Pol. J. Environ. Stud. Vol. 32, No. 2 (2023), 1419-1431
DOI: *10.15244/pjoes/157072* ONLINE PUBLICATION DATE: 2023-02-09

Original Research

Comparison of Statistical and Deep Learning Methods for Forecasting $PM_{2.5}$ Concentration in Northern Thailand

Weerinrada Wongrin, Kuntalee Chaisee, Kamonrat Suphawan*

Department of Statistics, Faculty of Science, Chiang Mai University, Chiang Mai 50200 Thailand

Received: 6 July 2022
Accepted: 1 December 2022

Abstract

This study applies statistical methods and deep learning techniques to forecast the daily average $PM_{2.5}$ concentration in northern Thailand, where the concentration is usually high and exceeds the safe level. The data used in the analysis are collected from January 2018 to December 2020 from 16 air monitoring stations. The statistical methods used are Holt-Winters exponential smoothing (ETS), autoregressive integrated moving average (ARIMA), and dynamic linear model (DLM). The deep learning techniques considered in this study are the recurrent neural network (RNN) and long-short term memory (LSTM). To compare the predictive performance of both methods, we use the root mean square error (RMSE). The result indicates that statistical methods, especially ARIMA, perform better than the deep learning techniques in most stations. Moreover, LSTM tends to provide higher accuracy than the RNN, especially with more number of nodes.

Keywords: $PM_{2.5}$ concentration, statistical methods, deep learning techniques

Introduction

Over the past several years, Thailand, especially in the northern region, has suffered from high particulate matter (PM) levels, particularly $PM_{2.5}$ during the dry season. Many northern provinces have an average of $PM_{2.5}$ concentration exceeding the safe level according to the WHO guideline. An air quality monitoring station in many provinces reported a $PM_{2.5}$ level higher than 200, which is above health-hazard levels. Some provinces in the region exceed the safe level for several consecutive days affecting people's health. The short-term effects are eye, nose, throat, and lung irritation, coughing, sneezing, and shortness of breath. The long-term effects may be associated with increased lung cancer and cardiopulmonary diseases. The sources of PMs in the region are mainly caused by wildfire smoke and agricultural burning. The ability to anticipate the levels of $PM_{2.5}$ is crucial as it can guide the people in the area on how to avoid exposure to air pollutants. Many techniques, such as machine learning and regression models, have been used to study and forecast PM concentrations using air pollutants and meteorological data [1-3].

Another approach to be used is time series analysis which aims to provide information and predict outcomes from historical data. Many time series methods, such as exponential smoothing (ETS) and autoregressive

*e-mail: kamonrat.s@cmu.ac.th

integrated moving average (ARIMA), is broadly used to capture structures in time series data. It is constructed based on the dependent relationship between an observation and some number of lagged observations. Another statistical method often used in time series prediction is a state-space model. It is also a dynamic linear model (DLM), commonly used in complex time series modeling, including linear, nonlinear, non-stationary, structural changes, and irregular patterns. DLM and ARIMA are often used interchangeably in time series modeling. Generally, statistical modeling is a versatile method to apply in diverse fields for studying trends and forecasting, for example, prediction of environmental data [4-8], in epidemiology [9], and in finance [10].

Air pollution forecasting is one of the popular topics for statistical modeling, and there is much literature about the application. [11] used ARIMA to forecast ambient air pollutants, particularly O_3, NO, NO_2, and CO, in Delhi, India. They suggested that the model is appropriately applied to forecast the pollutants for short-term purposes. [12] adapted ARIMA to forecast air quality in Hong Kong. They concluded that statistical modeling is suitable for the short run and forecasting performance can be diverse depending on locations and timescales. [13] used the time series models ARIMA and Holt Winter models to forecast short-term concentrations of pollutants in Indonesia. For predicting CO, NO_2, and O_3, the Holt Winter model outperforms the ARIMA model, while ARIMA was better at predicting PM_{10} and SO_2 concentrations. [14] used ARIMA, ETS, and singular spectrum analysis (SSA) to forecast 24-hour average for PM_{10} concentrations in the most polluted cities in Turkey. They found that, overall, the SSA model showed stronger results than ETS and ARIMA, especially short-term forecasts. However, the ETS performed better in some areas, especially for the long-term forecast.

As statistical modeling uses collections of probability distributions and assumptions to generate sample data and make predictions, machine learning, on the other hand, concentrates on a prediction by using learning algorithms to find patterns based on available data. As a result, these two fields overlap significantly, as they can apply to deal with data and make a prediction. Deep learning and machine learning are often used interchangeably. However, deep learning is considered a more specialized and sophisticated machine learning algorithm. It uses a layered structure of algorithms called an artificial neural network (ANN) inspired by the human brain's biological neural network. A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. In other words, it means that it allows previous outputs to be used as inputs while having hidden states. As a result, it can perform as a time series model as the computation takes historical information into account. A long short-term memory (LSTM) is also an artificial recurrent neural network that is an architecture used in deep learning. Generally, LSTM is an extended RNN that uses cell state, input, output, and forget gates to store long-term dependencies to overcome vanishing gradient problems in typical RNNs. The applications of deep learning techniques for time series prediction and forecasting can be seen in [15-18]. Many researchers used machine learning algorithms to forecast $PM_{2.5}$ concentrations [19-21].

Both statistical and deep learning neural networks-based methods can provide predictive values for time series data. Many researchers have examined the performance of statistical methods compared with machine learning-based techniques [22-25]. The results indicated that both methods are comparable. However, there is no sufficient evidence to certify that statistical methods are superior to machine learning and vice versa. The accuracy of the methods mainly depends on the data. According to the literature, statistical and deep learning methods are good for forecasting air pollution in many areas. In this study, for the first time, statistical and deep learning techniques are compared to forecast the daily $PM_{2.5}$ concentration in Thailand. The study area is the northern region of Thailand, where the number of days a year in which the $PM_{2.5}$ exceeds the standard levels is more than the other regions. This causes great concern for public health in the community. In this study, we aim to compare the performance of statistical methods and deep learning techniques for forecasting the daily $PM_{2.5}$ concentration in northern Thailand.

Materials and Methods

Study Area and Data

The data used in this research is the daily average (24-hour average) of $PM_{2.5}$ concentration (micrograms per cubic meter, $\mu g/m^3$) in the northern region of Thailand collected and provided by the Pollution Control Department, Air Quality and Noise Management Bureau, The Ministry of Natural Resources and Environment of Thailand. According to the data source, there are 17 provinces classified in the northern region: Chiang Mai, Chiang Rai, Lampang, Lamphun, Mae Hong Son, Nan, Phrae, Phayao, Tak, Nakhon Sawan, Uttaradit, Phitsanulok, Sukhothai, Phetchabun, Kamphaeng Phet, Phichit and Uthai Thani. However, the air monitoring stations in Uttaradit, Phitsanulok, Sukhothai, Phetchabun, Kamphaeng Phet, Phichit, and Uthai Thani are relatively new installed, so we do not include these stations in the study. As a result, the data from 16 stations in 10 provinces are analyzed. The locations of the considered stations are shown in Fig. 1. The dataset contains the daily average of $PM_{2.5}$ concentration from January 2018 to December 2020. Therefore, the complete dataset consists of n = 1095. However, in some stations, the data have

Fig. 1. Location of stations.

not been recorded in early 2018 and contain some missing values. These missing data are handled prior to modeling by the moving average method. The summary statistics of the daily average of $PM_{2.5}$ concentration for each station are provided in Table 1.

The averages of $PM_{2.5}$ concentration are between 20 to 40 μg/m^3, and the medians are slightly lower. Most stations have standard variation between 20-35 μg/m^3, except station 53t and 73t, which show high variations. The highest mean, standard deviation, median, and maximum $PM_{2.5}$ concentration is found at station 73t, located in Chiang Rai, the most northern province in Thailand. Meanwhile the smallest mean, standard deviation, median, and maximum $PM_{2.5}$ concentration are at station 41t, located in Nakhon Sawan, the lower northern region. The time series plots and boxplots of the daily average $PM_{2.5}$ concentration are displayed in Fig. 2 and 3, respectively. They illustrate that all stations have a similar pattern, having seasonality with high concentration period from January to May, and low concentration period from June to December. Station 73t has the highest peak, whereas 41t shows less peak than other stations. The boxplot shows that the data in all stations are right skewed with a large number of extreme values.

In addition, we model with the first 80% of the data, denoted by the training set, and the remaining 20% is used for evaluating the models' performance, denoted by the testing set. Note that the testing set is in a low concentration period, from July to December 2020. The data are normalized before modeling using the min-max normalization technique. To validate the prediction performance of the models, the root mean squared error (RMSE) is presented.

Methods

In this section, we present brief statistical methods as well as deep learning methods that are to be employed in predicting the $PM_{2.5}$ concentration. For statistical methods, we adopt three techniques, including exponential smoothing, autoregressive integrated moving average, and dynamic linear modeling. For deep learning methods, we adopt the recurrent neural network and long-short term memory.

Holt-Winters Exponential Smoothing (ETS)

The predicted value at time t is calculated based on the first observation, y_1, through the most recent observation y_{t-1}. The level L_t, trend T_t, and season S_t are updated through updating equations with three smoothing parameters α, β, γ. The ETS method can be implemented with an "additive" structure or a "multiplicative" structure depending on the behavior of time series data. The equations for the additive model are as follows:

$$L_t = \alpha(y_t - S_{t-c}) + (1-\alpha)(L_{t-1} + T_{t-1}), \tag{1}$$

$$T_t = \beta(L_t - L_{t-1} + (1-\beta)T_{t-1}, \tag{2}$$

$$S_t = \gamma(y_t - L_t) + (1-\gamma)S_{t-c}, \tag{3}$$

$$\hat{y}_{t+k} = L_t + kT_t + S_{t-c-k}, \tag{4}$$

where y_t is the observed value of the time series at time t, c is the number of periods in the seasonal cycle, k is the number of periods in the forecast lead-time, and $\hat{y}_{t+k}$ is the forecast at k periods ahead. The equations for the multiplicative model with multiplicative seasonality are as follows:

$$L_t = \alpha\frac{y_t}{S_{t-c}} + (1-\alpha)\left(L_{t-1} + T_{t-1}\right), \tag{5}$$

$$T_t = \beta\left(L_t - L_{t-1}\right) + (1-\beta)T_{t-1}, \tag{6}$$

Table 1. Descriptive statistics of daily average of $PM_{2.5}$ concentration.

Code	Province	Starting	n	Missing	Mean	S.D.	Median	Min.	Max.
35t	Chiang Mai	1 Jan 18	1095	3	33.028	27.508	23.000	7.250	228.250
36t	Chiang Mai	1 Jan 18	1095	62	29.559	27.128	19.667	3.550	209.708
37t	Lampang	17 Oct 18	807	0	29.895	26.550	20.000	2.833	157.136
38t	Lampang	17 Oct 18	807	3	24.766	22.312	14.000	2.750	127.524
39t	Lampang	17 Oct 18	807	2	25.833	28.311	12.917	2.792	265.524
40t	Lampang	1 Jan 18	1095	12	28.292	23.503	19.542	3.357	151.042
41t	Nakhon Sawan	17 Oct 18	807	5	26.842	15.582	23.104	6.125	76.042
57t	Chiang Rai	18 Jul 18	897	1	29.300	33.139	17.917	3.708	254.333
58t	Mae Hong Son	21 Jul 18	894	2	27.471	41.294	10.783	1.833	270.792
67t	Nan	20 Jul 18	895	4	26.582	24.529	17.333	3.667	167.583
68t	Lamphun	18 Jul 18	897	23	29.021	21.510	22.500	3.056	182.958
69t	Phrae	17 Oct 18	807	3	29.711	26.135	21.688	3.083	157.708
70t	Phayao	17 Oct 18	807	117	27.496	26.777	18.542	3.458	245.500
73t	Chiang Rai	17 Oct 18	807	13	39.379	55.304	18.500	2.833	398.125
75t	Nan	1 Jan 18	1095	25	24.824	31.530	12.771	2.545	262.208
76t	Tak	1 Jan 18	1095	21	28.508	22.523	21.176	2.176	133.250

$$S_t = \gamma \frac{y_t}{L_t} + (1-\gamma) S_{t-c}, \tag{7}$$

$$\hat{y}_{\text{illegible}} = \left(L + kT\right) S_{-+}. \tag{8}$$

Autoregressive Integrated Moving Average (ARIMA)

A standard notation for the model is ARIMA (*p*, *d*, *q*), where *p* is the number of previous (lag) observations that related to the current observation, *q* is the number of previous (lag) errors that related to the current observation, and *d* is the degree of differencing between observation and previous observation, in order to make the time series stationary. The values of *p* and *q* can be determined using the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots. The specific ARIMA (*p*, *d*, *q*) model is written as

$$\begin{aligned}\hat{y}_t = c + \phi_1 y_{t-1} + \in_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \theta_1 \in_{t-1} \\ + \theta_2 \in_{t-2} + \cdots + \phi_q \in_{t-q} + \in_t\end{aligned} \tag{9}$$

where *c* is a constant, ϕ_1, ..., ϕ_p, θ_1, ..., θ_p are parameters to be estimated, and ϵ_t is an error term. The point forecast of *k*-step ahead forecasting in ARIMA model, given the observations up to time *t*, is denoted by $\hat{y}_{t+k}$. It can be obtained by replacing *t* with *t* + *k*, starting with *k* = 1. These steps are repeated for *k* = 2, 3, ... for all future periods required.

Dynamic Linear Model (DLM)

The DLM is a class of state space model when state transition and observation functions are set to be linear, with design matrices *F* and *G*, with Gaussian noises. A basic structure of the DLM is expressed as follows:

$$x_t = F x_{t-1} + w_t, \quad w_t \sim N(0, W_t), \tag{10}$$

$$y_t = G x_t + v_t, \quad v_t \sim N(0, V_t). \tag{11}$$

The x_t denotes the state of the system at time *t* and y_t denotes the corresponding observation. The w_t is a stochastic component representing a simulator noise, $w_t \sim N(0, W_t)$, and v_t denotes a stochastic measurement noise, $v_t \sim N(0, V_t)$. The four parameters *F*, *G*, W_t, V_t are used for completely determining posterior distributions of the states in the DLM.

The posterior distribution of the state at time *t* given the observations up to time *t*, $p(x_t \mid y_{1:t})$, where $y_{1:t} = \{y_1, ..., y_t\}$ is assumed to be Gaussian. Thus, recursive Bayesian filtering algorithms, or Kalman filter, can be used to compute the posterior mean, $m_{t|t}$, and variance, $C_{t|t}$,

$$x_t \mid y_{1:t} \sim N\left(m_{t|t}, C_{t|t}\right). \tag{12}$$

Consequently, the predictive distribution of the state $p(x_{t+k} \mid y_{1:t})$ and the predictive distribution of the

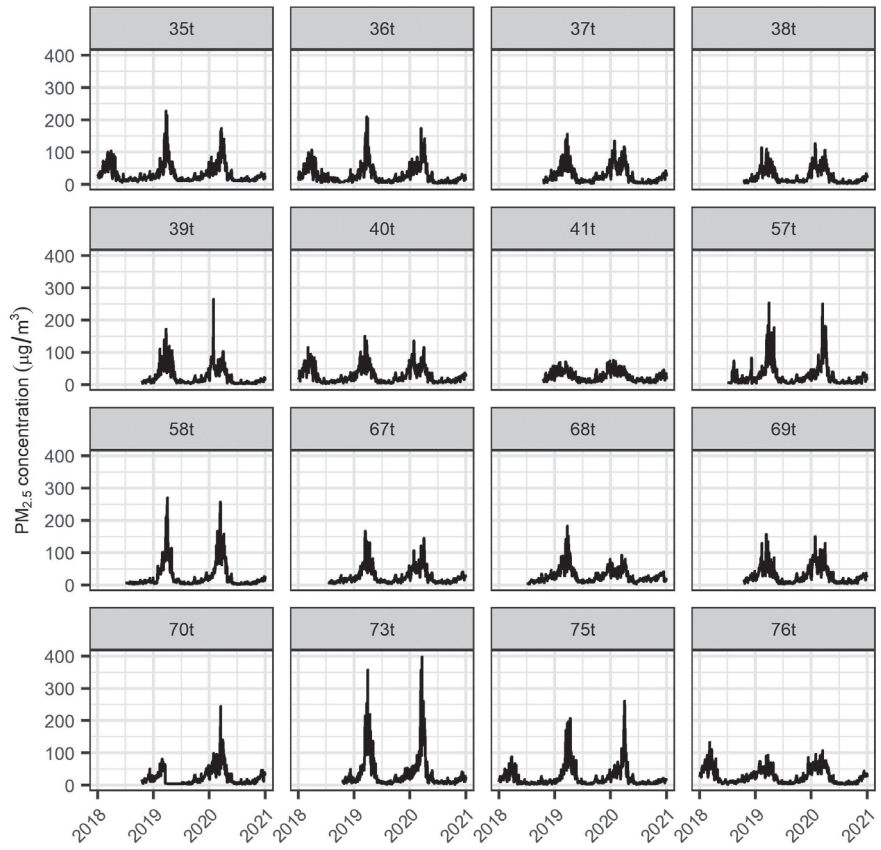

Fig. 2. Time series plots of the daily average of $PM_{2.5}$ concentration.

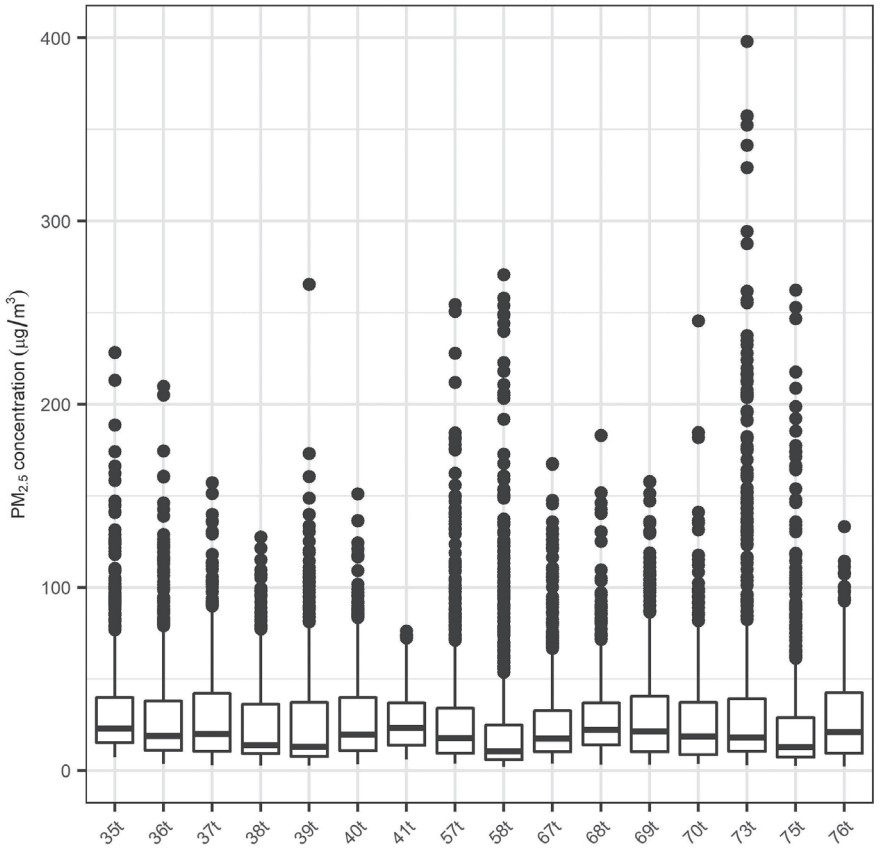

Fig. 3. Boxplots of the daily average of $PM_{2.5}$ concentration.

observation $p(\hat{y}_{t+k} \mid y_{1:t})$ at k-step ahead forecasting is also Gaussian. The predictive mean and variance can be obtained by propagating the previous forecasts

$$x_{t+k} \mid y_{1:t} \sim N(a_{t+k|t}, R_{t+k|t}), \tag{13}$$

$$\hat{y}_{t+k} \mid y_{1:t} \sim N(f_{t+k|t}, Q_{t+k|t}). \tag{14}$$

Recurrent Neural Network (RNN)

The RNN has the recurrent hidden layer that means the input of the hidden layer also contains the state of the previously hidden layer. In other words, the nodes of the hidden layer can be self-connected or interconnected. The RNN network can be express as the following equations:

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h), \tag{15}$$

$$y_t = \sigma_y(W_y h_t + b_y), \tag{16}$$

where h_t are the hidden layer vectors, x_t is the input vector, y_t is the output vector, W_y is the weighted matrix, U_h is the transition matrix, σ_h denotes the activation function in the hidden layer and σ_y denotes the activation function in the output vector, and b_h and b_Y are bias terms.

Long-Short-Term Memory (LSTM)

The LSTM network is an extension of RNN, which consists of a more complex structure in the hidden layer and has forget gates to store long-term dependencies to overcome vanishing gradient problems in typical RNN. The LSTM processes the data sequentially, passing the information as it propagates forward. The operations within the LSTM allow it to forget or keep the information. The LSTM network can be expressed as the following equations:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f), \tag{17}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i), \tag{18}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o), \tag{19}$$

$$\tilde{c}_t = \sigma_h(W_c x_t + U_c h_{t-1} + b_c), \tag{20}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t, \tag{21}$$

$$h_t = o_t \circ \sigma_h(c_t), \tag{22}$$

where f_t is the forget gate's activation vector, i_t is the input/update gate's activation vector, o_t is the output gate's activation vector, and $\tilde{c}_t$ is the cell input activation vector. The c_t and h_t are the cell and hidden layer vectors where ° denotes the element-wise product operator, and x_t is the input vector. The W and U are the weighted matrices of the input and recurrent connections, respectively. The σ_g and σ_h are the activation functions. The b_f, b_i, b_o and b_c are bias terms.

Packages and Programming

In this work, the application of statistical and deep learning analyses is performed using R statistical software programming. For statistical methods, the ETS is performed via a function *ets* in package forecast. The function automatically estimates the model parameters and uses the Akaike information criteria (AIC) to select an appropriate ETS model, then returns fitted values and predicted values at k-step ahead forecasting; the ARIMA uses *auto.arima* function in forecast package which returns the best ARIMA model according to the AIC, or the Bayesian information criterion (BIC) value, and the DLM uses package dlm which includes functions for maximum likelihood estimation of the parameters of a DLM and Kalman filtering including functions *dlmMLE, dlmFilter*, and *dlmForecast*.

For deep learning techniques, we use *layer_simple_rnn* function for the RNN and *layer_lstm* in Keras package. To optimize the performance of deep learning models, it was necessary to find the optimal the hyper-parameters for each model [26-28]. The optimal values of the hyper-parameters of the examined deep learning methods were determined by searching grid on a training set. For the number of input node, it is based on the number of AR terms used in the Box-Jenkins model (p = 1) [29]. Deep learning models are trained on 100 epochs with 4 sample in batch sizes based on the GPU memory. We used a two-layer of hidden layer with 10, 30, and 60 neurons for each layer. For activation function, we use the sigmoid and hyperbolic tangent functions to find the best model with root mean squared error (RMSE) as a loss function.

Results and Discussion

We predict the daily $PM_{2.5}$ concentration for 16 stations. For each station, the first 80% of the data is used as the training set, and the remaining 20% is the testing set. To evaluate the performance of the models, the RMSEs are calculated and compared.

Statistical Models Evaluation

We focus on forecasting one and seven days ahead (k = 1, 7) for all periods in the testing set. To complete the k = 1, 7 days ahead prediction for all testing periods, we shift a training data set by one day walk-forward validation. The average RMSEs for one day and seven days ahead forecasting are presented in Table 2.

Table 2. The average RMSEs of statistical models for the one day and seven ahead forecasting.

Station	Training			Testing ($k = 1$)			Testing ($k = 7$)		
	ETS	ARIMA	DLM	ETS	ARIMA	DLM	ETS	ARIMA	DLM
35t	12.171	**11.285**	12.061	2.756	**2.643**	2.728	4.328	**4.145**	4.444
36t	12.759	**11.882**	12.603	2.952	**2.917**	2.926	4.570	**4.402**	4.711
37t	10.589	**9.865**	10.511	3.671	**3.552**	3.656	5.891	**5.224**	6.094
38t	8.493	**7.943**	8.432	3.215	3.258	**3.189**	4.645	**4.039**	4.701
39t	13.316	**12.680**	13.195	2.260	**2.206**	2.256	3.477	**2.993**	3.580
40t	10.022	**9.553**	9.943	3.496	**3.394**	3.462	5.140	**4.611**	5.234
41t	7.195	**6.789**	7.179	4.653	**4.462**	4.623	7.461	**6.611**	7.323
57t	15.325	**14.290**	15.110	3.570	3.700	**3.560**	6.105	**6.011**	6.138
58t	13.534	**13.221**	13.491	**2.092**	2.162	2.122	**3.086**	3.239	3.149
67t	9.124	**8.119**	9.052	**3.001**	3.017	**3.001**	5.099	**4.825**	5.188
68t	9.517	**8.759**	9.433	3.553	**3.443**	3.563	**5.926**	5.601	6.149
69t	10.759	**10.059**	10.675	3.404	3.398	**3.396**	5.739	**5.041**	5.882
70t	11.213	**9.886**	11.110	**3.602**	3.785	3.643	7.369	**6.955**	7.580
73t	22.681	**20.744**	22.409	3.683	**3.607**	3.682	**6.030**	6.199	6.150
75t	11.522	**10.982**	11.421	2.345	2.302	**2.291**	4.009	**3.692**	3.796
76t	8.142	**7.723**	8.147	2.828	**2.814**	2.821	4.263	**4.210**	4.380

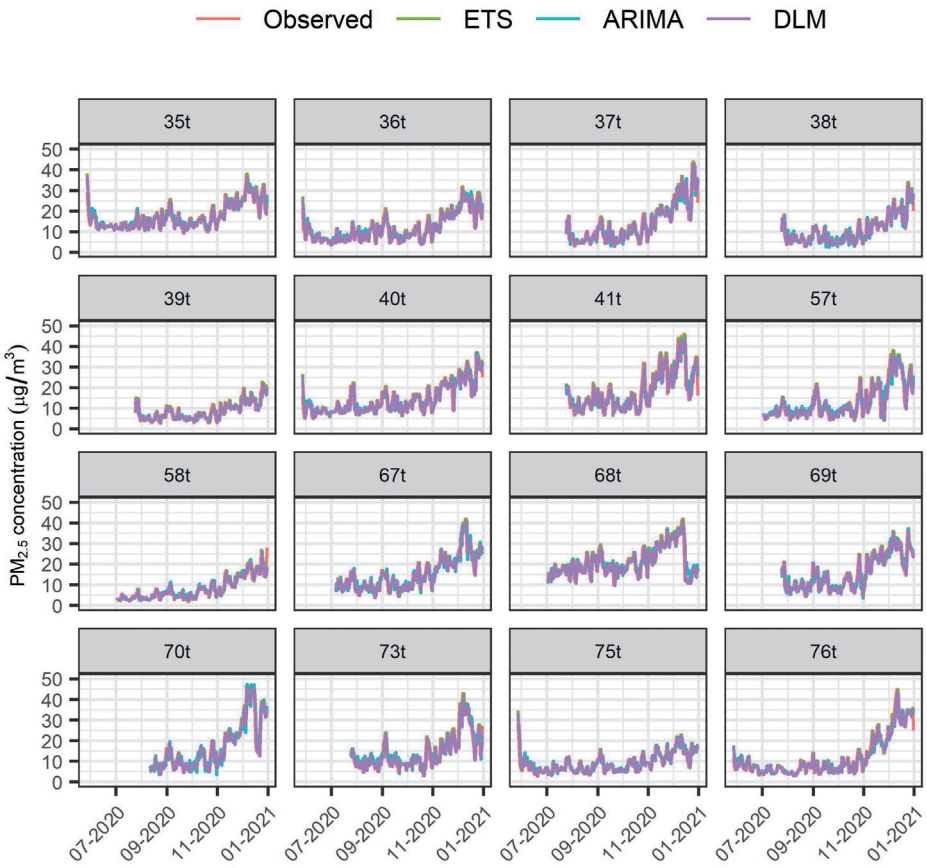

Fig. 4. One day ahead forecasting results.

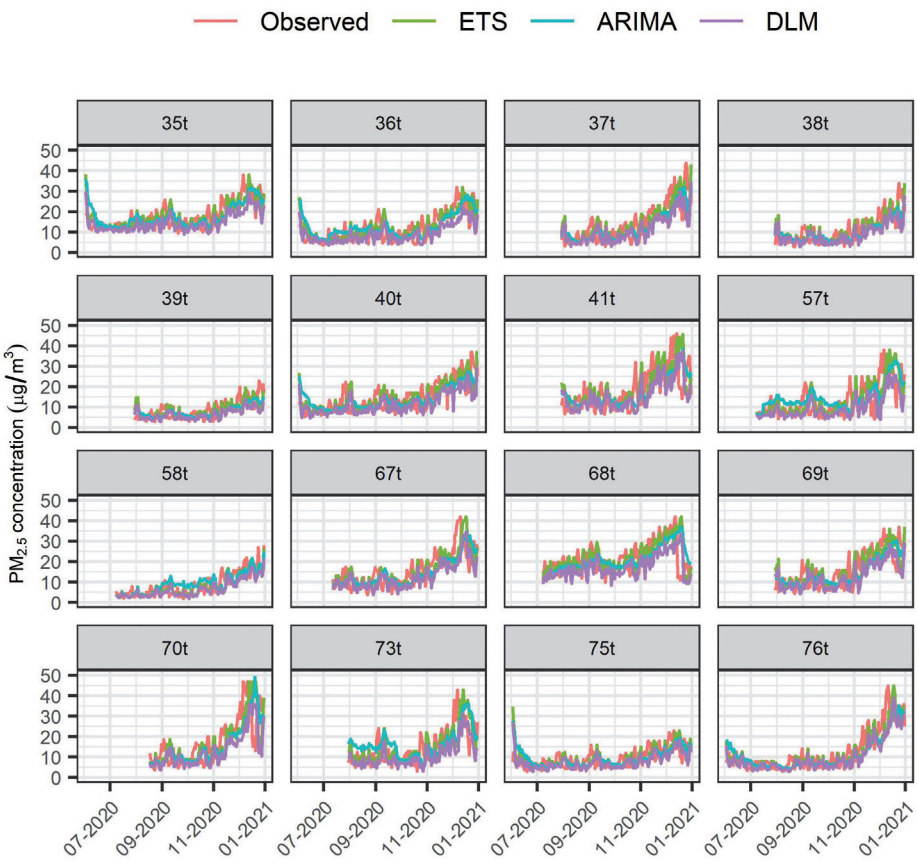

Fig. 5. Seven days ahead forecasting results.

From Table 2, the ARIMA outperforms the ETS and DLM models as it provides the smallest values of the RMSE in all stations in the training set. In the testing set, on the other hand, the DLM and ETS models perform better than ARIMA in some stations, especially when k =1. Unsurprisingly, when k = 7 all models perform worse than k = 1. The RMSEs in the training set are higher than in the testing set in all stations. One possible reason is that the training set data are in the hot season (March-May) that $PM_{2.5}$ concentrations are usually relatively high. In contrast, the data in the testing set start from June or July to December 2020, which are in the wet and cool season (June – February) with low levels of $PM_{2.5}$ concentrations. Interestingly, station 41t, with the smallest mean, standard deviation, median, and maximum $PM_{2.5}$ concentration in the dataset, as shown in Table 1, has the lowest RMSE in the training set, but the highest RMSE in the testing set. Also, the percentage change of the RMSEs from training to testing set of station 41t is the smallest compared to other stations.

To illustrate the prediction of the methods, we plot the observed and predicted of $PM_{2.5}$ concentration of the testing set for one day ahead (k = 1) and seven days ahead (k = 7) forecasting shown in Fig. 4 and 5, respectively. From Fig. 4, forecasting with k = 1, the predicted values from all methods resemble each other. But for forecasting with k = 7 in Fig. 5, the predicted values, especially those obtained from DLM, are more different from the observed ones than ARIMA and ETS. The forecasting plots for k = 1, 7 in each station show similar patterns but more errors in k = 7 forecasts.

Deep Learning Techniques Evaluation

Initially, we model the deep learning techniques; RNN and LSTM for the training set using a two-hidden layer with a sigmoid activation function

Table 3. Number of nodes in the first and the second hidden layers.

Model		Number of nodes	
		First layer	Second layer
RNN1	LSTM1	10	10
RNN2	LSTM2	10	30
RNN3	LSTM3	10	60
RNN4	LSTM4	30	10
RNN5	LSTM5	30	30
RNN6	LSTM6	30	60
RNN7	LSTM7	60	10
RNN8	LSTM8	60	30
RNN9	LSTM9	60	60

and the Adam optimizer. To find the optimal models for each technique, we structure 9 different models based on the different numbers of nodes in the first and second layer. We denote the models according to the number of nodes as shown in Table 3. Note that we use dropout equal to 0.5 to reduce the overfitting problem. The RMSEs for all models are presented in Tables 4-5. In the training set, RNN9 and LSTM9 give smallest RMSE in most stations, followed by RNN6 and LSTM6. In the testing set, for RNNs,

Table 4. The RMSE of the RNN models.

Station	Training								
	RNN1	RNN2	RNN3	RNN4	RNN5	RNN6	RNN7	RNN8	RNN9
35t	21.372	19.374	21.702	21.085	18.398	**16.434**	20.422	20.878	17.552
36t	21.551	15.743	15.831	20.751	16.837	15.915	19.979	19.424	**14.991**
37t	21.457	19.463	19.041	21.385	21.326	17.316	20.789	15.361	**14.501**
38t	17.180	15.447	15.652	17.418	12.992	13.330	16.429	**11.355**	11.976
39t	22.569	20.531	18.240	21.703	18.077	**15.123**	23.771	17.571	15.557
40t	17.679	14.048	14.056	15.519	12.537	12.662	15.906	12.803	**11.450**
41t	12.832	11.015	11.742	13.485	11.423	**9.754**	13.568	11.365	10.690
57t	27.688	26.934	23.999	26.330	22.814	21.350	26.434	24.267	**17.533**
58t	35.818	27.285	27.654	30.656	27.716	21.917	30.438	23.757	**20.518**
67t	20.490	18.844	19.038	18.584	**12.766**	13.142	19.600	17.520	14.170
68t	18.964	15.806	14.209	17.897	14.055	14.471	17.803	13.896	**13.748**
69t	21.619	19.394	19.091	20.739	**14.925**	18.961	20.308	18.381	15.188
70t	23.048	21.272	21.164	22.421	21.217	**17.622**	22.756	19.794	18.914
73t	49.394	46.286	41.397	43.775	32.432	**32.194**	44.319	38.059	38.031
75t	27.509	18.272	19.573	22.469	**13.444**	16.125	21.185	16.744	17.419
76t	17.869	15.370	13.365	16.466	13.294	**11.723**	16.856	14.606	12.824
Station	Testing								
	RNN1	RNN2	RNN3	RNN4	RNN5	RNN6	RNN7	RNN8	RNN9
35t	15.476	10.116	9.656	11.925	6.892	**3.013**	12.973	7.321	3.585
36t	18.876	10.564	6.446	14.219	8.102	7.392	13.581	10.446	**6.396**
37t	9.237	6.986	6.522	6.379	7.147	6.194	6.578	**4.962**	6.470
38t	8.370	5.656	5.020	7.392	4.251	4.354	7.272	**4.027**	5.022
39t	10.081	5.036	3.850	6.535	**2.779**	4.336	8.435	3.020	4.308
40t	13.930	9.105	7.728	9.505	6.205	4.704	10.072	4.416	**4.374**
41t	6.909	5.690	6.436	6.996	5.650	**5.254**	7.116	5.625	5.851
57t	12.029	10.093	7.613	8.752	**3.897**	4.310	8.798	4.606	4.855
58t	16.366	13.926	10.165	9.453	4.743	4.412	10.687	**3.728**	4.311
67t	10.031	8.808	6.978	6.703	**4.194**	4.710	6.839	7.319	4.226
68t	5.217	4.281	**4.193**	4.442	4.308	5.950	4.599	4.308	5.088
69t	8.249	6.222	6.348	5.428	**3.803**	6.533	5.640	3.918	5.789
70t	8.815	**6.046**	7.329	6.980	6.542	6.483	7.140	6.512	8.555
73t	15.869	13.992	8.107	11.417	**4.041**	5.415	11.902	5.875	5.762
75t	17.398	17.428	15.958	12.394	**2.659**	3.335	9.809	4.523	3.410
76t	17.105	11.296	10.034	17.018	11.423	**3.650**	13.984	9.600	5.661

Table 5. The RMSE of the LSTM models.

Station	Training								
	LSTM1	LSTM2	LSTM3	LSTM4	LSTM5	LSTM6	LSTM7	LSTM8	LSTM9
35t	25.144	17.183	16.976	21.418	16.694	15.862	22.733	18.448	**14.244**
36t	19.820	31.515	**13.446**	20.189	16.481	14.879	19.932	14.771	14.344
37t	21.763	20.883	13.309	17.931	14.846	13.936	19.673	28.415	**12.673**
38t	17.009	12.702	15.429	14.747	15.858	12.182	16.465	11.844	**11.583**
39t	22.345	20.838	15.913	21.766	16.960	15.725	23.295	18.768	**14.924**
40t	15.504	14.797	12.590	19.595	13.243	11.658	15.366	25.572	**11.794**
41t	12.668	11.962	11.393	12.805	11.535	9.670	12.753	10.876	**9.133**
57t	25.216	22.463	22.055	26.609	21.813	24.009	26.456	20.138	**17.433**
58t	33.516	22.626	25.589	30.755	24.298	21.995	32.317	23.877	**19.601**
67t	19.344	16.824	15.859	18.188	17.092	**13.392**	17.600	16.633	13.648
68t	18.002	16.948	**12.432**	17.343	14.613	13.714	18.527	15.422	15.001
69t	20.568	16.303	16.164	20.390	16.302	**14.661**	20.756	17.033	16.908
70t	23.578	20.173	20.582	22.020	20.817	**15.585**	22.592	17.956	16.492
73t	45.402	35.065	31.159	41.981	33.779	32.294	59.820	32.507	**29.361**
75t	22.875	16.679	13.043	46.094	13.251	14.588	18.457	**12.326**	15.238
76t	17.192	14.235	11.995	18.265	14.224	**11.789**	15.382	12.818	11.950
Station	Testing								
	LSTM1	LSTM2	LSTM3	LSTM4	LSTM5	LSTM6	LSTM7	LSTM8	LSTM9
35t	22.561	8.424	14.380	13.876	6.902	11.988	14.443	9.232	**4.276**
36t	14.649	35.572	4.577	14.323	8.082	6.096	15.233	8.925	**4.040**
37t	9.060	7.235	4.860	6.718	**3.627**	4.649	6.206	17.302	4.696
38t	7.370	4.988	5.203	6.347	5.506	3.532	6.504	**3.530**	3.538
39t	9.256	7.866	3.336	8.844	**2.959**	3.314	5.854	3.056	3.732
40t	10.840	12.182	4.851	17.201	5.886	**4.646**	9.970	24.510	5.081
41t	6.763	5.640	5.993	6.473	5.492	5.131	6.762	5.320	**4.856**
57t	9.185	5.629	10.135	9.818	4.612	9.360	8.845	**3.768**	4.389
58t	17.659	6.960	12.334	10.659	3.570	3.512	12.494	**2.790**	3.797
67t	7.448	7.360	4.009	6.209	6.513	**3.490**	6.431	7.216	3.809
68t	4.805	5.573	**3.789**	5.086	4.184	3.809	4.861	4.064	4.380
69t	6.032	3.789	4.055	6.321	3.921	5.178	5.227	**3.647**	4.504
70t	7.139	4.846	5.809	6.321	**4.822**	6.320	6.292	5.704	6.805
73t	12.401	12.369	7.231	9.496	5.854	10.768	29.339	**3.850**	8.217
75t	11.921	11.872	3.018	38.322	4.561	**2.803**	9.448	2.866	2.832
76t	16.604	12.356	5.473	19.864	12.785	**3.692**	13.506	7.413	6.127

RNN5 gives the smallest RMSE, followed by RNN8. Meanwhile, for LSTMs, LSTM8 shows the smallest RMSE, follows by LSTM6. Overall, the RMSEs from the LTSM are lower than the RNN in most stations.

Comparison

Several models from the deep learning techniques are applied, so we select the model providing the least RMSE from RNN and LSTM to compare with

Table 6. The comparison of the RMSE and R^2 in the testing set.

Station	Rank			RMSE (R^2)		
	1st	2nd	3rd			
35t	ARIMA	RNN6	LSTM9	2.643 *(0.811)*	3.013 *(0.752)*	4.276 *(0.500)*
36t	ARIMA	LSTM9	RNN9	2.917 *(0.815)*	4.040 *(0.643)*	6.396 *(0.105)*
37t	ARIMA	LSTM5	RNN8	3.552 *(0.857)*	3.627 *(0.852)*	4.962 *(0.723)*
38t	DLM	LSTM8	RNN8	3.189 *(0.777)*	3.530 *(0.727)*	4.027 *(0.645)*
39t	ARIMA	RNN5	LSTM5	2.206 *(0.757)*	2.779 *(0.616)*	2.959 *(0.564)*
40t	ARIMA	RNN9	LSTM6	3.394 *(0.767)*	4.374 *(0.614)*	4.646 *(0.564)*
41t	ARIMA	LSTM9	RNN6	4.462 *(0.777)*	4.856 *(0.738)*	5.254 *(0.693)*
57t	DLM	LSTM8	RNN5	3.560 *(0.818)*	3.768 *(0.796)*	3.897 *(0.782)*
58t	ETS	LSTM8	RNN8	2.092 *(0.858)*	2.790 *(0.748)*	3.728 *(0.550)*
67t	ETS/DLM	LSTM6	RNN5	3.001 *(0.868)*	3.490 *(0.822)*	4.194 *(0.742)*
68t	ARIMA	LSTM3	RNN3	3.443 *(0.757)*	3.789 *(0.706)*	4.193 *(0.640)*
69t	DLM	LSTM8	RNN5	3.396 *(0.834)*	3.647 *(0.809)*	3.803 *(0.793)*
70t	ETS	LSTM5	RNN2	3.602 *(0.905)*	4.822 *(0.830)*	6.046 *(0.734)*
73t	ARIMA	LSTM8	RNN5	3.607 *(0.813)*	3.850 *(0.788)*	4.041 *(0.766)*
75t	DLM	RNN5	LSTM6	2.291 *(0.754)*	2.659 *(0.644)*	2.803 *(0.604)*
76t	ARIMA	RNN6	LSTM6	2.814 *(0.914)*	3.650 *(0.857)*	3.692 *(0.853)*

the statistical models in the testing set. Then we rank first, second and third according to their RMSE scores. The results are shown in Table 6. The statistical models, especially ARIMA, outperform other models, followed by deep learning techniques, LSTM, and RNN. There are no promising results for the deep learning techniques to indicate which structure of the models is most outstanding. The most effective structure might incline to the data in each station. Overall, R^2 of the models in ranks 1, 2, and 3 are between 0.750-0.941, 0.614-0.857, and 0.500-0.853, respectively. These values are plausible; hence the models shown in Table 6 can adequately be used in the prediction.

Discussion

The statistical method, especially the ARIMA, shows the outstanding performance to predict the daily average of $PM_{2.5}$ concentration as it occurs in the first rank in Table 6. The DLM seems to be the second good option. The LSTM can be regarded as the better model compared with the RNN as it tends to show the smaller RMSE scores in many stations. Although, we cannot determine the best choices of model structure, the more complex structure tends to give the better prediction. While statistical models provide the best predictive values, it is important to note that they are obtained from using the most updated data to train the models. Therefore, the models are needed to be adjusted every time for different k days ahead. On the other hand, the deep learning approach only use one model for any k days ahead prediction. Although it can be computationally demanding depending on the model structure, we model once to show all predictions.

The errors from models tend to be higher when the data are in the hot season (March-May) because of high $PM_{2.5}$ concentrations. However, the models might perform differently when the data are in other seasons. For example, of all stations, station 41t, located in Nakhon Sawan, has the poorest forecasting results despite its lowest maximum concentration of only 76.042 $\mu g/m^3$, whereas other stations are more than 100 $\mu g/m^3$. One possible reason is that the data in this station have no clear seasonal pattern, as shown in Fig. 2. Nakhon Sawan is the lower north province where the topography is rather different from other stations which are mostly located in the upper north provinces.

Our results agreed with the works of Liu et al. [12] and Syafei et al. [13] that suggested that ARIMA provided the best results, especially for short-term forecasting of air pollution data. Nonetheless, we could indicate that the statistical methods are superior to the machine learning techniques, as the first ranks from all stations are completely from statistical methods. Furthermore, as the structure of the deep learning techniques plays a vital role in model performance, more possible structures could be investigated to improve the predictive results. Unfortunately, this is leading to more computationally expensive modeling.

Conclusions

In summary, to predict the daily average of $PM_{2.5}$ concentration for the northern region of Thailand, the statistical method, particularly ARIMA, is highly recommended. It should be noted that the data used in the models are needed to be updated to get high accuracy. Nonetheless, the deep learning techniques, especially LSTM is also considered as a good method. Deep learning methods tend to be more computationally expensive than the statistical methods, but they can be more practical for long term prediction.

Acknowledgments

This research is supported by the Faculty of Science, Chiang Mai University. We would like to thank Pollution Control Department, Air Quality and Noise Management Bureau, The Ministry of Natural Resources and Environment of Thailand for valuable data.

Conflict of Interest

The authors declare no conflict of interest.

References

1. HARISHKUMAR K.S., YOGESH K.M., GAD I. Forecasting air pollution particulate matter ($PM_{2.5}$) using machine learning regression models. Procedia Computer Science, **171**, 2057, **2020**.
2. FOSZCZ D., NIEDOBA T., SIEWIOR J. Models of air pollution propagation in the selected region of Katowice. Atmosphere, **12** (6), 695, **2021**.
3. LEI T.M., SIU S.W., MONJARDINO J., MENDES L., FERREIRA F. Using Machine Learning Methods to Forecast Air Quality: A Case Study in Macao. Atmosphere, **13** (9), 1412, **2022**.
4. ELSARAITI M., MERABET A., AL-DURRA A. Time Series Analysis and Forecasting of Wind Speed Data. 2019 IEEE Industry Applications Society Annual Meeting, IEEE, 1-5, **2019**.
5. GEETHA A., NASIRA G.M. Time-series modelling and forecasting: Modelling of rainfall prediction using ARIMA model. International Journal of Society Systems Science, **8** (4), 361, **2016**.
6. MURAT M., MALINOWSKA I., GOS M., KRZYSZCZAK J. Forecasting daily meteorological time series using ARIMA and regression models. International Agrophysics, **32** (2), 253, **2018**.
7. RAY S., DAS S.S., MISHRA P., AL KHATIB A.M. G. Time series SARIMA modelling and forecasting of monthly rainfall and temperature in the South Asian countries. Earth Systems and Environment, **5**, 531, **2021**.
8. SWAIN S., NANDI S., PATEL P. Development of an ARIMA model for monthly rainfall forecasting over Khordha district, Odisha, India. In Recent Findings in Intelligent Computing Techniques, Springer, 325, **2018**.
9. NOBRE F.F., MONTEIRO A.B.S., TELLES P.R., WILLIAMSON G.D. Dynamic linear model and SARIMA: a comparison of their forecasting performance in epidemiology. Statistics in Medicine, **20** (2), 3051, **2001**.
10. OMEKARA C.O., OKEREKE O.E., EHIGHIBE S.E. Time series analysis of interest rate in Nigeria: A comparison of ARIMA and state space models. International Journal of Probability and Statistics, **5** (2), 33, **2016**.
11. KUMAR U., JAIN V.K. ARIMA forecasting of ambient air pollutants (O_3, NO, NO_2 and CO). Stochastic Environmental Research and Risk Assessment, **24**, 751, **2010**.
12. LIU T., LAU A.K., SANDBRINK K., FUNG J.C. Time series forecasting of air quality based on regional numerical modeling in Hong Kong. Journal of Geophysical Research Atmospheres, **123** (3), 4175, **2018**.
13. SYAFEI A.D., RAMADHAN N., HERMANA J., SLAMET A., BOEDISANTOSO R., ASSOMADI A.F. Application of Exponential Smoothing Holt Winter and ARIMA Models for Predicting Air Pollutant Concentrations. EnvironmentAsia, **11** (3), 251, **2018**.
14. CEKIM H. O. Forecasting PM_{10} concentrations using time series models: a case of the most polluted cities in Turkey. Environmental Science and Pollution Research, **27** (20), 25612, **2020**.
15. CHOI J.E., LEE H., SONG J. Forecasting daily PM_{10} concentrations in Seoul using various data mining techniques. Communications for Statistical Applications and Methods, **25** (2), 199, **2018**.
16. LI Y., ZHU Z., KONG D., HAN H., ZHAO Y. EA-LSTM: Evolutionary attention-based LSTM for time series prediction. Knowledge-Based Systems, **181**, 104785, **2019**.
17. QADEER K., REHMAN W. U., SHERI A.M., PARK I., KIM H.K., JEON M. A long short-term memory (LSTM) network for hourly estimation of $PM_{2.5}$ concentration in two cities of South Korea. Applied Sciences, **10** (11), 3984, **2020**.
18. SELVIN S., VINAYAKUMAR R., GOPALAKRISHNAN E.A., MENON V.K., SOMAN K.P. Stock price prediction using LSTM, RNN and CNN-sliding window model. 2017 international conference on advances in computing, communications and informatics (ICACCI), IEEE, 1643, **2017**.
19. CHEN Y. Prediction algorithm of $PM_{2.5}$ mass concentration based on adaptive BP neural network. Computing, **100** (8), 825, **2018**.
20. LU G., YU E., WANG Y., LI H., CHENG D., Huang L., LIU Z., MANOMAIPHIBOON K., LI L. A novel hybrid machine learning method (OR-ELM-AR) used in forecast of $PM_{2.5}$ concentrations and its forecast performance evaluation. Atmosphere, **12** (1), 78, **2021**.
21. GUPTA P., ZHAN S., MISHRA V., AEKAKKARARUNGROJ A., MARKERT A., PAIBONG S., CHISHTIE F. Machine Learning Algorithm for Estimating Surface $PM_{2.5}$ in Thailand. Aerosol and Air Quality Research, **21**, 210105, **2021**.
22. CECAJ A., LIPPI M., MAMEI M., ZAMBONELLI F. Comparing deep learning and statistical methods in forecasting crowd distribution from aggregated mobile phone data. Applied Sciences, 10 (18), 6580, **2020**.
23. RAJULA H.S.R., VERLATO G., MANCHIA M., ANTONUCCI N., FANOS V. Comparison of conventional statistical methods with machine learning in medicine: diagnosis, drug development, and treatment. Medicina, **56** (9), 455, **2020**.

24. SPILIOTIS E., MAKRIDAKIS S., SEMENOGLOU A. A., ASSIMAKOPOULOS V. Comparison of statistical and machine learning methods for daily SKU demand forecasting. Operational Research - An International Journal, 1, **2020**.
25. VENTURA L.M.B., DE OLIVEIRA PINTO F., SOARES L.M., LUNA A.S., GIODA A. Forecast of daily $PM_{2.5}$ concentrations applying artificial neural networks and Holt-Winters models. Air Quality, Atmosphere & Health, **12**, 317, **2019**.
26. BERGSTRA J., BENGIO Y. Random search for hyper-parameter optimization. Journal of Machine Learning Research, **13** (10), 281, **2012**.
27. HUTTER F., HOOS H.H., LEYTON-BROWN K. Sequential model-based optimization for general algorithm configuration. In International conference on learning and intelligent optimization, Springer, 507, **2011**.
28. ZHANG G., PATUWO B.E., HU M.Y. Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting, **14** (1), 35, **1998**.
29. TANG Z., FISHWICK P.A. Feedforward neural nets as models for time series forecasting. Informs Journal on Computing, **5** (4), 374, **1993**.