements as in Hengl *et al.* (2018). To represent temporality, we use two variables `hoy`, an hour of the observation point, and `chours`, the elapsed time since the first observation point `2018-01-01 00:00:00` (yyyy-mm-dd-hh:mm:ss) These variables mean an hour of the year representing seasonality and the cumulative hours representing long-term trends, respectively. We also use geographical variables `lat` (latitude) and `log` (longitude) to represent spatial elements. The list of variables is given

in the Table 1.

We use the data from April to May 2020 for model fitting. For performance comparison, we select 50 points of time randomly from May 1^{st} to May 31^{st}, 2020. For each time point, data from the time point to 11 hours later were used as test data, and data from 2 weeks before to the time point were used as training data. For instance, if a time point `2020-05-04-08:00:00` selected, the corresponding training data is the data observed between `2020-04-20-08:00:00` and `2020-05-04-07:00:00`, and the test data is the data observed between `2020-05-04-08:00:00` and `2020-05-04-19:00:00`.

We assume that the $PM_{2.5}$ at specific time point can be inferred from other meteorological variables over the preceding 12 hours. For every model, the explanatory variables were all variables except $PM_{2.5}$ for 12 hours, and the response variable was $PM_{2.5}$ in the last hour. For example, to predict $PM_{2.5}$ at `2020-05-04-15:00:00`, we use all variables but $PM_{2.5}$ such as SO_2 from `2020-05-04-03:00:00` to `2020-05-04-15:00:00` as explanatory variables. In a forecasting problem, since the future time values of explanatory variables are unknown, we replace the values of explanatory variables with the values of the observed data. For this, we set this replaced value as close to the actual value as possible. Specifically, O_3, NO_2, `temp`, and `humidity` have a daily trend as shown in Figure 2. In this case, we replace the values in the test data with the values of the train data from the day before. Since other variables except O_3, NO_2, `temp`, and `humidity` have no daily trend, these are replaced by the most recent observations. For instance, given train data observed between `2020-04-20-08:00:00` and `2020-05-04-07:00:00`, if we want to predict $PM_{2.5}$ at `2020-05-04-08:00:00`, the explanatory variables O_3, NO_2, `temp`, and `humidity` at `2020-05-04-08:00:00` are replaced by the the values of `2020-05-03-08:00:00`. The other variables except these four are replaced by the values of `2020-05-04-07:00:00`. All explanatory variables are standardized using the mean and standard deviation of each column of the training data.

3.2. Results

In this section, we predict $PM_{2.5}$ using the three methods discussed in Section 2. We fit the models using the training set and test the models by forecasting with the test set. We compare the prediction results of 1-step, 3-step, and 6-step ahead forecast for each methodology.

In the descriptive spatio-temporal basis function model (ST model), the EM algorithm is used to find the maximum likelihood estimates. We assume that the hyperparameter σ_ϵ is the standard error of the OLS regression. In the ST model, the spatial correlation is reflected as the coordinates. In order to compare under the same criteria as other models, two temporal variables (`hoy` and `chours`) also used as covariates. For the neural networks model (NN model) described in Figure 1, the width of each layer is 512, and the neural networks model is stacked so that there are 3 skip connections. We fit the model with the Python `PyTorch` package (Paszke *et al.*, 2019) and on a GPU server with one Nvidia RTX 3090. We used the `AdamW` optimizer (Loshchilov and Hutter, 2017) to train the neural networks model, and the `CosineAnnealingWarmRestarts` (Loshchilov and Hutter, 2016) scheduler to improve performance. The starting learning rate was 0.005, the epoch was 500, and the batch size was 1024.

In the RF model, we use the default value for hyperparameter settings in the R-package, `randomForest`. The RF model provides variable importance naturally and can be implemented in R as mentioned in Section 2.3. Figure 3 is the variable importance of the RF model. There are two types of importance measures. The left plot's measure is the mean decrease in accuracy and the right plot's measure is the mean decrease in node impurity. The left plot and the right plot show that PM_{10} is the most important

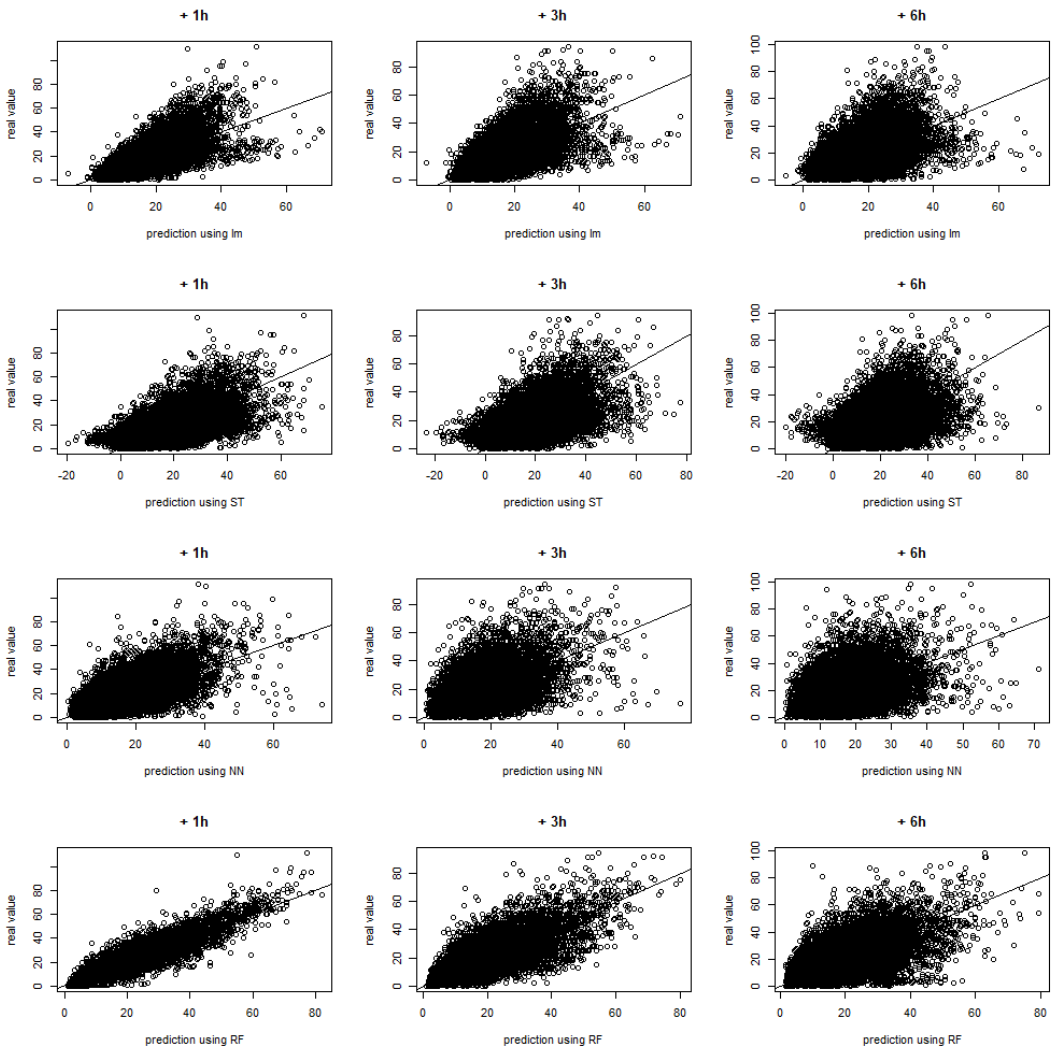


Figure 4: *Plot between prediction and real value of $PM_{2.5}$ using four methods a linear regression model as a control method, the ST model, the NN Model and the RF model from top to bottom. The linear regression model is named 'lm'. The left column of the plot is 1-step forecasting. The middle column of the plot is 3-step forecasting. The right column of the plot is 6-step forecasting.*

variable. Practically, the train and test datasets are 50. Thus, we show just the first train dataset's variable importance plot for demonstration. The rest of the other dataset's variable importance rank is similar to Figure 3, not precisely numerically identical.

In Figure 4, the scatter plots between the predicted values and the actual observations are given. Each row of Figure 4 is for a linear regression model, the ST model, the neural networks model, and the RF model, respectively. The linear regression model is the control method to compare with other models

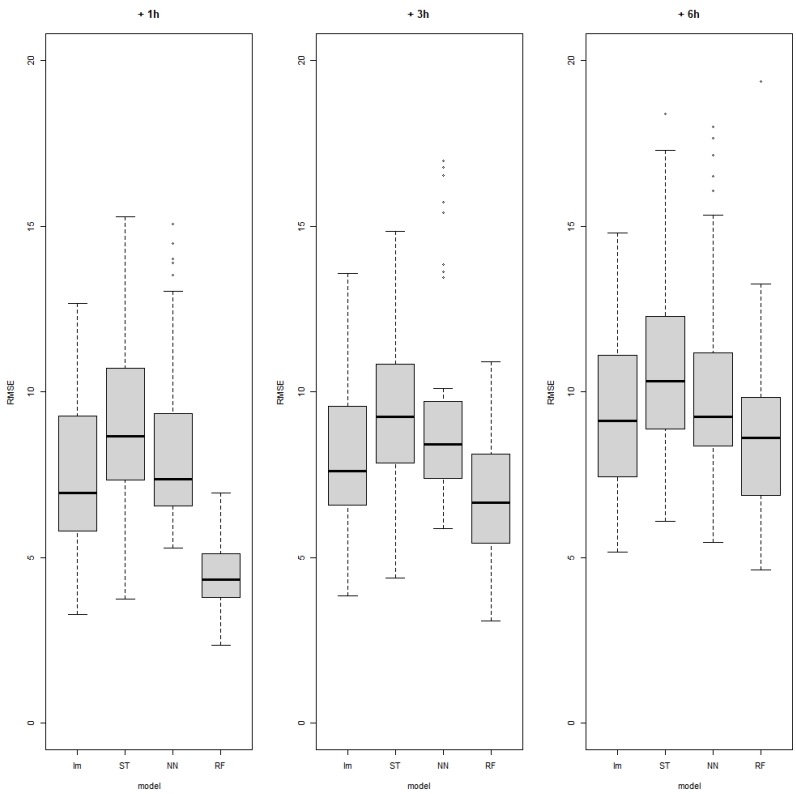


Figure 5: *Boxplots of RMSE using four methods. There are three plots. The left plot is 1-step forecasting, the middle one is 3-step forecasting, and the right one is 6-step forecasting. The step of forecasting is mentioned above each plot. In each plot, there are four boxplots of RMSE using four models. The models are a linear regression model, the ST model, the NN model and the RF model. The linear regression model is used as control method and named 'lm'.*

Table 2: (50%, 75%) quantile of RMSE using four methods

`Method`	(50%, 75%) & +1h	(50%, 75%) & +3h	(50%, 75%) & +6h
lm	(6.9495, 9.1587)	(7.6195, 9.4762)	(9.1246, 11.0667)
ST	(8.6712, 10.6362)	(9.2574, 10.8311)	(10.3301, 12.2183)
NN	(7.3651, 9.2230)	(8.4174, 9.6568)	(9.2467, 11.0664)
RF	(**4.3359**, **5.0957**)	(**6.6513**, **8.0957**)	**(8.6188, 9.8213)**

The methods are a linear regression model named 'lm', the ST model, the NN model and the RF model from top to bottom. Columns of table mean forecasting step. The second column is 1-step forecasting, the third is 3-step forecasting, and the fourth is 6-step forecasting form left to right.

and named 'lm' in this paper. The linear model implements in R, using the `lm` function. The columns of Figure 4 show 1-step, 3-step, and 6-step ahead forecast from the left to the right. The straight lines in Figure 4 represent $y = x$. The closer each point is to the straight line, the better prediction performance is. The ST model tends to overestimate response variables compared with other methods, and the predicted values of $\mathtt{PM}_{2.5}$ get smaller when the forecast is further away in time. Namely, the ST model predicts values by pulling them toward the mean. The predicted values using the NN model

Table 3: The range of categories based on the rule which is provided by Korea's Ministry of the Environment

Range	Category
$PM_{2.5} < 16$	good
$16 \leq PM_{2.5} < 36$	normal
$36 \leq PM_{2.5} < 76$	bad
$76 \leq PM_{2.5}$	very bad

Table 4: The table of specificity uses four methods a linear regression model, the ST model, the NN model, and the RF model according to the forecasting step

Step	Method	Category			
		Good	Normal	Bad	Very bad
+1h	lm	0.6917	0.8399	0.0967	0
	ST	0.6330	0.7054	0.2620	0
	NN	0.7390	0.6852	0.1511	0
	RF	**0.8621**	**0.8488**	**0.5677**	**0.1471**
+3h	lm	0.6527	**0.7994**	0.0930	0
	ST	0.6114	0.6890	0.2628	0
	NN	0.6847	0.6545	0.1205	0
	RF	**0.7669**	0.7619	**0.3886**	0
+6h	lm	0.5915	**0.7352**	0.0685	0
	ST	0.5602	0.6651	0.1954	0
	NN	0.6453	0.5997	0.0663	0
	RF	**0.6618**	0.6920	**0.2709**	0

are widespread in terms of the straight line, and they get more widespread as the forecasting step is further. However, the values are almost on a straight line not like the ST model. The RF model's prediction performance for 1-step forecasting is the best because points are located near the straight line while it is getting widespread from the left to the right in the fourth row of Figure 4. Indeed, it can be seen that the difference in the spread is the most severe among the three models.

To compare the performance of three prediction methods, we use

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum (y_i - \hat{y}_i)^2}$$

as a performance measure. Figure 5 displays boxplots of RMSE and Table 2 shows (50%, 75%) quantile of RMSE. In Figure 5, each plot means 1-step, 3-step, and 6-step forecasting from left to right. Also, each plot has four boxplots using the linear regression model, the ST model, the NN model, and the RF model from left to right. The NN model's RMSE is larger than the linear regression model though it's quantile range is more narrow than the control method. The RF model is the best in terms of RMSE and the interquartile range of it. The RMSE values of the linear regression model and the three methods increased by forecasting further step. Still the NN model's RMSE maintain short interquartile range, and the RF model has the smallest RMSE value.

It is primary to predict what category $PM_{2.5}$ is belong to for weather forecast, although it is also essential to predict exact value. For this reason, the $PM_{2.5}$ values are divided into categories based on a rule and the performance compared. In this paper, we divide predicted values into 4 categories: Good, normal, bad, and very bad based on the rule which is provided by Korea's Ministry of the Environment and then evaluate the classification prediction performance. The rule is given in Table 3.

Table 4 shows specificity value using the control model and the three models. Specificity is the ratio

Table 5: The table of F1 score using four methods according to forecasting step

Step	Method	Category			
		Good	Normal	Bad	Very bad
+1h	lm	0.7295	0.7697	0.1592	0
	ST	0.6389	0.6851	0.3124	0
	NN	0.6822	0.6939	0.2375	0
	RF	**0.8434**	**0.8463**	**0.6556**	**0.2564**
+3h	lm	0.6807	0.7395	0.1527	0
	ST	0.6177	0.6718	0.3037	0
	NN	0.6323	0.6625	0.1950	0
	RF	**0.7469**	**0.7608**	**0.4601**	0
+6h	lm	0.6139	0.6798	0.1121	0
	ST	0.5757	0.6373	0.2330	0
	NN	0.5916	0.6095	0.1107	0
	RF	**0.6519**	**0.6802**	**0.3417**	0

of real category to correctly predicted category. It is one of the measures to compare the performance when the response variable is categorical variable. Table 4's first column means forecasting step. The step goes further from top to bottom. Table 4's method column means the control model and all three models, and category column displays the specificity values according to all four categories. For instance, when 1-step forecasting in good category using the ST model, 63 percent of the sample in the good category is predicted correctly. The ST model's specificity is not greater than other methods though not different according to the forecasting step. The NN model can not classify the bad category still provide good performance with the good and normal categories. According to the Table 4, the RF model is good at prediction compared with the other two methods notably, when 1-step forecasting it classifies the very bad category. All of the model's ratios are getting worse when the forecasting step is further also when the category goes to 'very bad'. Above all they can not detect the very bad category except the RF model when 1-step forecasting.

It is insufficient to use only specificity when assess the categorical response variable's performance. Accordingly the F1 score (Powers, 2020) is used to analyze classification performance numerically. The F1 score is the harmonic average between precision and recall called specificity. Moreover it is valid to use the F1 score because the $PM_{2.5}$ data is imbalanced. Table 5 has the same structure as Table 4 but not the value. The ST model's F1 score is not bad as well as not that different according to the forecasting step. Moreover, it is good enough in the bad category for 3-step and 6-step forecasting compared with the RF model which is the best performance. The NN model's F1 score also does not change significantly according to the forecasting step. The RF model has the largest F1 score in Table 5, and remarkably offer classification of the very bad category. Indeed all four models decreased the F1 score when predicting further steps.

4. Discussion

In this paper, we proposed three prediction methods and related algorithms for AQI data. For the spatio-temporal model, it is common to assume that observations are spatio-temporally correlated over the study locations. To incorporate this correlation and curse of dimensionality, the random forest model, random effects with spatio-temporal basis functions model, and neural networks model are considered. These methods can be applied to interpolate the prediction of the new space-time location.

The results show that the ST model tends to overestimate $PM_{2.5}$ compared with the other methods. In Section 3, the RF model has the best performance among the three methods. That's because of the RF model's robustness to noise and overfitting (Leo, 2001) which is mentioned in Section 1. However, we can see that the variance of the prediction increases over time. This is not surprising, because prediction performance decreases as time moves away from the training set in all of the proposed methods. Also, Tables 4 and 5 suggest that the proposed estimates are well predicted in the normal stage, but the prediction performance decreases as it goes to the extreme stages.

The descriptive spatio-temporal model with basis functions is based on the random effect model with spatio-temporal basis functions. Its main goal is to reduce the computation costs due to the large covariance matrix. On the other hand, this method has a limit when applied to real data. For example, the structures of the basis function have to be chosen well. For the spatial process model, this choice is usually not that critical since the structures of the basis function do not significantly affect spatial dependence. However, if we consider the spatio-temporal model, the decision of the basis functions leads to different results. In addition, the descriptive model also tends to regress to the average value even at the extreme concentration of $PM_{2.5}$. This seems to be because the model setting itself is based on the residual of the regression equation.

The neural networks model have a more complex form than existing machine learning models. This means that the model is prone to be overfitted, and these problems become frequent when the training data size is small. During model training, the RMSE on the training data had a very small value. However, as the results above, the performance of the neural networks model is inferior to that of other models. To solve this problem, we can consider increasing training data size or applying regularization techniques.

The random forest model has two limits. First, random forest is hard to be implemented when the categorical predictor has too many categories. So the number of categories should be reduced to use the random forest as a preprocessing in that data structure. Second, a dataset for prediction must have values of covariates used to learn a random forest. This is the difference between the neural networks model introduced in Section 2.2. The random forest needs the values of covariates for forecasting, unlike the neural networks model. Therefore, the prediction performance is affected by how the covariate's values are set. Since the purpose of our study is to efficiently predict spatio-temporal big data rather than spatio-temporal interpolation, spatial and temporal effects are simply assumed to be fixed effects. However, it can also be considered as the random effects with spatio-temporal interaction term. Therefore, this paper can be used as the base for more complex modeling of AQI data in the future.

In general terms, our analysis results indicate that the proposed methods used in AQI data are successful in predicting the normal concentration but still have problems such as under/overestimation. We suggest that the forecasting performance derived by the three methods will be improved if more sufficient observations for more periods of time are used when calculating. We also note that the three methods in Section 2 are not completely free from the computation costs. Some possible approaches to address these issues have to be developed.

Acknowledgement

This research was supported by the Basic Research Program through the National Research Foundation of Korea (NRF) funded by the MSIT (NRF-2020R1A4A1018207).

References

Bakar KS and Kokic P (2017). Bayesian Gaussian models for point referenced spatial and spatio-temporal data, *Journal of Statistical Research*, **51**, 17–40.

Baran B (2019). Prediction of air quality index by extreme learning machines, In *Proceedings of International Artificial Intelligence and Data Processing Symposium (IDAP)*, Malatya, Turkey, 19079408, Available from: http: doi.org/10.1109/IDAP.2019.8875910

Herrera VM, Khoshgoftaar TM, Villanustre F, and Furht B (2019). Random forest implementation and optimization for big data analytics on LexisNexis's high performance computing cluster platform, *Journal of Big Data*, **6**, 1–36.

Hengl T, Nussbaum M, Wright MN, Heuvelink GB, and Gräler B (2018). Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables, *PeerJ*, **6**, e5518, Available from: https://doi.org/10.7717/peerj.5518

Ioffe S and Szegedy C (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." International conference on machine learning, pmlr, 2015.

Jiang W (2021). The data analysis of Shanghai Air Quality Index based on linear regression analysis, *Journal of Physics: Conference Series*, **1813**, 012031, Available from: https://doi.org/10.1088/1742-6596/1813/1/012031

Johnson RA and Wichern DW (2013). *Applied Multivariate Statistical Analysis*, Pearson Educated Limited Harlow, England.

Leo B (2001). Random forests, *Machine Learning*, **45**, 5–32.

Loshchilov I and Hutter F (2016). SGRD: Stochastic gradient descent with warm restarts, Available from: arXiv preprint arXiv:1608.03983

Loshchilov I and Hutter F (2017). Decoupled weight decay regularization. arXiv preprint, Available from: arXiv:1711.05101

Nair V and Hinton GE (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814.

Paszke A, Gross S, Massa F *et al.* (2019). Pytorch: An imperative style S, high-performance deep learning library, *Advances in Neural Information Processing Systems*, **32**, 8024–8035.

Powers DMW (2020). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation, *International Journal of Machine Learning Technology*, **2**, 37–63, Available from: https://arxiv.org/abs/2010.16061

Quinlan R (1986). Induction of decision trees, *Machine Learning*, **1**, 81–106.

Searle SR (2017). *Matrix Algebra Useful for Statistics*, Wiley Hoboken, New Jersey.

Simonyan K and Zisserman A (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*, Available from: https://arxiv.org/abs/1409.1556

Wang J, Li X, Jin L, Li J, Sun Q, and Wang H (2022). An air quality index prediction model based on CNN-ILSTM, *Scientific Reports*, **12**, 8373, Available from: http://doi.org/ 10.1038/s41598-022-12355-6

Wikle CK, Zammit-Mangion A, and Cressie N (2019). *Spatio-temporal Statistics with R*, CRC Press, Taylor & Francis Group, Florida.

Yoon J, Jordon J, and van der Schaar M (2018). Gain: Missing data imputation using generative adversarial nets, *International Conference on Machine Learning*, **80**, 5689–5698.

Ma H, Yue S, and Li J (2020). Air quality evaluation method based on data analysis, In *Proceedings of 2020 39th Chinese Control Conference (CCC)*, Shenyang, China, 3162–3167.

Received July 21, 2022; Revised January 13, 2023; Accepted January 15, 2023