



Long-term time-series pollution forecast using statistical and deep learning methods

Pritthijit Nath¹ · Pratik Saha² · Asif Iqbal Middy¹ · Sarbani Roy¹ 

Received: 19 August 2020 / Accepted: 3 March 2021 / Published online: 3 April 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract

Tackling air pollution has become of utmost importance since the last few decades. Different statistical as well as deep learning methods have been proposed till now, but seldom those have been used to forecast future long-term pollution trends. Forecasting long-term pollution trends into the future is highly important for government bodies around the globe as they help in the framing of efficient environmental policies. This paper presents a comparative study of various statistical and deep learning methods to forecast long-term pollution trends for the two most important categories of particulate matter (PM) which are PM_{2.5} and PM₁₀. The study is based on Kolkata, a major city on the eastern side of India. The historical pollution data collected from government set-up monitoring stations in Kolkata are used to analyse the underlying patterns with the help of various time-series analysis techniques, which is then used to produce a forecast for the next two years using different statistical and deep learning methods. The findings reflect that statistical methods such as auto-regressive (AR), seasonal auto-regressive integrated moving average (SARIMA) and Holt–Winters outperform deep learning methods such as stacked, bi-directional, auto-encoder and convolution long short-term memory networks based on the limited data available.

Keywords Time-series analysis · Air pollution · Statistical models · Deep learning · Long-term forecast

1 Introduction

The problem of urban air pollution has become more and more serious due to rapid industrialization in recent times, thus badly affecting not only our physical health but also the environment around us. Research on pollution forecasting has thus become a key issue in environmental protection to better evaluate the necessary steps to be taken

to curb its long-term effects. Major cities around the world have set up various automatic air quality monitoring stations that detect the levels of particulate matter (PM) such as PM_{2.5} [11] and PM₁₀ [11], in specific areas spread throughout the city.

Air quality forecasting methods proposed till now can be broadly classified into two main categories, namely statistical methods and deep learning methods. The performance of each method depends on multiple factors such as trend, seasonality and noise in the data as well as meteorological and socio-economic trends [12], which also equally play an important role in contributing to pollution in a specific region.

Ong et al. [13] proposed a two-stage approach where a recurrent neural network (RNN) is pre-trained on hourly data using an auto-encoder-based model, followed by fine-tuning to filter out sensor data. The resulting network was then used to predict PM_{2.5} concentrations. Bashir Shaban et al. [14] studied the forecasting of support vector machines (SVM), artificial neural networks (ANN) and model trees (M5P) using both univariate and multivariate

✉ Sarbani Roy
sarbani.roy@jadavpuruniversity.in

Pritthijit Nath
prithijit.nath@ieee.org

Pratik Saha
pratiksaha198@gmail.com

Asif Iqbal Middy
asifim.rs@jadavpuruniversity.in

¹ Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

² Department of Computer Science, SRM University, Kattankulathur, Chennai, India

modelling to predict hourly forecasts. Tao et al. [15] worked on air pollution forecasting using one-dimensional convolution neural networks (CNN) and bi-directional gated recurrent networks (GRU) based on Beijing PM2.5 dataset [16] which consists of hourly data extracted from different air quality monitoring stations in Beijing.

Mlakar et al. [17] explored the important task of feature selection in a model and put forward several algorithms for feature reduction, all of them based on the case of forecasting SO₂ half an hour in advance. Li et al. [18] proposed a multivariate CNN-LSTM model. The authors performed a thorough comparison with other hybrid long short-term memory (LSTM) models based on their root mean squared error (RMSE) and mean average error (MAE) along with their training times, and in the end, they proposed the hybrid CNN-LSTM model to be more effective than others. Wang et al. [19] explored the use of the seasonal auto-regressive integrated moving average (SARIMA) model, along with studies on the periodicity of monthly PM2.5 data as well as procedures for parameter estimation, diagnostic checking, to predict and forecast the air pollutants in an effective way. Other related studies in this area can be found summarized in Table 1.

All of these works are crucial and have been found to be extremely effective in carrying out short-term predictions

of pollution levels in a city. However, these works do not account the usage of those methods in prediction horizons which span for more than a year. To tackle the problem of long-term forecasting, a different approach had to be adopted, which involves the usage of monthly data for predicting long-term trends. Daily data from monitoring stations are resampled into monthly data in our study as it has been observed that long-term yearly forecasts performed on the daily data converged to the statistical mean, thus making the results produced ineffective.

This paper presents a comparative study of long-term pollution forecasts using the best four statistical methods such as auto-regressive (AR), seasonal auto-regressive integrated moving average (SARIMA), Holt–Winters and Prophet along with four best deep learning methods such as stacked, bi-directional, auto-encoder and convolution LSTMs. The study is based on the historical pollution data that are extracted from various government set-up monitoring station(s) of the city Kolkata (India). Here, the overall end-to-end approach for long-term forecasting of pollution level can be viewed as a combination of three main stages, namely data pre-processing, time-series analysis (based upon the pre-processed historical data) and data modelling (using various statistical and deep learning models to predict PM2.5 and PM10 values in future).

Table 1 Summary of forecasting models proposed by researchers in recent decades

Author	Year	Method	Description
Mahajan et al. [1]	(2017)	Neural network auto regression (NNAR)	Hourly forecast of PM2.5 was performed and its prediction was compared with ARIMA and Holt–Winters models
Xiang [2]	(2019)	Multiple kernel learning (MKL) framework	MKL was proposed to forecast the near future PM2.5 values and was compared to single kernel-based support vector regression (SVR) model
Xie [3]	(2017)	Deep neural network	The proposed model was based on manifold learning along with a deep belief network (DBN) developed to learn the features of the input candidates for local PM2.5 forecast
Luo et al. [4]	(2018)	Adaptive iterative forecast (AIF) Model	The proposed AIF model could predict the value of PM2.5 for the next few hours (by linear programming, normalization and time series) based on the trend of historical data
Feng et al. [5]	(2015)	Hybrid artificial neural network (ANN)	A hybrid model combining air mass trajectory analysis and wavelet transformation was proposed to improve the forecast's accuracy
Haiming and Xiaoxiao [6]	(2013)	RBF neural network	Along with PM2.5, other influence factors were chosen to predict its concentration and then compared with the classic BP network model
Yan et al. [7]	(2018)	Encoder–decoder model	Three prediction models: BP, stack GRU and encoder–decoder were constructed to predict the PM2.5 concentration of every hour of the next day
Maria et al. [8]	(2015)	Multilayer perceptron neural network and clustering algorithm	In addition to multilayer neural network, clustering algorithm was used to find relationships between PM10 and meteorological variables for increasing accuracy of forecasting
Al-Kassabeh et al. [9]	(2013)	Nonparametric artificial neural network (ANN)	For prediction of PM10, other meteorological parameters were also considered and an artificial neural network based auto regressive with external input (ANNARX) model was proposed to provide high calibre modelling
Lam and Mok [10]	(2007)	ANN applied three-layer feed-forward network (TLFN)	Along with six input parameters for each seasonal model, highest absolute values of correlation coefficients were selected to form the model input pattern to feed into the ANN for 24 hour predictions

Unlike previous studies [20–22], this study aims at finding the optimal combination of techniques for pre-processing, time-series analysis and finally forecasting, so that statutory bodies focussing on producing similar projections for their city can take advantage of the proposed approach to construct their forecasting infrastructure. After pre-processing (e.g. missing value imputation), an in-depth time-series analysis of both PM_{2.5} and PM₁₀ is conducted to find the major trends in the data. The results obtained thus are used to ascertain the hyper-parameters of the predictive models, which is further tuned using a popular hyper-parameter finding process called Grid-Search. Different performance metrics (namely RMSE, MAE) are utilized to analyse the performance of various models. The two-year forecast produced by different predictive models is then studied in detail, and domain-specific discussions are presented based on the projections made.

The main aim/objectives of this comparative study are:

- To evaluate a set of methods and find the optimal ones for all the stages ranging from data pre-processing to data modelling, in performing long-term forecasting of PM_{2.5} and PM₁₀ time-series data.
- To perform the imputation of missing values in the raw data by using different imputation techniques like multivariate imputation and mean before after [23].
- To conduct a comprehensive time-series analysis of both PM_{2.5} and PM₁₀ for analysing underlying trends.
- To carry out the evaluation and forecasts of various models using walk forward approach (WFA) allowing the results to be more accurate and close to real-world scenarios.

The rest of the paper is organized as follows: In Sect. 2, a brief background of the city and the pollutants that are a part of the study are presented along with a description and summary statistics of the data obtained from multiple sources. Section 3 consists of the detailed description of the techniques used in missing value imputation, time-series analysis and the models used in the study. Section 4 is about the approaches used in data preparation, time-series analysis, model training, evaluation and future forecasts. The results are laid out in great detail in Sect. 5, along with a detailed discussion on the quality of forecasts and efficiency of the models with regard to learning the underlying trends. The conclusions drawn from the findings are ultimately presented in Sect. 6.

2 Data description

Located on the eastern side of India, Kolkata is the capital city of the state of West Bengal. As per the 2011 Census, around 14 million people reside in the city making it one of

the major cities in the world. Due to high socio-economic activity, the air quality of Kolkata is sub-par due to the presence of significantly higher levels of particulate matter and toxic gases in the city atmosphere. Besides the usual contribution of industries, transportation is also one of the major air-polluting sectors due to ineffective control measures and high abundance of poorly maintained vehicles plying in the city [24]. This section deals mainly with the major pollutants and the statistical description of pollution data used in the study.

2.1 Pollutants

2.1.1 PM_{2.5}

Particulate matter (PM) is a mixture of coarse, fine and ultra-fine solid and liquid particles suspended in the air. PM_{2.5} refers to that particulate matter which has a diameter less than 2.5 μm, as a result, they remain suspended in the air for longer periods. These are mostly produced from burning fuels, forest fires, volcanic eruptions, etc. Exposure to PM_{2.5} can lead to multiple short-term and long-term health issues. Prolonged exposure may result in permanent respiratory problems such as asthma, chronic bronchitis and heart disease.

2.1.2 PM₁₀

PM₁₀ are those solid and liquid particles that have a diameter greater than 2.5 μm and less than 10 μm; hence, they persist in the air for lesser time compared to PM_{2.5}. These are particles that consist of smoke, dust from industries, roads and other places. Soil and rocks when crushed, create such particles that get blown away by the wind. Being heavier than PM_{2.5}, they cannot go deep enough into the lungs, hence are less risky than PM_{2.5}; however, they are responsible for lung injury and can cause ailments like chronic obstructive pulmonary disease (COPD) [25].

2.2 Pollution data

The pollution data of Kolkata were provided by the central pollution control board (CPCB) [26], responsible for providing field information regarding pollution of various places throughout the country. In this paper, the pollution data which form the basis of the study were extracted from the station positioned at Victoria Memorial Hall (22.5448° N, 88.3426° E), supplemented with data procured from other nearby stations.

Preliminary analysis of the data obtained found it to be daily in nature, spanning four years from 10th January 2016 to 18th February 2020. The air quality monitoring station at Victoria Memorial Hall supplied values for temperature, relative humidity, PM2.5 and PM10. Of the values supplied, large chunks of the raw data extracted were found to be missing due to external factors such as hardware failure, maintenance operations, etc. Hence, these values needed to be either found out from other external sources or have to be internally imputed using various techniques.

Temperature values originally absent were extracted from the University of Dayton's Temperature [27] archive. Relative humidity values were web scraped from Weather Underground [28], followed by further re-sampling to get the daily values needed. Missing daily PM2.5 values were extracted from the US Department of State's AirNow [29] web portal.

2.2.1 Descriptive statistics

The descriptive statistics for PM2.5, PM10, temperature and relative humidity can be seen in Table 2. It shows that during winter, the PM2.5 levels rise significantly. Low wind speeds present along with lower temperatures create conditions for temperature inversion. On the other hand, during summer and monsoon, comparatively lower levels of pollution are observed. This can be attributed to the increased circulation of air in the troposphere as well the squalls from the north-west direction which the city experiences during the months just before the onset of monsoon.

Table 2 Descriptive statistics for PM2.5, PM10, temperature and relative humidity

Month	PM2.5 ($\mu\text{g}/\text{m}^3$)				PM10 ($\mu\text{g}/\text{m}^3$)				Temperature ($^{\circ}\text{C}$)				Relative humidity (%)			
	μ	σ	min	max	μ	σ	min	max	μ	σ	min	max	μ	σ	min	max
Jan	163.35	69.28	46.38	508.0	194.18	75.04	75.32	451.42	18.34	1.93	11.61	23.28	70.69	7.61	49.77	95.51
Feb	111.49	44.40	18.33	281.0	159.88	66.16	27.58	303.09	23.05	2.82	17.28	30.17	65.22	9.69	45.00	95.15
Mar	67.18	27.42	2.71	159.0	82.02	32.60	29.15	193.57	27.44	2.25	20.40	31.22	65.01	10.13	43.10	88.90
Apr	38.09	13.47	3.04	74.0	56.62	20.27	20.18	137.81	30.02	2.05	25.62	34.89	69.33	7.78	44.20	81.70
May	37.08	14.53	0.72	114.0	55.88	19.97	2.29	120.07	30.34	1.61	25.35	33.28	73.01	6.06	59.10	94.20
Jun	33.58	19.25	0.30	172.0	53.25	43.81	0.59	298.22	30.08	1.59	25.32	34.18	78.82	6.20	61.89	95.89
Jul	29.61	15.44	2.00	112.0	42.00	39.63	8.97	288.13	28.97	1.18	26.25	31.51	85.07	5.89	71.80	97.57
Aug	28.75	14.07	0.04	72.0	37.50	16.13	6.74	85.66	28.74	1.29	25.07	31.61	85.19	9.04	14.72	98.88
Sep	30.77	18.10	3.29	113.0	44.30	26.95	5.22	111.23	28.79	1.42	25.17	31.94	84.08	5.68	70.80	97.27
Oct	62.83	38.37	8.91	257.0	92.93	51.59	13.01	204.75	27.37	1.89	22.57	31.83	79.52	8.19	63.30	97.77
Nov	120.50	65.64	12.38	308.0	165.74	68.57	17.34	354.31	23.73	1.94	19.00	28.61	72.46	8.65	54.60	97.57
Dec	152.33	67.50	26.00	402.0	193.37	66.46	74.53	365.14	19.47	2.26	14.67	24.06	72.59	7.09	49.84	90.50

μ , σ , min and max represent the mean, standard deviation, minimum and maximum, respectively

3 Methods

As the study concerns with a general approach for finding an optimal combination of techniques for pre-processing, missing value imputation and finally forecasting, only the best methods (as shown in Fig. 1) in statistical and deep learning-based modelling are studied in depth. The time-series analysis methods mentioned in Fig. 1 are specifically curated to help in investigating the underlying patterns and trends present in the data. In this section, all those techniques are presented in detail along with a brief discussion of the theory behind them.

3.1 Missing value imputation

Here, two widely used missing value imputation methods, namely mean before after and multivariate imputation, are discussed.

3.1.1 Mean before after

The mean before after method replaces the missing value at time i by the mean of the value at one time instant $i + 1$ in the future and the value at one time instant $i - 1$ in the past.

$$\bar{x}_i = \frac{x_{i-1} + x_{i+1}}{2} \quad (1)$$

Norazian et al. [23] showed that mean before after method of imputation gave the least error when compared to other univariate imputation methods. However, it must be noted that the mean before after method works best only when there are non-null values present in the window being considered. If there are a high amount of null values

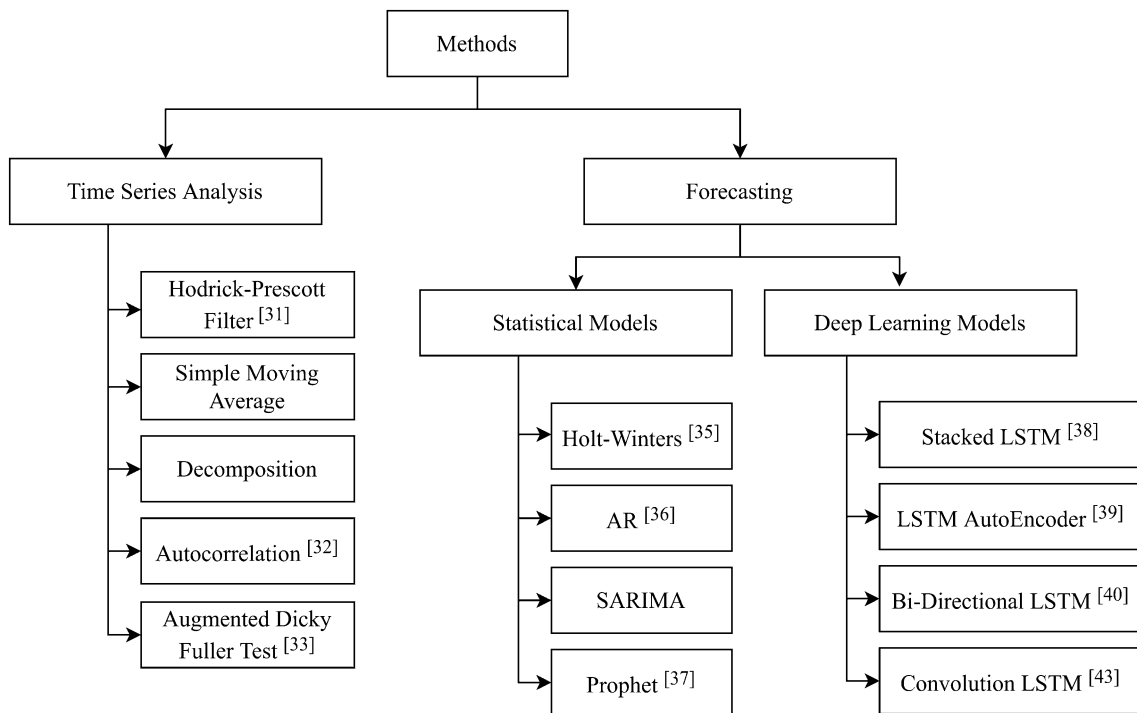


Fig. 1 Taxonomy of methods used in this study for time-series analysis and in statistical and deep learning-based modelling

present, this technique may not give satisfactory results. In such cases, other imputation techniques must be considered.

3.1.2 Multivariate imputation

The method of multivariate imputation on electronic computer devices was proposed by Buck [30]. If m out of n rows have the complete set of k observations for all the features, we can consider a matrix X containing all the n rows, having those m rows at the very first. From the matrix X , we can obtain k equations of the form

$$E(x_{rj}) = f_j(x_{r1}, x_{r2}, \dots, x_{rj-1}, x_{rj+1}, \dots, x_{rk}) \tag{2}$$

where f_j resembles the fitted regression function and x_{rj} ($r = 1, 2, 3, \dots, m$) implies the expected value by forming a multiple regression of j on the other $k - 1$ variables for the r th row. By replacing the value of r with the row i , we can estimate the value of the missing variable.

$$E(x_{ij}) = f_j(x_{i1}, x_{i2}, \dots, x_{ij-1}, x_{ij+1}, \dots, x_{ik}) \tag{3}$$

By extending this idea for univariate imputation, multivariate imputation can be performed by calculating the multiple regression formula for each missing variate on $k - v$ other variates. For any combination of v variates missing,

$$k \binom{k-1}{v-1} \tag{4}$$

equations have to be calculated. The missing value can be estimated by selecting the proper equation and solving it.

3.2 Time-series analysis

In this subsection, different methods for time-series analysis are discussed.

3.2.1 Hodrick–Prescott filter

The Hodrick–Prescott [31] (HP) filter is a mathematical tool used to remove the cyclical component of a time series from raw data. It is used to obtain a smoothed-curve representation of a time series, from which the long-term trend can be better observed compared to short-term variations.

The sensitivity of the trend to short-term variations can be adjusted by modifying a multiplier λ . The greater the value of λ , the closer the trend path will be a straight line.

Given a time series $y_t = \tau_t + c_t + \epsilon_t$ where τ_t is the trend component, c_t is the cyclical component and ϵ_t is the error component, for an adequately chosen λ , there exists a trend component which solves

$$\min_{\tau} \left(\sum_{t=1}^T (y_t - \tau_t)^2 + \lambda \sum_{t=2}^{T-1} [(\tau_{t+1} - \tau_t) - (\tau_t - \tau_{t-1})]^2 \right) \tag{5}$$

3.2.2 Simple moving average

In time-series analysis, simple moving average (SMA) is an arithmetic method which involves finding out the unweighted mean of the last n periods of data. Upon calculation of successive values, the oldest sum can be left out and the resulting new value can be calculated using the following equation:

$$\bar{p}_{SM} = \bar{p}_{SM,prev} + \frac{1}{n}(p_M - p_{M-n}) \tag{6}$$

where \bar{p}_{SM} denotes the simple moving average at that instant while p_M denotes the mean over the previous n periods of time.

3.2.3 Decomposition

Decomposition is the statistical approach of breaking down time series into its trend, seasonal, cyclical and the irregular components. A time series following an additive model can be thought of:

$$y_t = \tau_t + c_t + s_t + \epsilon_t \tag{7}$$

, whereas a multiplicative model would be expressed in the following way:

$$y_t = \tau_t \times c_t \times s_t \times \epsilon_t \tag{8}$$

where τ_t , c_t , s_t and ϵ_t are the trend, cyclical, seasonal and irregular (noise) components, respectively. To find the trend component τ_t of a time series with frequency f , a convolution filter with a linear kernel containing of elements equal to $1/f$ is applied. By removing the τ_t component, the seasonal component is found out by averaging over smoothed series for each period of the component left out.

3.2.4 Autocorrelation

The autocorrelation function proposed by Box and Jenkins [32] can be used to detect non-randomness in the data and also to identify parameters of appropriate time-series models.

Given measurements, Y_1, Y_2, \dots, Y_N at time $i = 1, 2, \dots, N$, respectively, the lag k autocorrelation function is defined as

$$r_k = \frac{\sum_{i=1}^{N-k} (Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^N (Y_i - \bar{Y})^2} \tag{9}$$

3.2.5 Augmented Dicky Fuller test

An augmented Dicky Fuller (ADF) test [33] uses the null hypothesis that a unit root is present in a time-series sample. The Dicky Fuller test is used if a time-series sample is a random walk or not.

$$\Delta y_t = y_t - y_{t-1} = \alpha + \beta t + \gamma y_{t-1} + e_t \tag{10}$$

Stationarity [34] refers to the time-series data being devoid of any trend or seasonal effects, thereby making the data easier to model as the summary statistics such as mean and variance tend to stay the same with respect to time. If $\gamma = 0$ then we have a random walk process, if not, then the data are a stationary process. The augmented Dicky Fuller test is an extension of the Dicky Fuller test, allowing for higher-order regressive processes of the form Δy_{t-p} where $1 \leq p < t$.

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \delta_2 \Delta y_{t-2} + \dots \tag{11}$$

The null hypothesis is that the data are non-stationary. We intend to reject the null hypothesis for this test, so we want a p value < 0.05 .

3.3 Statistical models

In this subsection, different statistical models for time-series forecasting are discussed.

3.3.1 Holt–Winters

The Holt–Winters [35] method comprises four equations, namely the forecast equation and three smoothing equations. The additive component form of the method is shown in Eqs. (12)–(15):

$$\hat{y}_{t+h|t} = \ell_t + hb_t + s_{t+h-m(k+1)} \tag{12}$$

$$\ell_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(\ell_{t-1} + b_{t-1}) \tag{13}$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1} \tag{14}$$

$$s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m} \tag{15}$$

where ℓ_t , b_t and s_t stand for level, trend and seasonal components, respectively, along with the corresponding smoothing factors α , β^* and γ . The seasonality is denoted by m , while k is the integer part of the fraction $(h - 1)/m$, which ensures that the estimates of the seasonal indices used for forecasting come from the last part of the sample.

The multiplicative component form of Holt–Winters is:

$$\hat{y}_{t+h|t} = (\ell_t + hb_t)s_{t+h-m(k+1)} \tag{16}$$

$$\ell_t = \alpha \frac{y_t}{s_{t-m}} + (1 - \alpha)(\ell_{t-1} + b_{t-1}) \tag{17}$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1} \tag{18}$$

$$s_t = \gamma \frac{y_t}{(\ell_{t-1} + b_{t-1})} + (1 - \gamma)s_{t-m} \tag{19}$$

Additive methods are used when the magnitude of the seasonal fluctuations does not vary with the level of the time series. On the other hand, multiplicative methods are used when there is variation in the seasonality which appears to be proportional to the level of the time series.

3.3.2 Auto-regressive (AR)

An auto-regressive model is a model which upon taking input of the previous observations predicts the values at the next time step. The model can be described in the form:

$$X_t = c + \sum_{i=1}^p \phi_i X_{t-i} + \epsilon_t \tag{20}$$

where $\phi_1, \phi_2, \dots, \phi_p$ are the parameters of the model, c is the constant term and ϵ_i is the noise term. p is referred to as the order of the model denoted by $AR(p)$. The coefficients of the AR model can be solved by ordinary least-squares (OLS) method or by using Yule–Walker [36] equations.

3.3.3 Seasonal auto-regressive integrated moving average (SARIMA)

An ARIMA model [32] consists of an auto-regressive (AR), integrated (I) and a moving average (MA) component to better understand a time-series data or to predict future time-series data. The AR component indicates that the evolving variable of interest is regressed on its own lagged values. The I component indicates that the values have been replaced by the present values and their previous values. The MA component indicates that the regression error is a linear combination of error terms that occurred in the past. The ARIMA model can be formulated as shown in Eq. 21.

$$\left(1 - \sum_{i=1}^p \phi_i L^i\right) (1 - L)^d X_t = \delta + \left(1 + \sum_{i=1}^q \theta_i L^i\right) \epsilon_t \tag{21}$$

where p , d and q denote the time lags of the AR component, the degree of differencing and the order of the MA component, respectively.

The seasonal ARIMA model is an extension of the ARIMA Model, with additional seasonal AR, I and MA terms as well as a periodic term denoted by m .

3.3.4 Prophet

Prophet [37] is a forecasting procedure developed recently by Facebook. The model focuses on providing fast and accurate forecasts that can be later tuned manually. It is based on an additive model where nonlinear trends are fit with yearly, weekly and daily seasonality, along with holiday effects. It works best with time series where seasonal effects are profound and the historical data spans several seasons.

3.4 Deep learning models

Here, different deep learning-based forecasting methods, namely stacked LSTM, LSTM auto-encoder, bi-directional LSTM and convolution LSTM are presented.

3.4.1 Stacked LSTM

Long short-term memory (LSTM) [38] networks are a special kind of recurrent neural networks (RNN) designed to be used to remember information for longer periods. They are explicitly designed to counter the vanishing and exploding gradient problem, unlike RNNs which are very much affected by it. LSTMs have four interacting layers in their repeating module compared to one in RNNs.

The layers of an LSTM network can be mathematically expressed as shown in Eqs. (22)–(27):

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{22}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{23}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{24}$$

$$\tilde{c}_t = \sigma_h(W_c x_t + U_c h_{t-1} + b_c) \tag{25}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \tag{26}$$

$$h_t = o_t \circ \sigma_h(c_t) \tag{27}$$

where f_t , i_t , o_t and \tilde{c}_t are the activation vectors of forget gate, input gate, output gate and cell input gate, respectively. c_t and h_t are the cell state and hidden state vectors. x_t is the input state vector. Matrices of the form W_q and U_q , respectively, contain the weights of the input and recurrent connections. In activation functions, σ_h denotes the hyperbolic tangent function, while σ_c denotes the sigmoid function.

Stacked LSTM is an extension of a vanilla LSTM network in which the LSTM layers are stacked on top of each other. This helps to increase model complexity. If the input is already the result from an LSTM layer then the current LSTM layer can create a more complex feature representation of the current input.

3.4.2 LSTM auto-encoder

An auto-encoder [39] is a type of artificial neural network which is used to learn the features of input data in an unsupervised manner. An auto-encoder aims to learn the encoding for a set of data, by training the model to ignore noise. After the reduction is completed, reconstruction is undertaken in which the model learns to generate an output as close as possible to the original input from the encoding done by the reduction side.

It consists of an encoder and decoder which can be expressed mathematically as shown in Eqs. (28)–(30):

$$\phi : X \rightarrow F \quad (28)$$

$$\psi : F \rightarrow X \quad (29)$$

$$\phi, \psi = \arg \min_{\phi, \psi} \|X - (\psi \circ \phi)X\|^2 \quad (30)$$

where ϕ and ψ denote the encoder and decoder components, respectively. X is the input, and F denotes the feature space generated by the mapping.

LSTM auto-encoder is a type of neural network in which an LSTM architecture is used in the encoder and decoder components to work on data arranged in sequences.

3.4.3 Bi-directional LSTM

Bi-directional LSTM is an extension of the Vanilla LSTM network and is the LSTM implementation of bi-directional recurrent neural networks [40] in which the two hidden layers of opposite directions are connected to the same output. Due to the added connection, the output layer can benefit from both the past (backward) and future (forward) states simultaneously.

In a bi-directional layer, the neurons are split into the positive and negative direction which corresponds to the forward and backward states, respectively. However, it must be noted that the output of the two states is not connected to the input of the opposite direction's state.

3.4.4 Convolution LSTM

Convolution neural networks (CNN) [41] are a type of neural networks where the layers employ a special kind of mathematical operation called convolution unlike matrix multiplication in other cases. Mainly used for analysing visual imagery, CNNs have a wide application in the time-series analysis.

Unlike, multi-layer perceptrons (MLP) [42] which are prone to overfitting due to the presence of fully connected layers, CNNs are regularized by taking advantage of the hierarchical pattern in data and hence assemble more complex patterns using smaller and simpler patterns.

As a convolution layer serves well for capturing spatial features, LSTM layers are used to detect correlations over time. However, by stacking these kinds of layers, the correlation between space and time features may not be captured properly. Shi et al. [43] proposed a network structure able to capture spatio-temporal correlations. In the convolution LSTM approach, convolutions are directly used as part of reading input into the LSTM units.

4 Proposed approach

It is to be noted that the air quality data (in our case, PM2.5 and PM10) in different locations vary depending on the degree of industrialization, population density, traffic density, topographical characteristics, etc. [44, 45, 46], and all these factors play an important role in the performance of any time-series forecasting method. The existing literature [46–49] confirms that there is no best single method that can perform well for any given forecasting situation. Hence, a model which is built based on the historical PM2.5 or PM10 data for a particular location may not provide similar accuracy for other locations. Due to this reason, selection of a single forecasting method as a proposed approach may not be realistic; thus, a set of methods instead of one has to be considered so that the best method could be selected based on their performance on location specific data.

The general overview of the approach undertaken in this study is shown in Fig. 2. Missing value imputation is done on the raw data to prepare it for further processing. Time-series analysis is then performed on the imputed data to understand and extract the underlying patterns of the data. The data are then modelled using various statistical and deep learning methods as mentioned in Fig. 1. In order to apply the statistical and deep learning models for long-term forecasting of PM2.5 and PM10 values of Kolkata, instead of directly following any existing implementation, a problem specific version of those models is developed. The models created thus are then used to train on the entire dataset to produce the next two-year forecast, which is then made the basis for the subsequent discussion presented in the latter part of this paper.

The PM10 data after extraction of PM2.5, temperature and relative humidity are found to contain missing values which are to be imputed internally using multivariate imputation and mean before after methods. In contrast to the existing methods [44, 50, 51] where univariate imputations are popularly used for missing value imputation in univariate time-series forecasting, in this work, a combination of univariate and multivariate approach is utilized to improve the missing value imputation ability. Pearson correlation is used to measure the amount of change caused

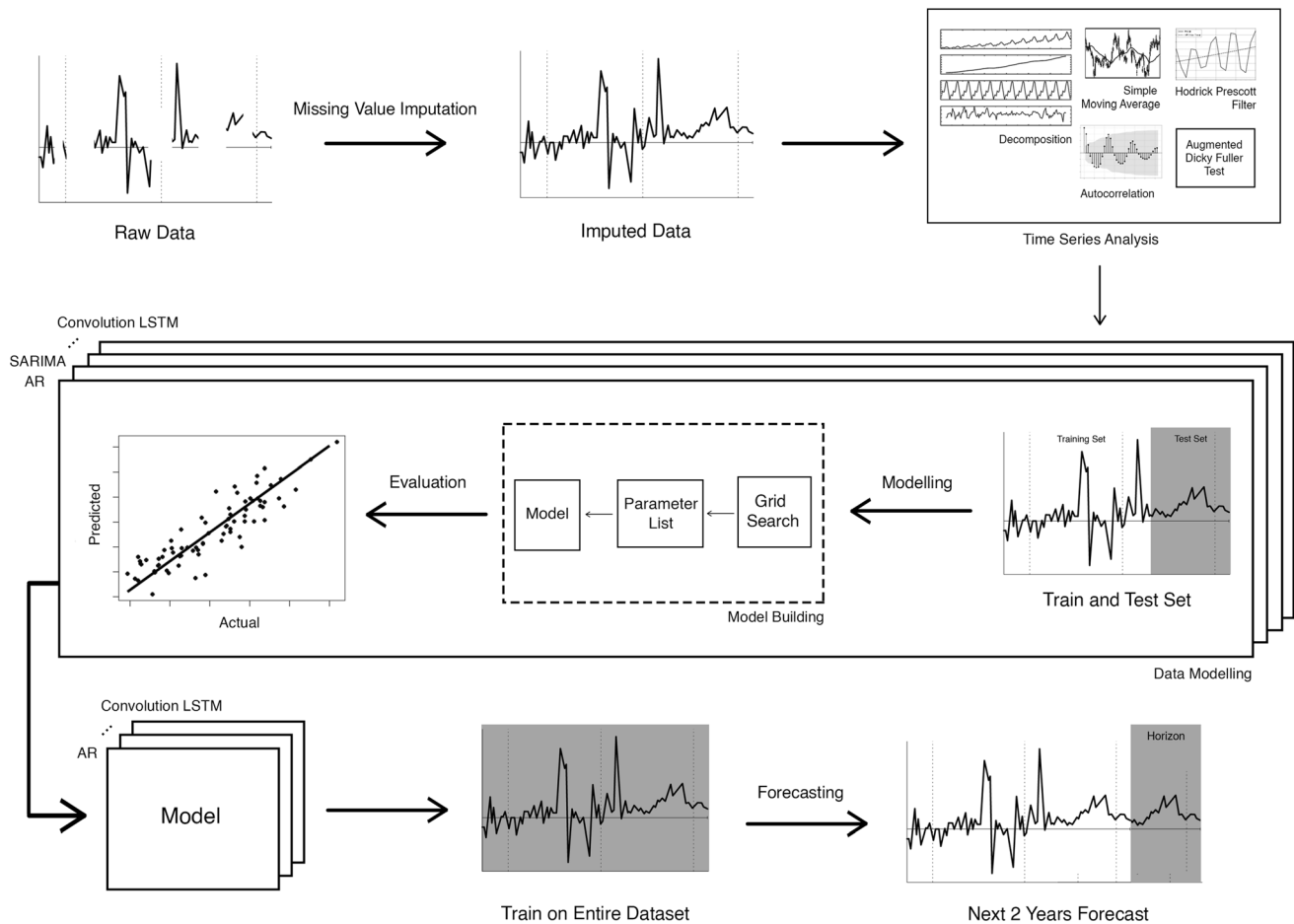


Fig. 2 General overview of the proposed approach

by the imputation. The formula used for calculating the Pearson correlation coefficient ρ is shown in Eq. (31).

$$\rho(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 (y_i - \bar{y})^2}} \quad (31)$$

where x_i, y_i refers to the i th sample in time series X and Y while \bar{x} and \bar{y} refer to the mean of all the samples in X and Y .

On completion of missing value imputation, time-series analysis is performed on the data to understand the underlying patterns present. HP Filter [31] is applied to the daily data to bring out the long-term trends. Although different multiplier values (λ) of HP filter corresponding to different frequencies have been suggested by Ravn and Uhlig [52], due to inadequate data in case of annual resample and the controversy regarding the λ value for monthly resample [52, 53], in this work data are resampled into quarterly and the suggested value of $\lambda = 1600$ is used as the multiplier value.

Next, the simple moving averages are plotted for windows spanning 1 week and 1 month. The daily data are

then resampled into monthly and time-series decomposition is performed, to get a better understanding of the trend and the seasonal components present in the data.

ADF test is performed to determine the stationarity of the time-series data. Many important statistical models require the data to be stationary for complexity reduction and effective analysis [54]. In this study, both PM2.5 and PM10 time series are non-stationary in nature as they show both trend and seasonal patterns; hence in order to model effectively, the times series need to be made stationary. By using repeated ADF tests, the number of lags or difference components is found out based on the p value score, which is then used to turn a non-stationary time series into stationary for further modelling and analysis.

The autocorrelation [32] values are found out on the time-series data and plotted to mathematically determine the seasonality based on statistically significant spikes present.

The monthly data are then trained using four statistical methods and four deep learning methods. In the statistical approach, we use AR, Holt–Winters [35], SARIMA [32] and Prophet [37] to carry out the model fitting and the

subsequent forecasting, while in the case of deep learning, we use four different variations of LSTM, namely stacked LSTM, LSTM auto-encoder, bi-directional LSTM and convolution LSTM [43] models.

Figure 3 describes the model architecture diagrams created using all four variations of LSTM. In case of LSTM auto-encoder-based model architecture as described in Fig. 3a, two LSTM layers l_1 and l_2 consisting of 100 and 50 units, respectively, serve as the encoder, while layers l_3 and l_4 consisting of 50 and 100 units, respectively, serve as the decoder with a repeat vector layer in between. A dense layer d_1 consisting of 1 unit is attached to the encoder–decoder architecture to produce the desired output. As shown in Fig. 3b, the bi-directional LSTM-based model architecture consists of two bi-directional LSTM layers b_1 and b_2 consisting of 200 units followed by a 100 unit LSTM layer l_1 and a dense layer d_1 to model the time series. For the convolution LSTM [43]-based model architecture as described in Fig. 3c, a $1 \times 10 \times 64$ ConvLSTM2D layer is used whose output is flattened and fed to an LSTM layer l_1 consisting of 150 units. The output of the LSTM layer is then provided as input to a 1 unit dense layer d_1 to get the final prediction value as the required output. In case of the stacked LSTM-based model architecture as shown in Fig. 3d, a 1 unit dense layer d_1 following $n = 8$ number of LSTM and dropout regularization layer pairs $ld_i, \forall 1 \leq i \leq n$, consisting of 50 units and

0.5 dropout rate, respectively, is utilized for modelling purposes.

Since the data used to perform the study do not possess a spatial component, the input shape is adjusted accordingly when the data is passed to the convolution LSTM model for training. All deep learning-based models are made to undergo 50 runs, to get a better understanding of the variance introduced due to random initialization of weights in the training process.

The process of training the monthly data in these different models is the same except for the use of min–max scaling in order to normalize the data before passing it into any deep learning-based model. Train/test split is performed in which the last year is made the test set ($\approx 25\%$) and the remaining part is made the training set ($\approx 75\%$).

Hyperparameter optimization is an important part of model building. As finding the set of optimal parameters is a tedious process, manually trying random combinations take a lot of time. To counter this, a parameter sweep (aka Grid-Search) can be done parallelly on different sets of optimal parameters thus reducing the time required in comparison with simple manual searching. Not only is this process faster, but also it is more accurate as all sets of parameters tested compared to few random sets that would have been done if it was performed manually. However, it must be noted that in each of the parallel runs corresponding to a parameter set chosen out of the entire search space, all of the required processes are done sequentially.

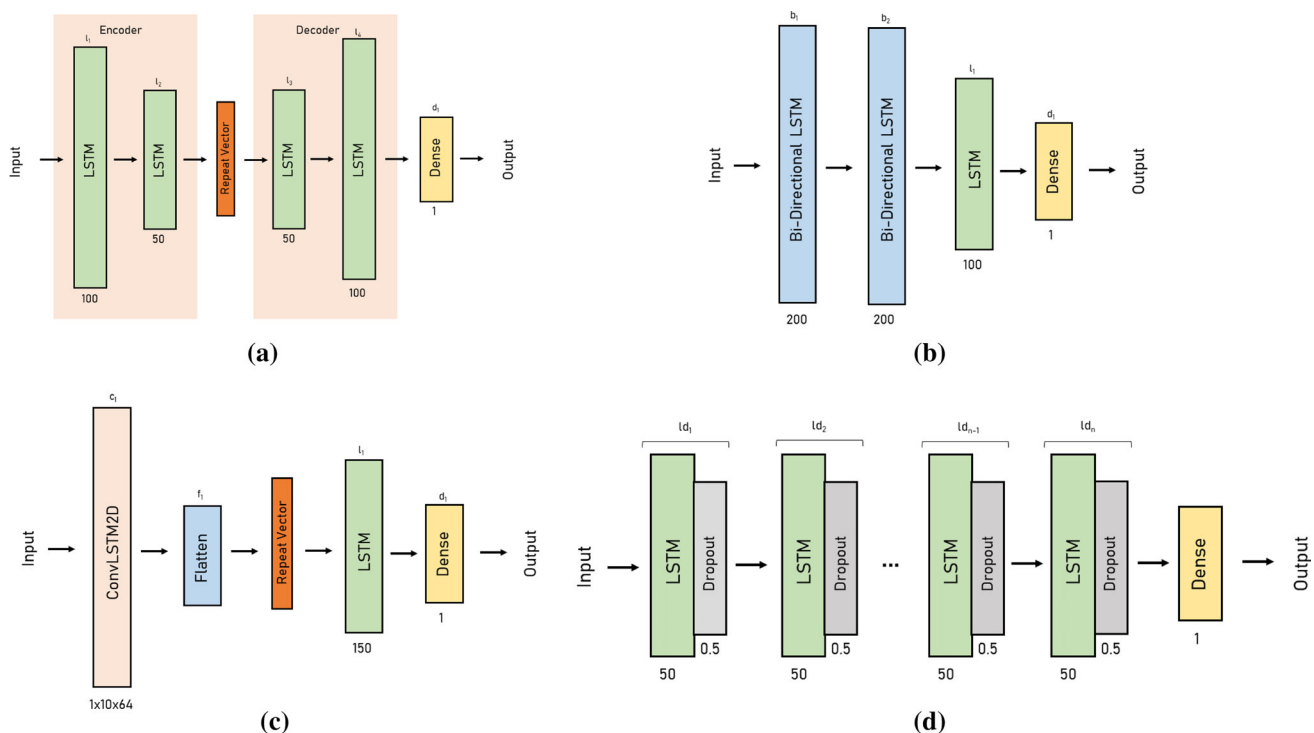


Fig. 3 Model architecture diagrams using **a** LSTM auto-encoder, **b** bi-directional LSTM, **c** convolution LSTM and **d** stacked LSTM

In this study, a detailed search space specific to the model is taken and Grid-Search is performed on it. In case of statistical models different combinations of lags, p , d , q (wherever applicable) are taken into consideration, whereas in case of neural networks, various combinations of the number of epochs, batch size, learning rate and optimizers are taken into consideration. The hyper-parameters are assessed based on a validation set which is created out of the training set. The best combination so found out, are finally evaluated on the test set made earlier. After the evaluation is completed, the entire monthly data are used to train the model, so as to perform the forecast of pollution levels for the next two years.

Walk forward approach (WFA) [55, 56] is used in both evaluation and forecasting. In WFA, first a window spanning a particular time period at the beginning is taken and is used to train and optimize the model. Another segment consisting of the data present right after the end of the window is used to validate the model. After this, the window is rolled over and the process is again repeated till the end of the training data is reached. The model is constantly trained as new data become available, unlike other common approaches which involve model training to happen only with historical data already present. As the real-world performance of the model is one of the key points of this study, WFA turns out to produce a more realistic outcome, especially for time-series data, where information is constantly added with time. In this study, a period of 12 month is taken as the window length for WFA. As the window is rolled over, the immediate next sample is added while the oldest sample is dropped off. The rolling over is continued till the end is reached. The performance on the test set periods denotes the out-of-sample performance of the models and is discussed in detail in the results section of this paper.

5 Results

In this section, the findings of this comparative study involving PM2.5 and PM10 is presented, along with a brief discussion about future trends as projected by the models.

The test bench used to carry out the study involves a 6C/12T Ryzen 5 3600 CPU clocked at 3.6 GHz coupled with 16 GB 3000 Mhz DDR4 RAM and a 1TB NVMe SSD for carrying out the mathematical computations. For deep learning purposes, an Nvidia RTX 2070 Super GPU is also used as a hardware accelerator to speed up matrix-related calculations.

The developmental code for this study was based on python [57], due to the presence of good high-end libraries like numpy [58], tensorflow [59], statsmodels [60] and scikit learn [61] to help in decreasing the overall complexity

of the code without compromising in efficiency and performance.

5.1 Missing value imputation

From the Pearson correlation heatmaps as shown in Fig. 4a, PM10 shows very strong correlation value of 0.82 with PM2.5, compared to temperature and relative humidity. This allowed the imputation of PM10 to be based upon PM2.5 when using the multivariate imputation method as mentioned before.

As observed in Fig. 4, the change in correlation values of PM2.5 and PM10, between the two heatmaps is found to be within a range of 0.1, thus indicating that the underlying patterns of the data were kept intact and preserved. Few missing values which remained in the data were imputed using mean before after method.

5.2 Time-series analysis

From the descriptive statistics presented in Table 2 and the daily time-series plot in Fig. 5a, it can be seen that PM2.5 values are higher in the winter months of December, January and February compared to monsoon months of June and July. It can also be inspected visually in the simple moving average as well as in the monthly plots presented in Figs. 6b, c, respectively, that the peak levels of PM2.5 are on a decreasing trend. This is mathematically confirmed from the dotted line in Hodrick–Prescott [31] plot in Fig. 6a and the decomposition trend plot of monthly data in Fig. 7a. Just like PM2.5, the PM10 values are higher in the winter months compared to the monsoon months. This is clearly evident in the descriptive statistics presented in Table 2 as well in the daily time series, simple moving average and the monthly plots in Figs. 5b–f, respectively. However, unlike PM2.5, PM10 values show an increasing overall trend as can be found out from the HP Filter [31] plot in Fig. 6d and the decomposition trend plot in Fig. 7b. In Figs. 7a, b, the direction of the trend line after 2019 is decreasing in nature, indicating that the pollution levels of both PM2.5 and PM10 declined in the year 2019 compared to previous years. One interesting observation that can be noted from Figs. 6a, d and 4 is that the trends of PM2.5 and PM10 are opposite in nature even though the Pearson correlation coefficient $\rho = 0.88$ is highly positive. Although the results of both the trend and correlation plots seem to contradict each other, there is an actual misconception [62] among many regarding the interpretation of correlation and trends. More specifically, high positive correlation can be possible between two time series even though their trends [calculated using Eq. (5)] are opposite in nature [62]. In Fig. 8, a step-by-step calculation of the Pearson correlation between PM2.5 and PM10 time series

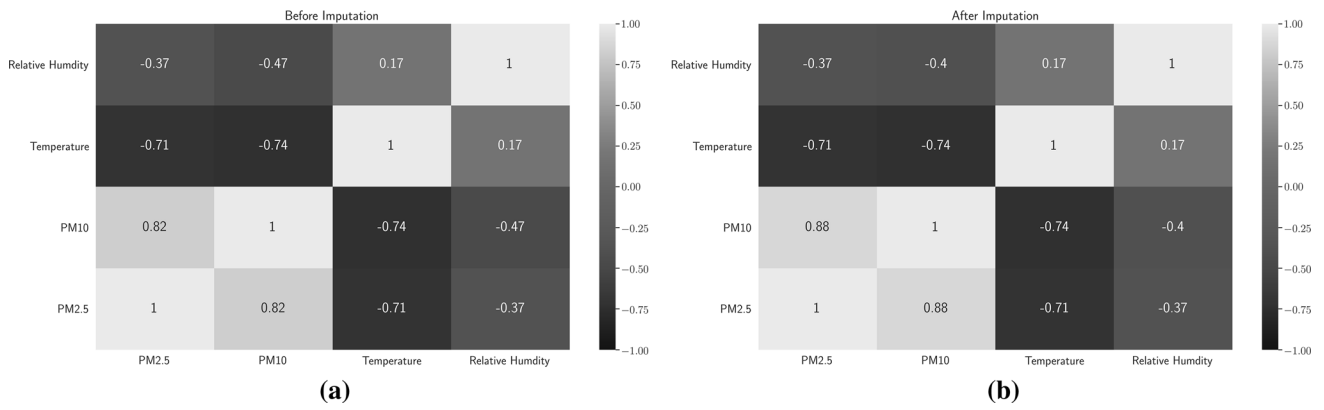


Fig. 4 Pearson correlation heatmaps **a** before imputation and **b** after imputation of PM10

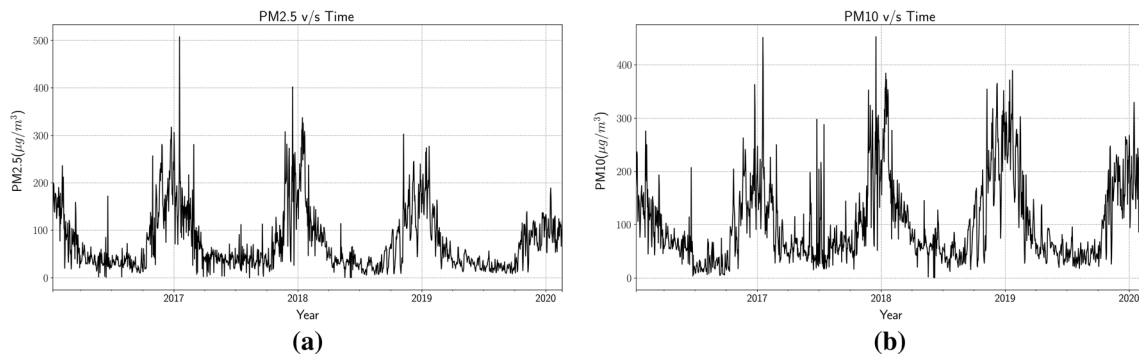


Fig. 5 Daily time-series plots for **a** PM2.5 and **b** PM10

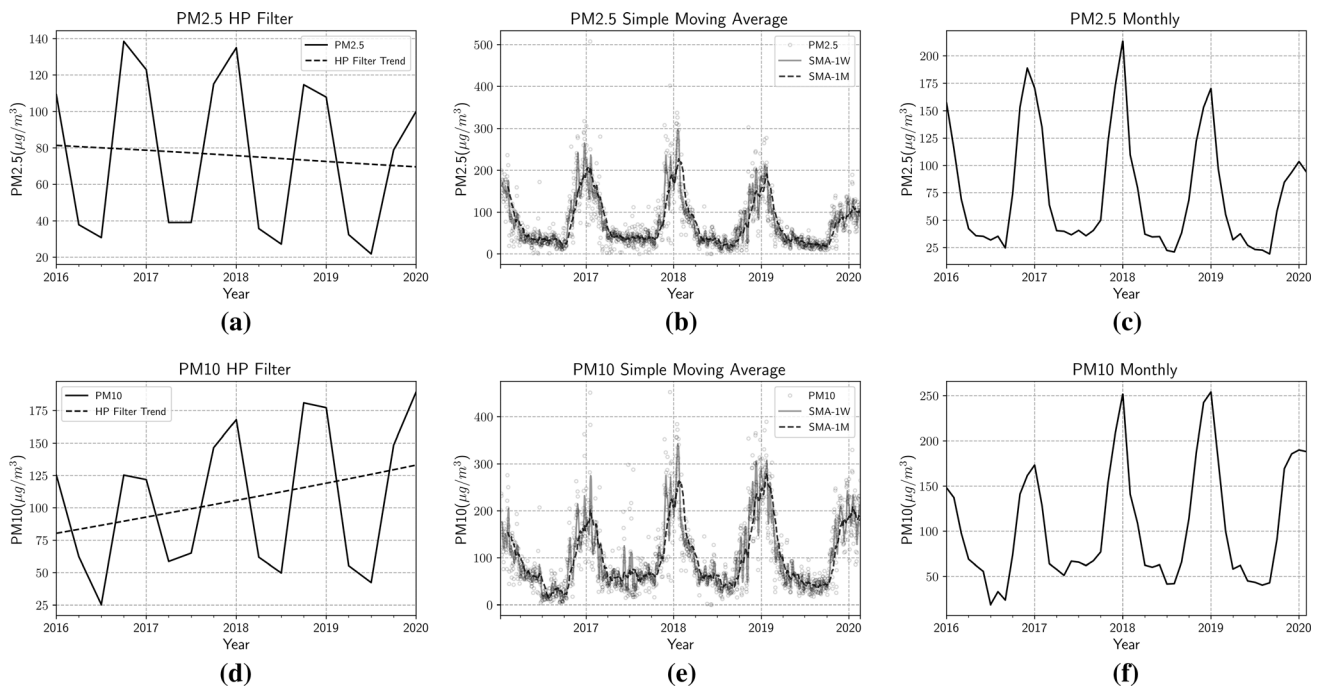


Fig. 6 HP Filter, simple moving average and monthly plots for **a–c** PM2.5 and **d–f** PM10

is provided, using the same data of Fig. 6a, d to validate the claim that there is indeed a strong positive correlation

between PM2.5 and PM10 even though their trends are opposite.

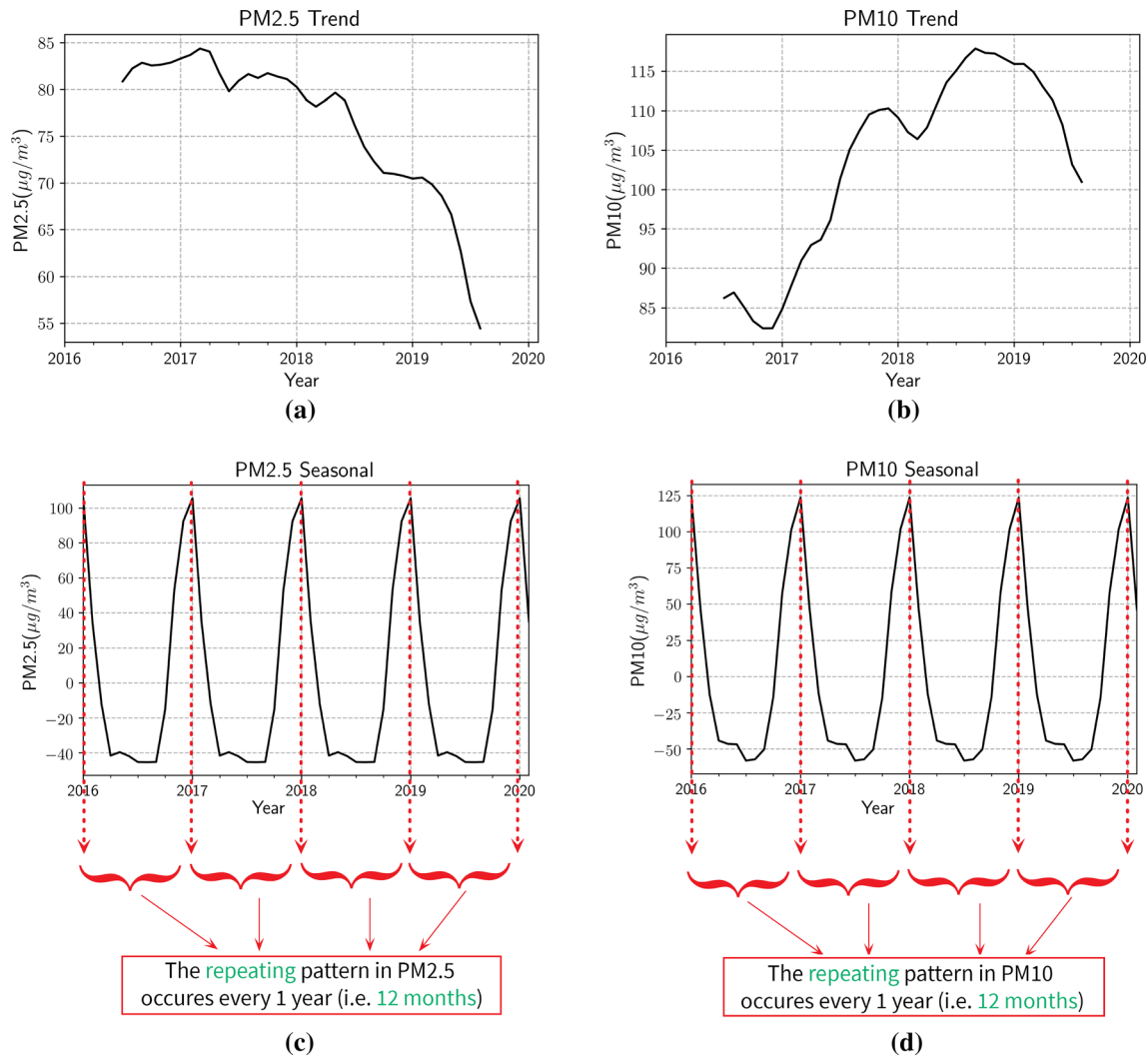


Fig. 7 a–b Trend and c–d seasonal plots for monthly PM2.5 and PM10

Augmented Dicky Fuller test [33] when performed on the monthly data gave us a p value = 0.995 and 0.647 for PM2.5 and PM10, respectively. As a p value greater than 0.05 is considered to be statistically significant, the null hypothesis cannot be rejected and the data are considered to be non-stationary in nature and possess a unit root. The results of the ADF test performed on the non-stationary time-series data differenced by one period gave a p value lesser than 0.05 thereby rejecting the null hypothesis and making it clear that a difference component needs to be present in the statistical models trained on the data.

The seasonal plots using monthly data of both PM2.5 and PM10 in Fig. 7c, d, respectively, indicate a seasonality of 12 time periods as it can be observed that the underlying pattern of the line plots repeats over every 1 year (12 months). It can also be confirmed from the lag having the highest positive correlation (i.e. lag 12) in the set of positive lags after the first set of negative correlation lags in

Fig. 9. The autocorrelation plots also show a gradual decrease to zero in contrast to a sharp decline, thus visually confirming that the time-series data is non-stationary. It is to be also noted, that seasonality of 12 months is not unusual in Kolkata [63, 64]. For instance, in every year, the concentration of particulate matter during winter (Nov–Feb) is higher compared to other seasons because of the longer residence time of particulate matter in the atmosphere during winter due to low winds and low mixing height [64].

5.3 Parameter setting and evaluation Metrics

5.3.1 Parameter setting

The hyper-parameters are obtained using Grid-Search on a search space defined based on the time-series analysis as performed before. Lags = 2 and 9 are produced to give the

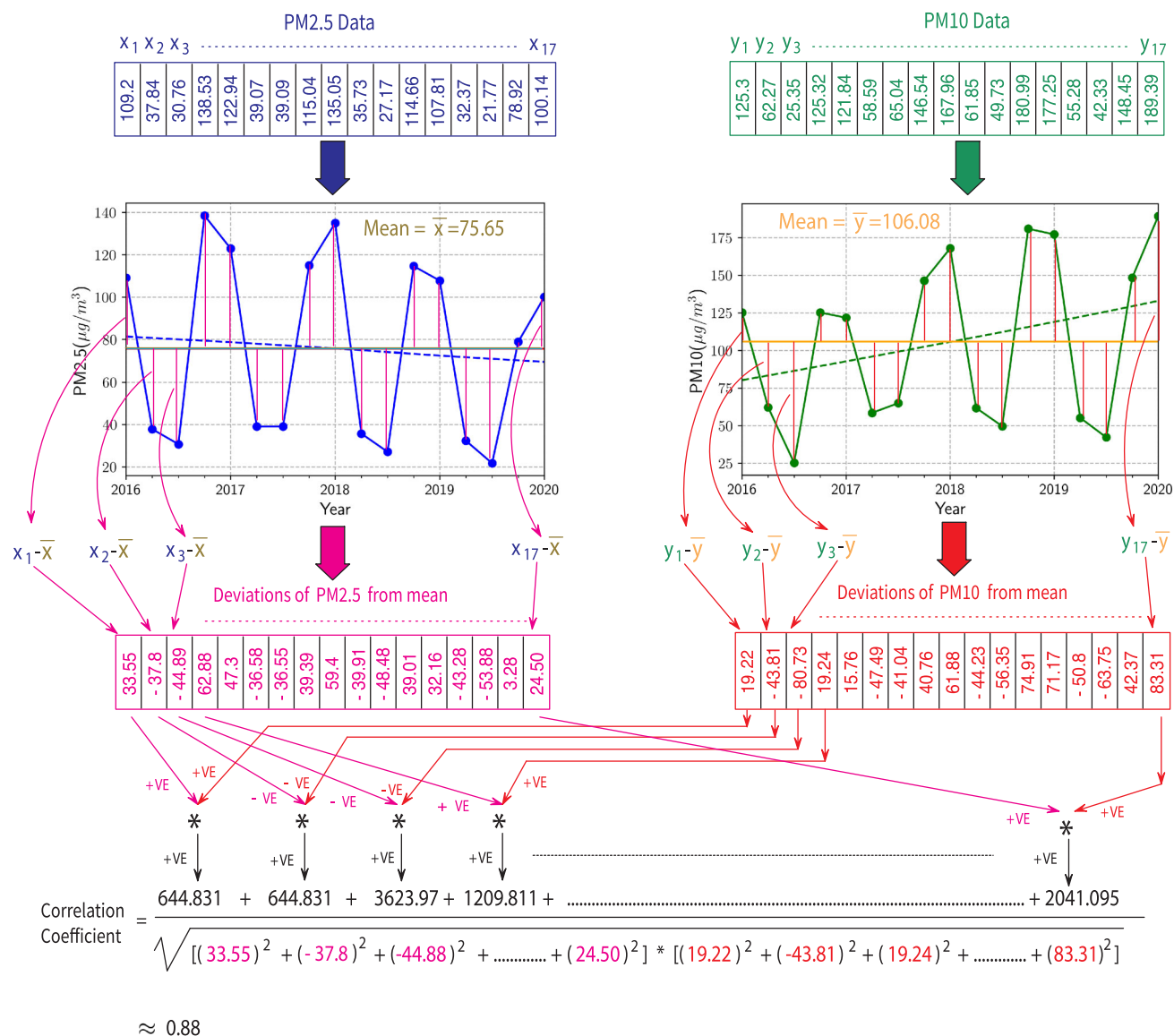


Fig. 8 Flow diagram demonstrating the calculation of the Pearson correlation coefficient. PM2.5 and PM10 data are shown in blue and in green, respectively. The trends (blue and green dotted lines for PM2.5 and PM10, respectively) are opposite in nature. The deviations

in PM2.5 and PM10 from their respective mean (i.e. 75.65 and 106.08) are shown in violet and red colour, respectively (Color figure online)

least RMSE for AR in case of PM2.5 and PM10, respectively. SARIMA (1, 0, 0)(1, 0, 1, 12) for PM2.5 and SARIMA (1, 0, 0)(0, 1, 1, 12) for PM10 are found out to show the best performance in terms of RMSE out of all SARIMA models. Multiplicative trend with seasonality of 12 is used for Holt–Winters, whereas default parameters are taken into consideration for Prophet in case of both PM2.5 and PM10. In case of deep learning models, all models are trained on a minimum of 100 epochs with batch sizes ranging from 1 to 64, based on the hyper-parameters found out by Grid-Search using a constant seed for reproducibility.

5.3.2 Evaluation metrics

Measurement of model performance is based on root mean squared error (RMSE) and mean average error (MAE) in comparison with the test set mean. The effect of each error on RMSE is proportional to the size of the squared error; thus, larger errors have a disproportionately larger effect on the RMSE.

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^T (\hat{y}_t - y_t)^2}{T}}, \quad \text{MAE} = \frac{\sum_{t=1}^T |\hat{y}_t - y_t|}{T} \tag{32}$$

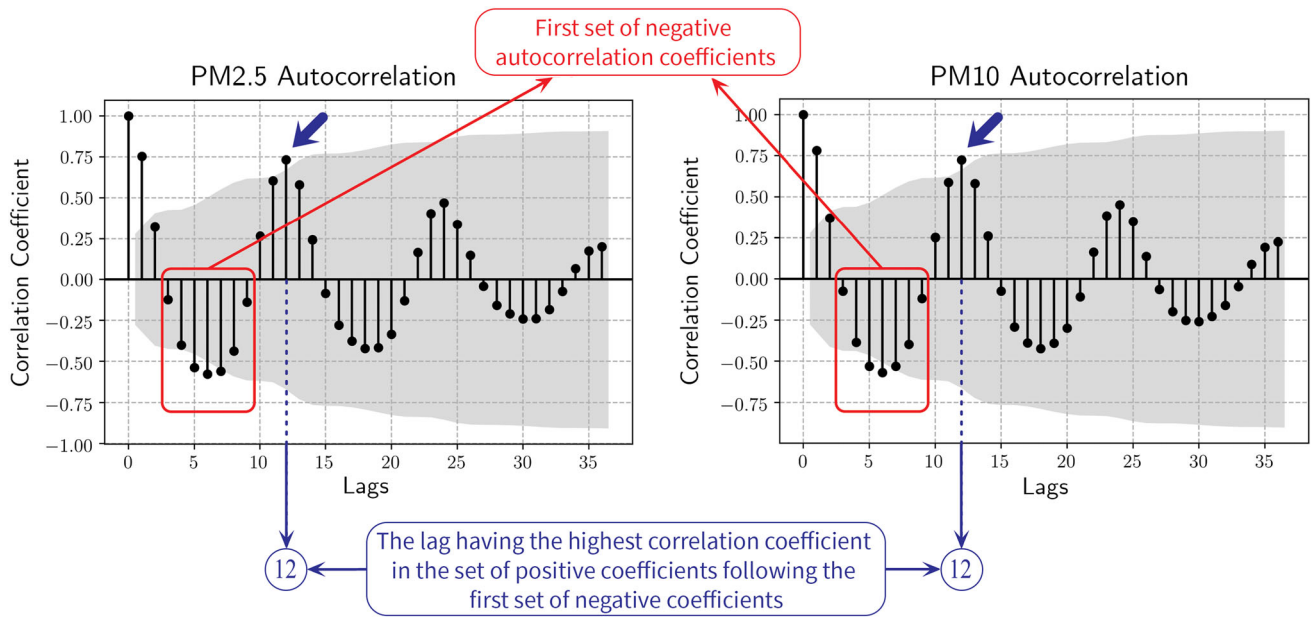


Fig. 9 Autocorrelation plots for monthly PM2.5 and PM10. The blue arrow marks show the lag having the highest positive correlation (i.e. lag 12) in the set of positive lags after the first set of negative correlation lags

where \hat{y}_t is the prediction made by the model and y_t is the actual value at instant t . Here, T denotes the count of the number of time-series samples.

Due to comparatively longer training time for deep learning models, two epochs were used to retrain the models during each WFA cycle in both evaluation as well as in forecasting.

5.4 Forecasting

As can be seen in Tables 3 and 4, out of the four statistical models and four deep learning models used to fit the data, Holt–Winters gave the overall best RMSE and MAE score combination, while in case of deep learning convolution LSTM [43] and stacked LSTM gave the best results in terms of RMSE and MAE with respect to a test set mean of 54.36 and 101.41 for both PM2.5 and PM10, respectively. The actual vs predicted correlation plots in Figs. 10 and 11 show that Holt–Winters (in Fig. 10b, f) has the best model performance compared to others. One interesting observation from Fig. 11 is that the values predicted by deep learning models are relatively scattered more in comparison with their statistical counterparts in Fig. 10, which is further reflected in their RMSE and MAE values.

Now, the forecast plots of PM2.5 and PM10 are shown in Figs. 12 and 13, respectively, for different statistical and deep learning models where the shaded portion representing the forecast region. On analysing the nature of the forecasts produced by different models as shown in Figs. 12 and 13, AR, stacked LSTM, bi-directional LSTM

Table 3 Performance metrics of statistical models for PM2.5 and PM10

Pollutant	Model	RMSE	MAE
PM2.5	AR	15.68	13.08
	SARIMA	12.19	10.12
	Holt–Winters	10.06	7.72
	Prophet	31.87	24.27
PM10	AR	21.98	19.48
	SARIMA	20.53	16.07
	Holt–Winters	15.45	11.33
	Prophet	39.57	35.58

Bold values indicate the best performing models with the respect to the metrics mentioned

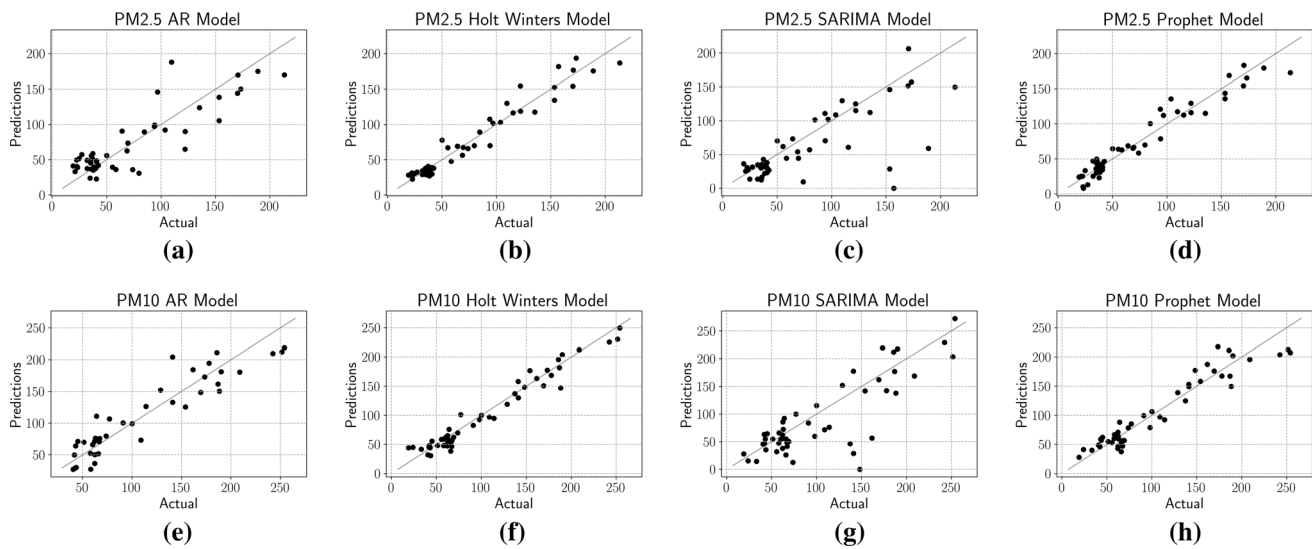
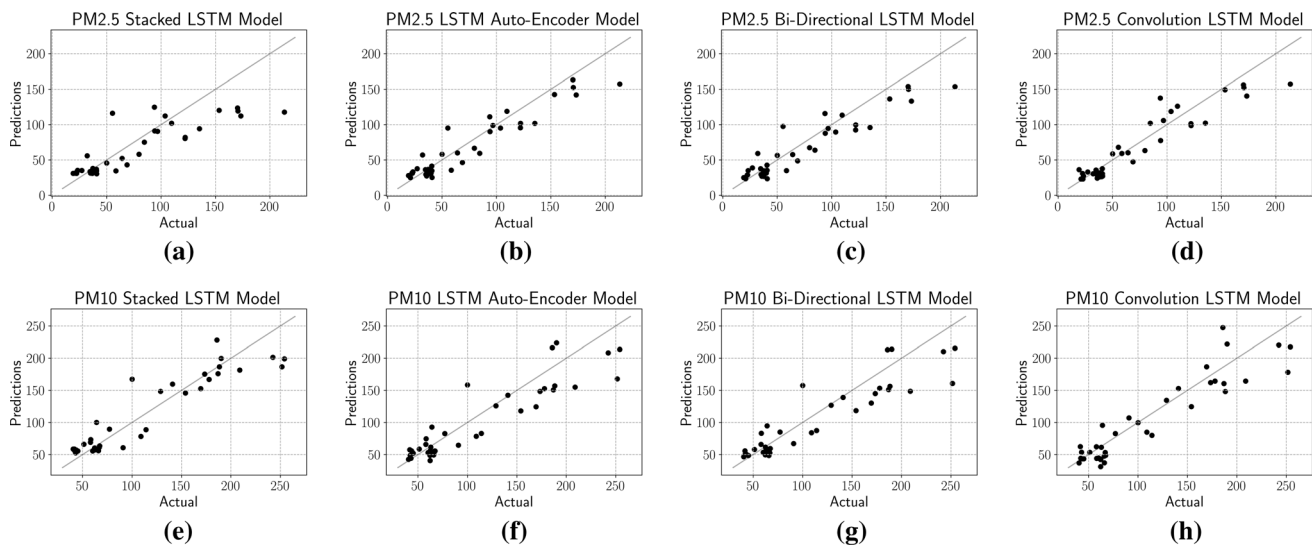
and LSTM auto-encoder (in Fig. 12a, c–g) showed a tendency of converging to the mean in the long-term for PM2.5. Although Prophet (as shown in Fig. 12d) was able to pick up the trend and the seasonal components clearly, the forecast produced became negative in the period between 2021 and 2022. The decrease in PM2.5 levels over the years as forecasted by SARIMA (in Fig. 12c) was relatively lower compared to Holt–Winters and convolution LSTM models as can be seen in Fig. 12b, h.

The behaviour, however, was a little different for PM10 where none of the models showed any explicit tendency of converging to the mean. Like PM2.5, Holt–Winters, SARIMA and convolution LSTM models, as evident from Fig. 13b, c and h accurately were able to extract the trend

Table 4 Performance metrics of deep learning models

Pollutant	Model	RMSE	MAE	Train time (in s)
PM2.5	Stacked LSTM	22.32 ± 1.76	16.62 ± 1.10	33.06 ± 1.51
	LSTM auto-encoder	18.88 ± 0.19	15.88 ± 0.19	9.50 ± 1.13
	Bi-directional LSTM	19.27 ± 0.98	16.57 ± 0.56	11.01 ± 0.86
	Convolution LSTM	16.98 ± 1.18	12.16 ± 0.97	4.39 ± 0.78
PM10	Stacked LSTM	29.33 ± 3.41	21.59 ± 2.01	19.81 ± 1.44
	LSTM auto-encoder	33.35 ± 6.40	26.58 ± 3.58	5.19 ± 0.46
	Bi-directional LSTM	29.92 ± 5.64	23.78 ± 2.96	13.36 ± 0.84
	Convolution LSTM	29.92 ± 4.35	22.73 ± 4.40	3.94 ± 0.49

Bold values indicate the best performing models with the respect to the metrics mentioned

**Fig. 10** Actual vs predicted scatter plots of statistical models for (a-d) PM2.5 and (e-h) PM10**Fig. 11** Actual vs predicted scatter plots of deep learning models for a–d PM2.5 and e–h PM10

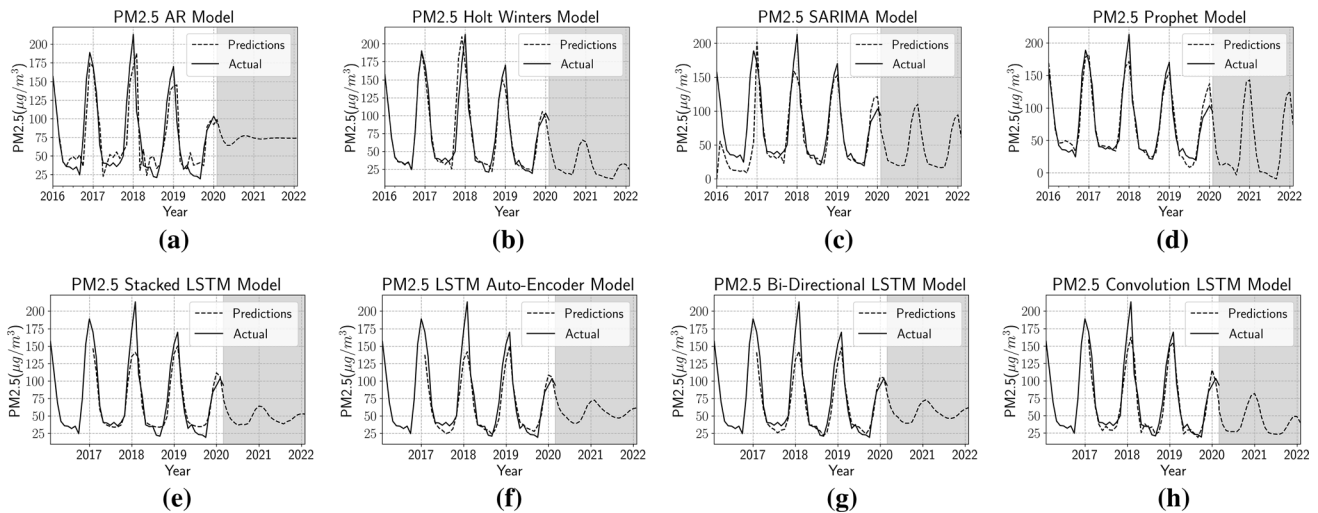


Fig. 12 PM2.5 forecast plots for statistical and deep learning models with the shaded portion representing the forecast region

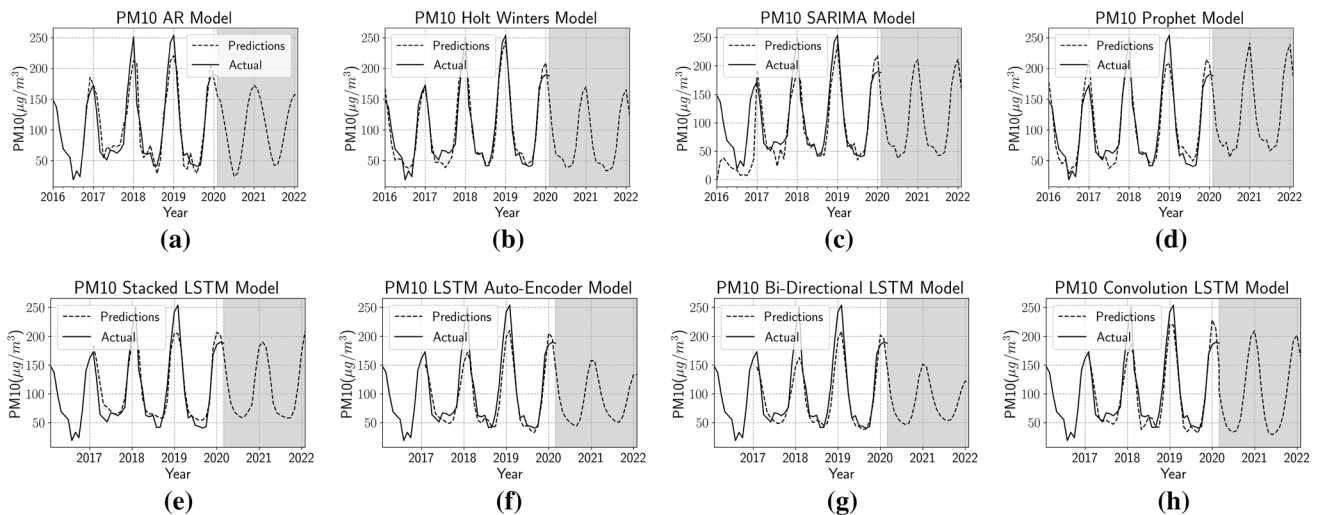


Fig. 13 PM10 forecast plots for statistical and deep learning models with the shaded portion representing the forecast region

and the seasonal components and produce a practical forecast. However, all models did not show a similar trend. In case of AR, Holt–Winters, bi-directional and auto-encoder LSTM (as shown in Fig. 13a, b, f–g), a decreasing trend could be seen in the future years. SARIMA (in Fig. 13c) showed a constant forecast while Prophet and stacked LSTM (in Fig. 13d, e) produced a forecast following an increasing overall trend. Out of all deep learning models that were a part of the study, except convolution LSTM, all models showed a forecast which was decreasing in nature. Convolution LSTM as can be seen in Fig. 13h produced a forecast having an increasing trend just like Prophet.

From the performance metrics in Tables 3 and 4, statistical methods performed better compared to deep learning. This performance difference can be attributed to the quantity of data available. As monthly data are

considered in this approach, the quantity of data will be limited in all practical situations; hence, statistical methods will be found to give better results.

5.5 Discussion

The decrease in PM2.5 pollution levels and the concave downward trend in PM10 levels as indicated by the forecasts can be a good indication of the recent measures taken by the Government of West Bengal and the Central Government to bring down pollution levels in Kolkata.

However, some forecasts showing a positive upward trend are still cause for alarm, as the present quality of PM10 levels is already significantly higher than the safe limit of $20\mu\text{m}^3$ prescribed by the WHO [65]. Even PM2.5 levels are significantly higher compared to the global safe limit of $10\mu\text{m}^3$.

The government should continue adopting strict policies regarding environmental pollution, especially focussing on large scale industries that are the main causes of PM10 levels. A complete ban of dumping sand, stone chips and other construction raw materials openly on roadsides should also be a part of their action plan to curb pollution. Measures such as promoting the usage of electric vehicles or vehicles based on CNG or LPG, a complete ban on the incineration of garbage in public places can be a part of an action plan set up by the government to curb PM2.5 levels in the city.

6 Conclusion

This study undertook a quantitative approach to understand the future trends of PM2.5 and PM10 based on historical pollution data extracted from various sources. The most widely used time-series modelling methods were put to the test to carry our long-term forecasts, and their efficiency was compared with each other. Based on the limited data available, statistical methods especially Holt–Winters were able to outperform deep learning methods. If the quantity of data available would have been higher, or if the proposed approach is used to forecast the next few months by using weekly resampled data, deep learning models could be expected to perform relatively better.

However, a certain shortcoming of this study is the absence of the use of exogenous variables. Although methods like Holt–Winters and AR can be used to model time-series data efficiently, those methods do not have the flexibility to account for exogenous variables. If exogenous variables were made a part of this study, models like SARIMAX and LSTMs could be expected to give more accurate results.

Even though the city taken in this study was Kolkata, the approach used in this study can be applied to any major city in the world. Based on the forecasts, concerned policy-making organizations can implement new measures and regulations to curb the pollution levels in their cities and make the environment healthier for the city's inhabitants.

Curbing pollution levels also will have a major positive impact on the environment. PM particles adversely affect ecosystems including plants, soil, water, etc. Water quality gets degraded and plant growth and yield also get largely affected. It is hoped that this study will help the policy makers to judge the gravity of the pollution scenario in their cities and aid them to implement better pollution measures.

This study was performed on data before the COVID-19 pandemic-related lockdown was enforced in Kolkata. Due to a huge reduction in socio-economic activity, the pollution forecasts performed may change drastically from the

actual values. Based on the changes in the nature of the pollution data during and after the lockdown, a study analysing those changes can be presented in the future.

Acknowledgements This comparative study was supported by the project entitled—“Participatory and Realtime Pollution Monitoring System For Smart City”, funded by the Department of Science and Technology, Government of West Bengal, India.

Declarations

Conflicts of interest The authors declare that they have no conflicts of interest.

References

1. Mahajan S, Chen LJ, Tsai TC (2017) An empirical study of PM2.5 forecasting using neural network. <https://doi.org/10.1109/UIC-ATC.2017.8397443>
2. Xiang X (2019) Forecasting air pollution PM2.5 in beijing using weather data and multiple kernel learning. *J Forecast.* <https://doi.org/10.1002/for.2599>
3. Xie J (2017) Deep neural network for PM2.5 pollution forecasting based on manifold learning. In: 2017 international conference on sensing, diagnostics, prognostics, and control (SDPC), pp 236–240
4. Luo C, Yang H, Huang L, Mahajan S, Chen L (2018) A fast PM2.5 forecast approach based on time-series data analysis, regression and regularization. In: 2018 conference on technologies and applications of artificial intelligence (TAAI), pp 78–81
5. Feng X, Li Q, Zhu Y, Hou J, Jin L, Wang J (2015) Artificial neural networks forecasting of PM2.5 pollution using air mass trajectory based geographic model and wavelet transformation. *Atmos Environ* 107:118–128. <https://doi.org/10.1016/j.atmosenv.2015.02.030>. <http://www.sciencedirect.com/science/article/pii/S1352231015001491>
6. Haiming Z, Xiaoxiao S (2013) Study on prediction of atmospheric PM2.5 based on RBF neural network. In: 2013 4th international conference on digital manufacturing automation, pp 1287–1289
7. Yan L, Wu Y, Yan L, Zhou M (2018) Encoder–decoder model for forecast of PM2.5 concentration per hour. In: 2018 1st international cognitive cities conference (IC3), pp 45–50
8. Cortina-Januchs MG, Quintanilla-Dominguez J, Vega-Corona A, Andina D (2015) Development of a model for forecasting of PM10 concentrations in Salamanca, Mexico. *Atmos Pollut Res* 6(4):626–634. <https://doi.org/10.5094/APR.2015.071>. <http://www.sciencedirect.com/science/article/pii/S1309104215301951>
9. Al-kasassbeh M, Sheta A, Faris H, Turabieh H (2013) Prediction of PM10 and tsp air pollution parameters using artificial neural network autoregressive, external input models: a case study in salt, jordan. *Middle-East J Sci Res* 14:999–1009. <https://doi.org/10.5829/idosi.mejsr.2013.14.7.2171>
10. Lam LH, Mok KM (2007) Prediction of ambient pm10 concentration with artificial neural network. In: *Computational methods in engineering and science*. Springer, Berlin, Heidelberg, p 276
11. Das M, Maiti SK, Mukhopadhyay U (2006) Distribution of PM2.5 and PM10-2.5 in PM10 fraction in ambient air due to vehicular pollution in Kolkata megacity. *Environ Monit Assess* 122(1–3):111–123
12. Jiao K, Xu M, Liu M (2018) Health status and air pollution related socioeconomic concerns in urban china. *Int J Equ Health* 17(1):1–11

13. Ong BT, Sugiura K, Zettsu K (2016) Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM_{2.5}. *Neural Comput Appl* 27:1553–1566. <https://doi.org/10.1007/s00521-015-1955-3>
14. Bashir Shaban K, Kadri A, Rezk E (2016) Urban air pollution monitoring system with forecasting models. *IEEE Sensors J* 16(8):2598–2606
15. Tao Q, Liu F, Li Y, Sidorov D (2019) Air pollution forecasting using a deep learning model based on 1D convnets and bidirectional GRU. *IEEE Access* 7:76690–76698
16. Liang X, Zou T, Guo B, Li S, Zhang H, Zhang S, Huang H, Chen S (2015) Assessing Beijing's PM_{2.5} pollution: severity, weather impact, apec and winter heating. *Proc R Soc A: Math, Phys Eng Sci* 471:257. <https://doi.org/10.1098/rspa.2015.0257>
17. Mlakar P (1997) Determination of features for air pollution forecasting models. In: *Proceedings intelligent information systems, IIS'97*, pp 350–354
18. Li T, Hua M, Wu X (2020) A hybrid CNN-LSTM model for forecasting particulate matter (PM_{2.5}). *IEEE Access* 8:26933–26940
19. Wang W, Guo Y (2009) Air pollution PM_{2.5} data analysis in los angeles long beach with seasonal arima model. In: *2009 international conference on energy and environment technology*, vol 3, pp 7–10
20. Bai L, Wang J, Ma X, Lu H (2018) Air pollution forecasts: an overview. *Int J Environ Res Public Health* 15(4):780
21. Kurt A, Gulbagci B, Karaca F, Alagha O (2008) An online air pollution forecasting system using neural networks. *Environ Int* 34(5):592–598
22. Xu X (2020) Forecasting air pollution PM_{2.5} in Beijing using weather data and multiple kernel learning. *J Forecast* 39(2):117–125
23. Norazian MN, Shukri YA, Azam RN et al (2008) Estimation of missing values in air pollution data using single imputation techniques. *ScienceAsia* 34(3):341–345
24. Bandyopadhyay K (1644) Banned vehicles found plying in kolkata in november. *Times News Network*. <http://timesofindia.indiatimes.com/articleshow/73062554.cms>
25. MacNee W, Donaldson K (2003) Mechanism of lung injury caused by PM₁₀ and ultrafine particles with special reference to COPD. *Eur Respir J* 21(40 suppl):47s–51s. <https://doi.org/10.1183/09031936.03.00403203>. https://erj.ersjournals.com/content/21/40_suppl/47s
26. Ministry of Environment, Forest and Climate Change, Govt. of India: Central Pollution Control Board. <http://www.cpcb.nic.in/>. Accessed 15 Aug 2020
27. Kissock JK University of Dayton Average Daily Temperature Archive. <http://academic.udayton.edu/kissock/http/Weather/>. Accessed 15 Aug 2020
28. The Weather Company (IBM): Weather Underground. <https://www.wunderground.com/>. Accessed 15 Aug 2020
29. US Department of State: Air Now International US Embassies and Consulates. <https://www.airnow.gov/international/us-embassies-and-consulates/>. Accessed 15 Aug 2020
30. Buck SF (1960) A method of estimation of missing values in multivariate data suitable for use with an electronic computer. *J R Stat Soc, Ser B (Methodol)* 22(2):302–306. <http://www.jstor.org/stable/2984099>
31. Hodrick RJ, Prescott EC (1997) Postwar U.S. business cycles: an empirical investigation. *J Money, Credit Banking* 29(1):1–16. <http://www.jstor.org/stable/2953682>
32. Box GEP, Jenkins G (1990) *Time series analysis, Forecasting and control*. Holden-Day Inc, USA
33. Fuller WA (1976) *Introduction to statistical time series*. Wiley, New York
34. Manuca R, Savit R (1996) Stationarity and nonstationarity in time series analysis. *Phys. D: Nonlinear Phenom* 99(2–3):134–161
35. Winters PR (1960) Forecasting sales by exponentially weighted moving averages. *Manag Sci* 6(3):324–342. <https://doi.org/10.1287/mnsc.6.3.324>
36. Walker GT (1931) On periodicity in series of related terms. *Proc R Soc Lond, Ser A, Contain Pap Math Phys Character* 131(818):518–532
37. Taylor SJ, Letham B (2018) Forecasting at scale. *Am Stat* 72(1):37–45
38. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
39. Kramer MA (1991) Nonlinear principal component analysis using autoassociative neural networks. *AIChE J* 37(2):233–243. <https://doi.org/10.1002/aic.690370209>
40. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. *IEEE Trans Signal Process* 45(11):2673–2681
41. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86:2278–2324. <https://doi.org/10.1109/5.726791>
42. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: *Rumelhart DE, McClelland JL (ed) Parallel distributed processing: explorations in the microstructure of cognition*. Foundations, vol 1. MIT Press, pp 318–362
43. Shi X, Chen Z, Wang H, Yeung DY, Wong WK, Woo WC (2015) Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: *Proceedings of the 28th international conference on neural information processing systems*, vol 1. MIT Press, pp 802–810
44. Cortina-Januchs MG, Quintanilla-Dominguez J, Vega-Corona A, Andina D (2015) Development of a model for forecasting of PM₁₀ concentrations in Salamanca, Mexico. *Atmos Pollut Res* 6(4):626–634
45. Middy A, Roy S, Dutta J, Das R (2020) Jusense: a unified framework for participatory-based urban sensing system. *Mob Netw Appl* 25:1249–1274
46. Dutta J, Chowdhury C, Roy S, Middy A, Gazi F (2017) Towards smart city: sensing air quality in city based on opportunistic crowd-sensing. In: *Proceedings of the 18th international conference on distributed computing and networking*. Association for Computing Machinery
47. Wang X, Smith-Miles K, Hyndman R (2009) Rule induction for forecasting method selection: meta-learning the characteristics of univariate time series. *Neurocomputing* 72(10–12):2581–2594
48. Armstrong JS (2001) *Principles of forecasting: a handbook for researchers and practitioners*, vol 30. Springer
49. Meade N (2000) Evidence for the selection of forecasting methods. *J Forecast* 19(6):515–535
50. Box GE, Jenkins GM, Reinsel GC, Ljung GM (2015) *Time series analysis: forecasting and control*. Wiley
51. Moritz S, Bartz-Beielstein T (2017) imputeTS: time series missing value imputation in R. *R J* 9(1):207
52. Ravn MO, Uhlig H (2002) On adjusting the Hodrick–Prescott filter for the frequency of observations. *Rev Econ Stat* 84(2):371–376
53. Borio C (2014) The financial cycle and macroeconomics: what have we learnt? *J Bank Finance* 45:182–198
54. Kirchgässner G, Wolters J, Hassler U (2012) *Introduction to modern time series analysis*. Springer
55. Żbikowski K (2015) Using volume weighted support vector machines with walk forward testing and feature selection for the purpose of creating stock trading strategy. *Expert Syst Appl* 42(4):1797–1805

56. Kirkpatrick CD II, Dahlquist JA (2010) *Technical analysis: the complete resource for financial market technicians*. FT Press
57. Van Rossum G, Drake FL Jr (1995) *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands
58. van der Walt S, Colbert SC, Varoquaux G (2011) The Numpy array: a structure for efficient numerical computation. *Comput Sci Eng* 13(2):22–30
59. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado G, Davis A, Dean J, Devin M et al (2015) TensorFlow: large-scale machine learning on heterogeneous systems
60. Seabold S, Perktold J (2010) Statsmodels: econometric and statistical modeling with python. In *Proceedings of the 9th python in science conference*, p 61
61. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. *J Mach Learn Res* 12:2825–2830
62. Lhabitant FS (2011) Correlation vs. trends: a common misinterpretation. <https://risk.edhec.edu/sites/risk/files/1328885974025.pdf>. Accessed 15 Aug 2020
63. Jayamurugan R, Kumaravel B, Palanivelraja S, Chockalingam M (2013) Influence of temperature, relative humidity and seasonal variability on ambient air quality in a coastal urban area. *Int J Atmos Sci* 2013:1–7
64. Karar K, Gupta AK, Kumar A, Biswas AK (2006) Seasonal variations of PM10 and TSP in residential and industrial sites in an urban area of Kolkata, India. *Environ Monitor Assess* 118(1–3):369–381
65. World Health Organization: Ambient (outdoor) air quality and health. [https://www.who.int/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-andhealth](https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-andhealth). Accessed 15 Aug 2020

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.