Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

Forecasting Air Quality: A Comparative Study of Time Series Approaches

Satya Dev Pasupuleti[1] | **Simone A. Ludwig**[2]

[1]Computer Science, North Dakota State University, ND, USA

[2]Computer Science, North Dakota State University, ND, USA

Correspondence
Corresponding author Satya Pasupuleti,
Email: satya.pasupuleti@ndsu.edu

Present address
1320 Albrecht Blvd, NDSU Computer Science, Room 258, Fargo, ND 58102

Abstract

Air pollution is a major problem in many countries and especially in India. In November 2019, New Delhi recorded an air quality index level of 900, which is considered higher than the 'severe' level. The air pollution forecasting method predicts the pollution based on the available dataset and the data features and certain method performance in forecasting at high accuracy depends on the method and the measures used. Forecasting air quality levels in countries like India is very important because it has a direct impact on public health, and thus, is used for decision making. The main goal for this paper is to investigate air quality index prediction based on different algorithms, so experts can identify the methods that require development and it is useful as a starting point for novice researchers. The problem of the Air Quality Index (AQI) prediction in this paper is approached with different Fuzzy Inference Systems (FIS), Neural Networks, Swarm intelligence techniques and so on. The results of the experiments shows that the LSTM model performs better than all the deep learning based models and Fuzzy based models discussed in this paper

KEYWORDS
Air Quality, Time Series, ANFIS, LSTM, ARIMA, Forecasting, CNN, ARIMA-LSTM, Prophet, Fuzzy Logic, DENFIS

1 | INTRODUCTION

Air pollution is comprised of air pollutants introduced in the environment that have adverse effects[1]. Air pollution can cause damage to the health of human beings, plants, and animal species. Therefore, monitoring and forecasting air pollution is crucial for all countries to propose control measures for air quality.

Particulate matter (PM) is the term used for the composition of solid and liquid particles present in the air. PM comes in different sizes, large enough to see them with the naked eye and very small and only visible with the aid of a microscope. PM is divided based on the size and the most common Particle pollution types found in the air are PM_{10} and $PM_{2.5}$. PM_{10} describes particles with a diameter of 10 micrometers or smaller, and $PM_{2.5}$ describes particles with a size of 2.5 micrometers or smaller. Both these particle types pose a great risk to the health due to their small size and they can get inside the lungs and bloodstream of human beings quite easily[2].

According to a 2021 report that measured air pollution on the basis of $PM_{2.5}$ levels, there are 14 cities in India in the top 20 most polluted cities in the world[3]. Beijing city in China reduced the average $PM_{2.5}$ levels from 104.5 in the year 2010 to 27.5 in the year 2020[4]. This result was possible because of the Chinese government efforts of increasing their number of existing federal monitoring stations to three times between the years 2012 and 2020. The data generated from these monitoring stations helped the city's regulators to impose air quality regulations[5].

The air quality index data is a stochastic time series model, and therefore, can be forecasted using historical data. In this paper, we would like to forecast the AQI (Air Quality Index) of the Delhi (India) region. Currently, Delhi is one of the top 10 cities that have the highest AQI and is listed as the world's most polluted capital since 2019. Delhi's extreme pollution is caused by the heavy smoke from the thermal plants in and around the city, public transportation, automobiles, and frequent biomass burning of agricultural lands in neighboring states. Accurate forecasting of air quality is important for the cities like

Delhi to aid in modifying the environment policies by the government and to adjust the protection measure proposals by the health department[6].

The data for this paper is taken from the AirNow website[7] where the Environmental Protection Agency (EPA) calculates the current and forecasted air quality for all the states in the USA and their embassies all over the world. EPA started issuing the AQI values in 1976, and frequently updates its health-based air quality standards in accordance with the changes in the earth's atmosphere. EPA issued AQI values considering all the five major pollutants that are used in measuring the pollution in the air. The pollutants are PM, carbon monoxide, Ozone, Nitrogen Dioxide, and Sulfur Dioxide.

The main pollutants that are considered by EPA for AQI calculations are Particulate matter-based AQI and Ozone-based AQI. The time frame for both the pollutants are different as PM-based AQI is calculated within a 24-hour time frame, and ozone based AQI values are calculated within an 8-hour time frame.

If the value of AQI is greater than 100 (which is a moderate level) then it is harmful for people with lung disease, old adults, children, people who are active outdoors, and people with certain genetic variants. EPA categorized the AQI values into 6 types ranging from values of 0 to 300 and higher, and each category is assigned a representative color. EPA records the data from their forecasters from different states and embassies. The forecasters use many satellite images, computer forecasting models to calculate the air pollution concentration. In addition to the computer-generated models, the forecasters use their knowledge about the pollution behavior to issue the AQI forecasts. EPA calculates AQI values in the data set called 'Nowcasts' using two algorithms. These algorithms calculate hourly readings from the air quality monitors based on the AQI for Ozone and AQI for particle pollution. The Nowcast algorithm provides current hour data by using a calculation that involves past multiple hours data. It uses the longer averages when there are constant air quality values, and shorter averages when the air quality value is changing swiftly[7].

The aim of this paper is to survey different weather forecasting approaches to forecast the Air Quality Index of Delhi city and to evaluate which method is efficient in forecasting using the model's metrics MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error).

We implemented the machine learning and deep learning based methods LSTM, LSTM+Autoencoder, LSTM+ARIMA, Autoregressive LSTM and the fuzzy inference system based methods ANFIS, DENFIS, Wavelet-HFCM and ABC+ANFIS and lastly Facebook's Prophet and the time series forecasting algorithm ARIMA. All the above methods are implemented using the Python programming language. Our ABC+ANFIS hybrid implementation seems to be the first to be implemented and applied to time-series data.

The rest of the paper is organized as follows: Section 2 goes through the previous work related to weather related forecasting methods. Section 3 gives the introduction for all the ten approaches applied in this paper. Section 4 explains the experiments and results, and finally Section 5 concludes the paper.

2 | RELATED WORK

2.1 | Overview of Time Series Data

Time series data is the measurement of observations through repeated measurements over time. Time series data is everywhere now because time is a constituent of everything that is observable. Every industry is incrementally dependent on the instruments, the electronic devices like sensors and machines that generates continuous stream of time series data[38]. This data is increasing at a vigorous rate with the smart devices we use in our daily life. The time series data is playing a pivot role in health, finance, technology, retail industries, and so on. The time interval can be measured at any time intervals like seconds, minutes, hours, days, weeks, months and years. The most important examples of the time series data are hourly air quality, electric consumption, stock market trend, brain and heart activities[39].

The observation at time points $t_1, t_2, t_3 \ldots t_T$ are expressed as $y(t_1), y(t_2), y(t_3), \ldots\ldots y(t_T)$. The usage of different prediction models depends on the range the values to be predicted. In the one step ahead prediction model, the prediction of each data point is computed using the trend states and level of computation for current data points and the seasonal state for the previous seasonal period. In the K-step ahead prediction model, the prediction values are computed based on the observed time series data[40].

Researchers have applied many statistical models, numerical models, machine learning, and deep learning models for the forecasting of different features in the weather. The statistical models are considered to be more accurate, and they use historical

data to predict the future data based on certain variables in the data. Several linear and auto-regression models fall under the statistical approaches but because of their shortcoming to consider the dynamic nature, this results in inaccurate forecasts. Numerical models rely on simple assumptions, and are therefore, not suitable for short-term predictions[8].

2.2 | Hybrid Forecasting Models Based on Deep Learning

Machine learning models consider several parameters, and Artificial Neural Networks (ANN), in particular the Multilayer Perceptron (MLP), is most popular for forecasting AQI[9]. Other research studies used hybrid models or a combination of models based on deep learning techniques like Neural Networks for prediction. The exponential Smoothing approach is used to implement the air pollutant matter prediction because of its simple implementation and its ability to incorporate the trend and seasonality[10]. Support Vector Machine (SVM) and the moments approach is used in the paper where the model is based on the concept of training SVM with the image features that are extracted by stability, color moments, color correlogram, and wavelet features. This method yields an accuracy of 82.3%[11].

The LSTM cells and parallel Dense NNs (DNN) were applied to forecast the AQI values in one paper[12]. In this paper, the addition of DNN is to consider the weather, temperature, and other factors that affect AQI. The hybrid model of DNN and LSTM performed well compared to LSTM and it was concluded that the performance increased with the increment of the time period.

The researchers constructed three prediction models to predict the hourly $PM_{2.5}$ concentration and of those models, the Encoder-Decoder model performed better than the other two models achieving a Root Mean Square Error value of 43.17[13]. The authors used Wavelet Analysis to pre-process the AQI data and applied a combination model using ARIMA and LS-SVM to predict the AQI values. The combined model performed well with higher prediction accuracy than the single model[14].

The authors focused on the performance of a time series model as supervised learning and proposed LSTM, CNN and LSTM-CNN to perform the air pollution forecasting. The authors concluded that CNN has the lowest error value whereas CNN-LSTM has the lowest error value without the normalization task performed[15].

A meta-heuristic model with gaussian process is implemented to determine the best parameter setting for the Recurrent Neural Network algorithm, and thus, the gaussian processes allows the algorithm to work with a Bayesian probabilistic framework. The authors model performed well with the lowest forecasting error of 0.0488 compared to Recurrent neural Networks and Back-propagation neural network with both having error values of 0.0516 and 0.15464, respectively[41].

The Gated Recurrent Unit + CNN is used to train the pre-processed data, and to extract the temporal features and the spatial features of the air pollutant concentration data. Then, the temporal and spatial features are integrated into the kernel extreme learning machine model. The authors model performed well compared to K-Nearest Neighbor, Decision Tree, Support Vector Machine and Random Forest models[42].

A novel DL framework combining An innovative multitask multichannel (MTMC) and multiple nested long short term memory networks (NLSTM) is proposed for accurate AQI forecasting enlightened with the federated learning[43]. The training and forecasting process of the multi-variate AQI data was performed in parallel along with the federated learning. The experimental results show that the performance of the proposed method is superior to conventional ML models, DL methods, as well as hybrid DL models

2.3 | Hybrid Forecasting Model Based on Fuzzy Inference Systems

The authors discussed about the development of a model based on the neuro-fuzzy inference system for predicting the short term of particulate matter concentration in the air[16]. The same authors experimented with ANFIS to predict the next hour PM concentration with different parameters as the input for the system[17].

A new approach with a combination of LSTM and a Fuzzy evaluation system is implemented and in this model, the AQI values of a Taiwanese city are generated based on the top 3 pollutants $PM_{2.5}$, O_3, and PM_{10} to compute two types of forecasts - cross-hour Direct Forecast (DF) and Update Forecast (UF). The results of experiments show that the DF accuracy decreases with the increase in the forecasted time, whereas the UF accuracy remains stable. The proposed LSTM-Fuzzy evaluation system achieved almost 100% accuracy within 12 hours of time period[43].

The authors used the ANFIS model in combination with artificial bee colony (ABC) to predict the tunnel boring machine performance in order to avoid the disastrous incident in rock mechanics and engineering and we call this method as ABC+ANFIS

(Artificial Bee Colony + Adaptive Neuro-Fuzzy Inference System) as part of our experiments. The ABC method is used to find the optimum ANFIS membership function to increase the model accuracy [19].

A new Fuzzy inference system called dynamic evolving neural-fuzzy inference system (DENFIS) for dynamic time series prediction is introduced [20]. During the experiment, the model creates and updates new fuzzy rules and DENFIS uses the fuzzy inference system based on most activated fuzzy rules that are chosen dynamically from a set of fuzzy rules. The authors implemented two approaches in which one approach creates a first order Takagi-Sugeno Fuzzy rule set for the online DENFIS model and in another approach created the same type rule set or an expanded higher order one for the offline DENFIS model.

The Wavelet transform is combined with the higher order fuzzy cognitive maps to work with the large scale non stationary time series data. The authors used wavelet transform method to decompose the large-scale times series into multi-variant time series and then applied the higher order fuzzy cognitive maps for modeling and prediction of the time series that is obtained using the wavelet transform [21] .

3 | COMPARISON APPROACHES

This section discusses the approaches that are applied to the Air Quality Index data. Both the Fuzzy inference system and the deep learning approach are used in this paper. The Fuzzy based forecasting methods used in the experiments are ANFIS (Adaptive Neuro-Fuzzy Inference System), High-Order Fuzzy Cognitive Maps and Wavelet Transform (Wavelet-HFCM), Artificial Bee Colony (ABC) algorithm mixed with the ANFIS model, and Dynamic Evolving Neural-Fuzzy Inference System (DENFIS). The deep learning forecasting methods used in the experiments are LSTM (Long Short-term Memory), ARIMA (Auto-regressive Integrated Moving Average), CNN (Convolution Neural Networks), Auto-regressive LSTM, ARIMA-LSTM, and Prophet.

3.1 | Long Short-term Memory Neural Network

The problem with the classic recurrent neural networks (RNN) is the vanishing gradient problem. The gradient is so small that it will vanish, which means almost reaches zero and that will prevent the weights to be updated during backpropagation. If the vanishing gradient problem is severe, then it may stop the neural network from further training. LSTM can partially overcome this problem. Since each neuron acts as a memory cell and connects the previous data to the current neurons. Each neuron in LSTM contains three gates namely the input gate, forget gate, and output gate.

As shown in Figure 1, the input gate, which is also called the sigmoid layer, determines which information is needed to update and pass on to the forget gate. The forget gate decides which information will be removed from the cell, the input gate determines which new incoming information to remember in the cell state, and the output gate decides which information will be the output in the cell state [12].

3.2 | Auto-regressive Integrated Moving Average

ARIMA is a linear method that deals with the prediction of time-series data. The size is independent of the memory cost and the parameters of the model are estimated online in an efficient way. The ARIMA model is based on 3 parameters (Equation 8):

1. Order of the auto-regressive model (p) – A model that uses the relationship between observations and lagged observations.
2. Order of integrated model (q) – This model is used for differencing to make the data stationary.
3. Order of moving average (MA) model (d) – This model uses the dependency between the observation and residual error from the MA model, which is applied to lagged observations.

$$\text{ARIMA(p,d,q)} \tag{1}$$

Afterwards, we determine the values of the auto-correlation and partial correlation coefficients. Finally, we perform the prediction using the model and parameter values, and evaluate the model efficiency using the performance metrics [22].

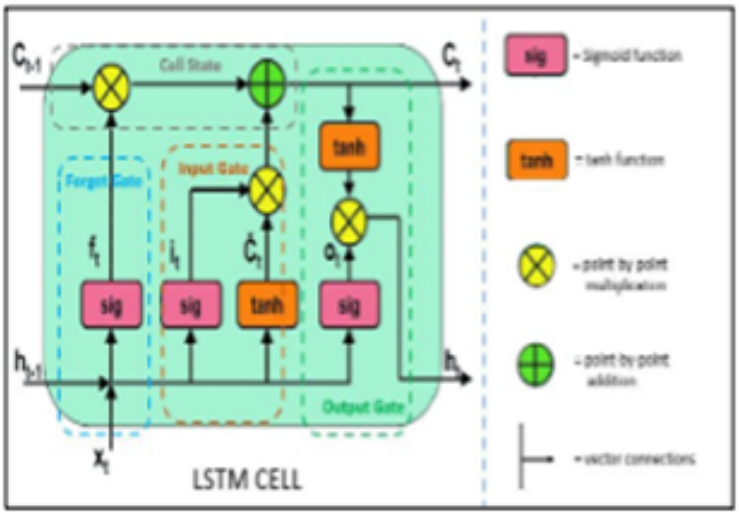

FIGURE 1 LSTM NN Architecture [12]

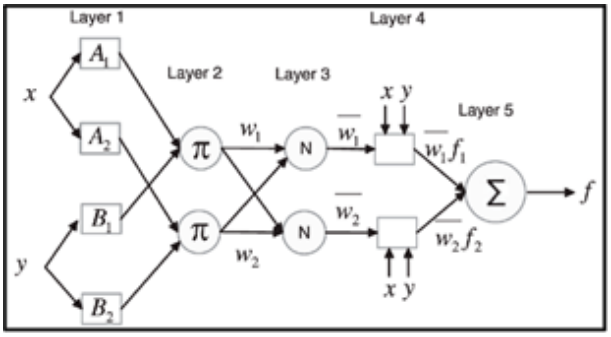

FIGURE 2 ANFIS Architecture [23]

3.3 | Adaptive Neuro-Fuzzy Inference System

ANFIS is a hybrid architecture that is composed of a Neural Network (NN) and Fuzzy Inference System (FIS), and it, therefore, takes advantage of both methods to model data uncertainty. ANFIS can take both the numerical and linguistic knowledge as the input, and it is also more transparent compared to NN.

The FIS part in the ANFIS model contains five units – (1) Fuzzification unit that changes a crisp value to the fuzzy set, the (2) defuzzification unit that converts a fuzzy set back to the crisp value at the end, and the (3) database unit that comprises Membership Functions (MFs) for both the input and output variables, (4) the rule base unit that contain FIS rules, and (5) decision rules that operate based on fuzzy rules. The ANFIS architecture contains five layers based on the Sugeno rules as shown in Figure 2.

The first layer determines the MFs based on the input values, the second layer computes the product of MFs, the third layer normalizes the sum of the inputs, the fourth layer calculates the contribution of the rules to the output, and the fifth layer sums all the inputs [23].

3.4 | Convolutional Neural Network (CNN)

Figure 3 shows the CNN architecture. CNN is mostly used for 2D image classification but it is also used for time series forecasting. The model learns a function that is used for sequence mapping of previous input data to output data. The CNN architecture consists of an input layer followed by hidden layers that comprise of one or more convolution layers based on the data sequences. After that follows the pooling layer, which is used to process the output of the convolution layer to extract meaningful elements.

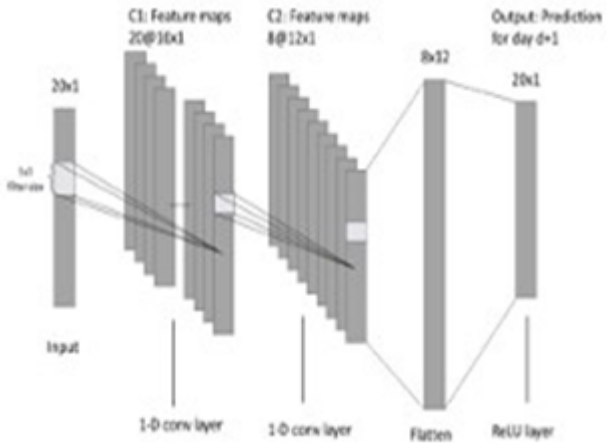

FIGURE 3 CNN Architecture [18]

The densely fully connected layer is used to interpret the salient features extracted by the convolution part. There is also a flatten layer to reduce the features to a one-dimension vector [28]. CNN views the data as a sequence instead of the time series and functions the same way as for the classification of a one-dimensional image.

3.5 | Facebook Prophet

The Prophet model was developed by Facebook as an open-source software and it is available in both Python and R languages. As the Prophet model is very sensitive to missing data, the data is pre-processed before model application. During the pre-processing steps, the missing data is filled with the average of the univariate column and the outliers are removed.

Since the model deals with only 2 columns, it performs well with seasonality, which can be set as annual, daily or weekly. The hourly seasonality is applied in this case as our data consists of AQI values for every hour [29].

3.6 | Autoregressive LSTM

There are two types of predictions for time series data. Single time step predictions where at each time only a single data point is predicted using the previous data points, and there is a second one called the multi-time step model where it predicts a sequence of values in future. The Auto-regressive LSTM model is a multi-step model. For this model, we trained the model on all the training data and the model learns to predict the future values using the value given as the window size. For this, the window size is set as 24, which means the model is trained and applied to predict 24 hours into the future.

We use LSTM for the training and then run an auto-regression loop. Therefore, the architecture consists of a LSTM layer and then a dense layer, which converts the output of the LSTM layer to predictions, and then the predictions will be sent back using the auto-regressive loop [31].

3.7 | Autoencoder+LSTM

Figure 4 shows the Autoencoder+LSTM architecture. This model is taken from the solution proposed in a research paper [25] and is modified to suit the time series data used for this paper. For this method, the features were extracted from the data by training using a simple LSTM Autoencoder. These new features were combined with the new input and fed to the LSTM prediction model. For this model, the raw concentration of AQI and NowCast concentration values were added to the data to check how these extreme features will affect the outcome of the predictions.

The workflow of this model is to train the data using the autoencoder, and then later feed it to the LSTM prediction model and further train it to predict the output for a certain window size. The newly created LSTM Autoencoder contains a LSTM encoder and then a LSTM decoder.

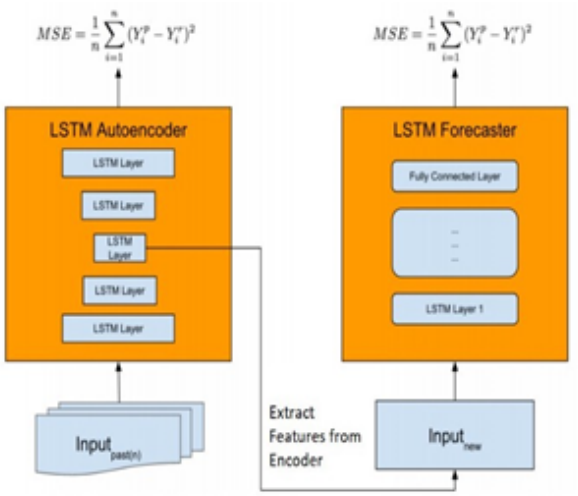

FIGURE 4 Autoencoder+LSTM Architecture [24]

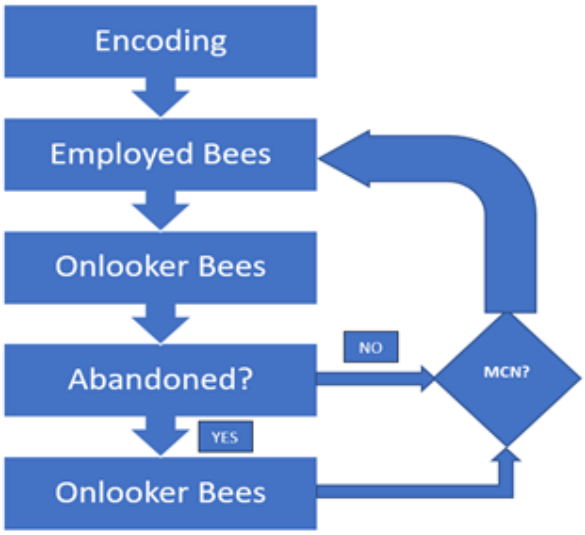

FIGURE 5 Framework ABC+ANFIS [19], MCN is Maximum Cycle Number

3.8 | Artificial Bee Colony + Adaptive Neuro-Fuzzy Inference System

In addition to constructing an ANFIS based on the 5 layers described in the ANFIS section, we combine Artificial Bee colony (ABC), which is an optimization algorithm (OA). The role of ABC in this hybrid model is to optimize he membership functions of the ANFIS model to achieve a higher accuracy [19].

The Artificial Bee colony is a type of evolution-based algorithm, which is also a search method based on natural selection and genetics. ABC type algorithms use operators like crossover, mutation and natural selection and it is strongly adaptable and self-organizing algorithm. In each generation, the entire population will be replaced by a new generation using the above operators. ABC uses foraging behavior of bees for exploring the most accessible and richest food sources. The ABC algorithm contains 3 types of bees – Foraging bees, onlooker bees and scout bees. Each iteration of ABC, the bees with best fitness is selected as the forager bee and the others are selected as scout bees. The role of the onlooker bees is to evaluate the nectar information of forager bees so that it adjusts its moving trajectory in the next iteration. The scout bees control the diversity of the population and the forage bees help with the exploitation of the search space [24].

As described by Karaboga, who proposed the ABC algorithm[24], the possible solution of the optimization problem is represented as the position of food sources, and the amount of nectar in the food source represents the probability or fitness of the proposed solution, which is shown in Figure 5. Each food source is searched by only one employed bee, which explains that the number of food sources represent the number of employed bees that exist in the hive. If the food source of an employed bee is abandoned, then it becomes a scout bee. Half of the portion of the colony is occupied by employed bees, and the other half is occupied by the onlooker bees[24].

In the first step of ABC[26], the initial food sources are randomly generated using Equation 2:

$$X_{ij} = X_{min,j} + rand(0,1)(X_{max,j} - X_{min,j}) \tag{2}$$

where $i = 1, 2, 3, \ldots..., SN$, and $j = 1, 2, 3, \ldots..., D$.

Equation 2 represents the vector of structure of a food source or solution with size D. SN is the population size of the user; D is the population size of the solutions; $rand(0,1)$ is a function that produces a random value between 0 and 1; $X_{min,j}$ and $X_{max,j}$ are the lowest and highest values of the cell j; i is the index of the solution that is selected randomly; and j is a random number between 1 and D.

An employed bee modifies a solution based on Equation 3:

$$V_{i,j} = X_{i,j} + \theta_{i,j}(X_{i,j} - X_{j,k}) \tag{3}$$

where $\theta_{i,j}$ is a random value in the range of [-1, 1]; V_i, X_i, X_k are new, current, and the neighbor individuals, respectively; and k is the randomly selected solution's index. The nectar amount is computed for every generated new solution. When the new solution is generated, the employed bee remembers it's position and the bee moves to that position and checks whether the new solution's quality is better than the old one. Equation 4 is used in the probabilistic selection process to calculate the chance of a solution. This process happens in the onlooker bee phase.

$$P_i = \frac{f_i}{\int_{n=1}^{SN} f_n} \tag{4}$$

The probability value P_i is calculated based on the nectar amount of the food sources evaluated by the employer bees; f_i is the quality of the individual i. Once P_i is computed, it is compared with a random value between 0 and 1. If the P_i is larger than the value, the onlooker bees go to the area X_i located at the food source to determine a new neighboring food source.

Each bee generates a new solution once onlooker bees reach the neighborhood. The employed bee leaves the neighborhood, if the new solution quality is not improved by the number of iterations that are given. After leaving the neighborhood, the employee bee is converted to a scout bee.

ABC is used here to optimize the antecedent and conclusion parameters of the ANFIS model. Most of the OAs need different control parameters and the adjustment of certain parameters will affect the search process such as slow convergence, trapping in local minima. Unlike other OAs, the ABC optimization algorithm is not depended on the hyper parameters and the only parameters that are set by the user are population size and the maximum iteration.

3.9 | A Dynamic Evolving Neural-Fuzzy Inference System (DENFIS)

As it is proposed in[20], the Takagi–Sugeno type fuzzy inference engine is used in both the online and offline DENFIS models. This DENFIS inference engine is composed of m fuzzy rules and the rules are $m \times q$ fuzzy propositions since m antecedents form m fuzzy rules, respectively.

All fuzzy membership functions are triangular type functions in both, the online and offline DENFIS models. If we have the crisp constants consequent functions, then it is a zero-order Takagi–Sugeno type fuzzy inference system, If we have the linear functions, then it is a first-order Takagi–Sugeno type fuzzy inference system, and finally system is called high-order Takagi–Sugeno fuzzy inference system, if the functions are nonlinear functions.

The first-order Takagi–Sugeno type fuzzy rules are employed in the DENFIS online model and the linear functions in the consequences can be created and updated by the training data's linear least-square estimator (LSE)[27,28].

The Takagi–Sugeno fuzzy inference system used in DENFIS is a dynamic inference. The DENFIS online model is different from other inference in creating and updating fuzzy rules. Moreover, they also differ in the way they chose rules which is from a fuzzy rule set for forming a fuzzy inference system. This method of choosing a rules system depends on the current position

of the input vector in the input space. If there are two input vectors that are close in position to each other, they might assign the same fuzzy rule inference group in the inference system in the DENFIS offline model, but might assign a different fuzzy rule inference group in the DENFIS online model. This happens because the two input vectors might be presented to the system at different times, and fuzzy rule for the first might get updated before the second input vector arrives. The Takagi–Sugeno fuzzy inference system used in DENFIS also varies in assigning antecedents of the fuzzy rules, which depend on the current position of the input vector in the input space.

3.10 | High-Order Fuzzy Cognitive Maps with Wavelet Transform (Wavelet-HFCM)

Fuzzy Cognitive Maps (FCMs) were first introduced by Bart Kosko in 1986[27] as part of their research on causal reasoning. As per Kosko the "The FCMs are the fuzzy-graph structures for representing causal reasoning. Their fuzziness allows hazy degrees of causality between hazy causal objects (concepts). Their graph structure allows systematic causal propagation, in particular forward and backward chaining, and it allows knowledge bases to be grown by connecting different FCMs".

The weighted graphs are used to represent the FCMs, which includes N_c concept nodes, and state values of these concepts are denoted as a vector A[20]:

$A = [A_1, A_2, ..., A_{Nc}]$

where $A_i \in [0, 1]$ (or $A_i \in [-1, 1]$), $i = 1, 2, ..., N_c$.

The state value of each node represents its activation degree. The relationships between nodes are defined as an $Nc \times Nc$ weight matrix W as shown in Equation 5:

$$W = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1Nc} \\ w_{21} & w_{22} & \dots & w_{2Nc} \\ \vdots & \vdots & \vdots & \vdots \\ w_{Nc1} & w_{Nc2} & \dots & W_{NcNc} \end{pmatrix} \tag{5}$$

where w_{ij} stands for the relationship originating from the *jth* node and pointing to the *ith* node and taking the value from the range of $[-1, 1]$, where $i, j = 1, 2, ..., Nc$.

Thus, the dynamics of FCMs can be described by Equation 6:

$$A_i(t+1) = \psi \times \int_{j=1}^{Nc} (W_{ij} A_j(t)) \tag{6}$$

where $Ai(t)$ is the activation level of node i at the j^th iteration, and ψ is the transfer function mapping the activation level into the unit range. The sigmoid function works as transfer functions in FCMs in most of the cases. But when the node's value gets negative, the tanh function is necessary to be used. Wavelet transform is a signal analysis method, which is more accurate and detailed, and therefore, can be successfully applied to the extraction of the signal and analysis of the non-stationary signal[28].

The discrete wavelet transform's decimation effect won't be suitable for forecasting the time series. The redundant Haar wavelet transform solves this problem by transforming the wavelet coefficient's length as the same as the input time series length at each resolution level. In order to learn the high order fuzzy cognitive maps, the redundant Haar wavelet transform of the original time series data yields the historical time series data.

4 | EXPERIMENTS

In this section, we discuss the description of the data, experiment parameters, model results, and the comparison of the results.

4.1 | Data Description

The data set used in this paper is extracted from the AirNow.gov website. The data contains historical hourly data of Delhi city air pollution ranging from February 2015 to February 2023. For this research investigation, we considered the timestamp, Air Quality Index, $PM_{2.5}$ level readings, and AQI that is divided into 6 categories based on the AQI levels.

The Environmental Protection Agency (EPA) calculated AQI values in this data set using two algorithms called 'Nowcasts'. These algorithms calculate hourly readings from the air quality monitors, based on the AQI for Ozone and AQI for particle pollution. The Nowcast algorithm gives current hourly data by using a calculation that involves past multiple hours data. It uses longer averages when there are constant air quality values and shorter averages when the air quality value is changing swiftly.

4.2 | Pre-Processing Procedure

For all experiments, the min-max normalization technique was applied to the AQI feature in the weather dataset. Assume that x_{max} and x_{min} are the maximum and minimum values of AQI in the weather dataset, and assume x_{high} and x_{low} are the highest and lowest values of the normalized time series, respectively. Therefore, the time series is normalized using Equation 7:

$$\bar{x} = \frac{x - x_{min}}{x_{max} - x_{min}}(x_{high} - x_{low}) + x_{low} \tag{7}$$

whereby $\bar{x}$ refers to the normalized data, and x to the original data.

Furthermore, 80% of our weather time-series data is used for training and the remaining 20% is used for testing.

4.3 | Evaluation Measures

As previously mentioned, RMSE and MAE are being measured.

The RMSE can be calculated using Equation 8:

$$\mathrm{RMSE}(y, \hat{y}) = \sqrt{\frac{\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2}{N}} \tag{8}$$

The MAE can be calculated using Equation 9:

$$\mathrm{MAE}(y, \hat{y}) = \frac{\sum_{i=0}^{N-1}|y_i - \hat{y}_i|}{N} \tag{9}$$

4.4 | Experiments and Results

A total of ten different experiments were done for evaluating all the described models.

4.4.1 | Experiment 1

For the LSTM model, the following parameters were selected for the minimal loss:

- Input layer = 4
- Optimization algorithm = Adam
- Number of epochs = 300

The loss of the model for training and validation data is reduced from 0.08 to 0.005 as shown in Figure 6. Figure 7 shows the training results of the time series analysis using the LSTM model in comparison to the original time series of the AQI values.

4.4.2 | Experiment 2

For the ARIMA model, the following parameters were selected:

- Order of auto-regressive model (p) = 1
- Order of moving average model (d) = 1

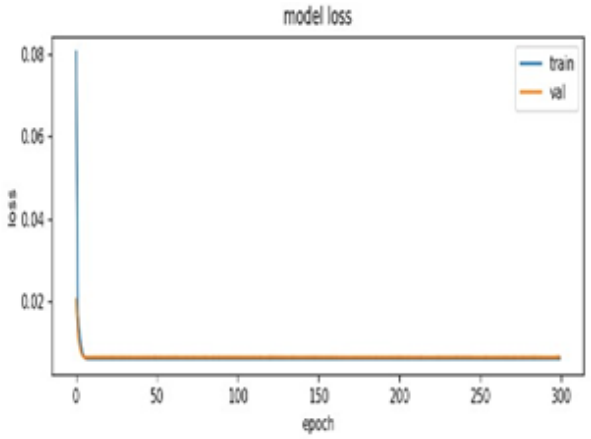

FIGURE 6 LSTM Training and Validation Loss

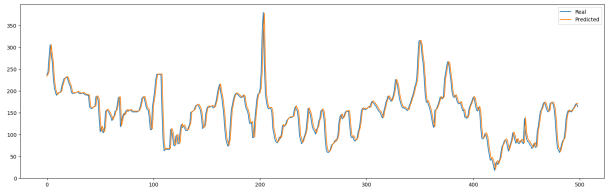

FIGURE 7 LSTM Actual vs. Forecasted Values

	coef	std err	z	P>|z|	[0.025	0.975]
const	171.5649	3.582	47.901	0.000	164.545	178.585
ar.L1.AQI	0.9734	0.001	952.532	0.000	0.971	0.975
ma.L1.AQI	0.4812	0.003	145.126	0.000	0.475	0.488

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.0273	+0.0000j	1.0273	0.0000
MA.1	-2.0780	+0.0000j	2.0780	0.5000

FIGURE 8 ARIMA Model Results for AQI

- Order of integrated model (q) = 1
- Optimization algorithm = Adam

Figure 8 shows the coefficients of the ARIMA model. The auto-regressive and moving average terms both have a p-value lower than 0.05, and thus, we have a good ARIMA model. Figure 9 shows the training results of the time series analysis using the ARIMA model in comparison to the original time series of the AQI values.

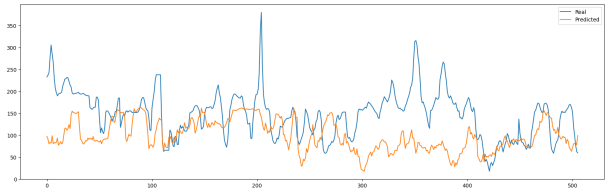

FIGURE 9 ARIMA Actual vs. Forecasted Values

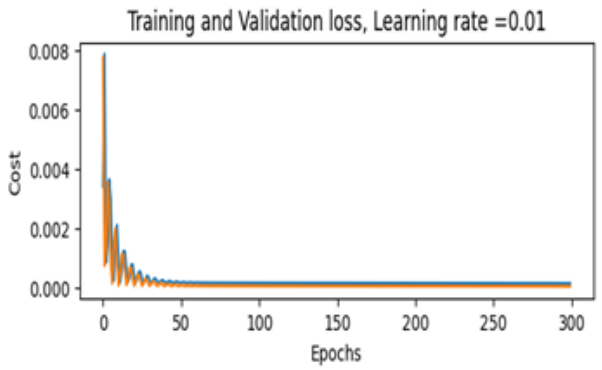

FIGURE 10 ANFIS Training and Validation Error

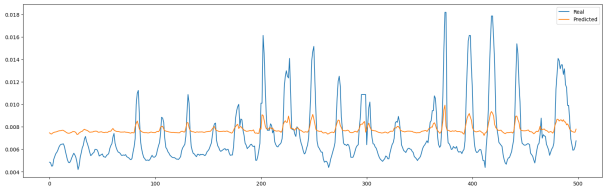

FIGURE 11 ANFIS Actual vs. Forecasted Values

4.4.3 | Experiment 3

The Python programming language is used in this experiment to evaluate the ANFIS model and the parameters are adjusted in the process to minimize the model loss. The parameters that are evaluated for this experiment are the number of nodes, number of fuzzy rules, number of epochs, loss function, and optimization method.

The selected parameters for the ANFIS model are:

- Learning rate = 0.01
- Number of fuzzy rules = 6
- Number of epochs = 300
- ANFIS Optimization method = Hybrid
- FIS type = Grid partition
- INPUT MF Type = gaussmf
- OUTPUT = Linear

The Hybrid ANFIS optimization method is used in this experiment. This method is used by both the ANFIS and Neuro-Fuzzy designer for updating the membership function parameters. The Hybrid method uses the backpropagation method and it is associated with the input membership functions and the least square estimation for the parameters associated with the output membership functions. For the ANFIS hybrid model, the training and validation loss reduced from 0.008 to almost 0 in 300 epochs as shown in Figure 10. Figure 11 shows the training results of the time series analysis in comparison to the original time series of the AQI values using the ANFIS model.

4.4.4 | Experiment 4

The CNN model for forecasting the Air Quality Index is one-dimensional as we are using only univariate data for this experiment. The model has filters connected to the input, which generates feature maps, and between the convolution layers there will be

pooling layers. This experiment consists of 64 filter maps with a kernel size of 2. After that, the model has a max pooling layer and a dense layer to extract the input data. The output only generates a single value, which is the predicted output.

The selected parameters for the CNN models were the following:

- Convolution layers = 2 (filters = 64 and kernel size = 2)
- Pooling layers = 2
- Fully connected layer = 1
- Sequence of layer = C-P-C-P-F
- Subsampling = max pooling
- Activation Function = ReLu
- Epochs = 300

Figure 12 shows the predicted values and original input values for the window size of the following 27 values using the CNN architecture.

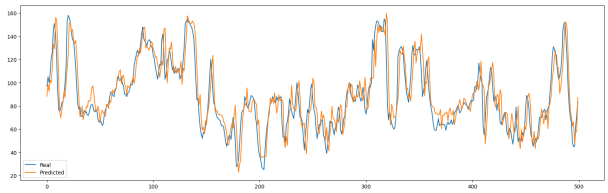

FIGURE 12 CNN Test Data - Actual vs. Forecasted Values

4.4.5 | Experiment 5

First the original data is modeled using Prophet . Prophet can be fitted easily after pre-processing using the available Python packages. This model is also easiest to apply as Prophet will detect the seasonality without any extra work.

Figure 13 shows the predicted results of the time series analysis in comparison to the original time series of the AQI values using the Prophet model. Most of the predicted values fall within the range of the original values but not nearly as close to the original values.

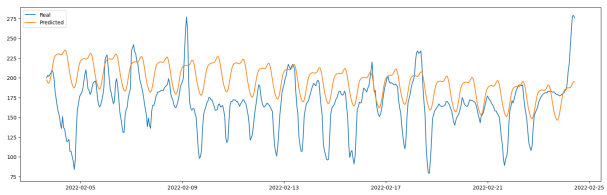

FIGURE 13 Prophet Test Data - Actual vs. Forecasted Values

4.4.6 | Experiment 6

For the Auto-regressive LSTM model, we feed input to the LSTM and the output predicted values will feed back to the input using the auto-regression layer. The model requires a warm-up process to set the input state based on the values. After training, this state will extract the relevant data from the input, which is a single step process, and thus, returns both the LSTM state and the single time step prediction. Now, after generating the single time step it will feed back as input at each step and we use the

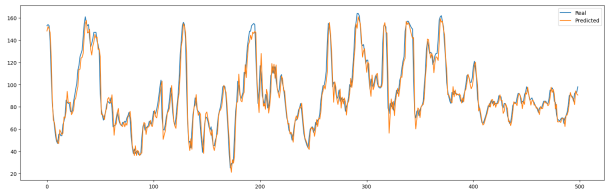

FIGURE 14 Predicted Values for Window Size = 24

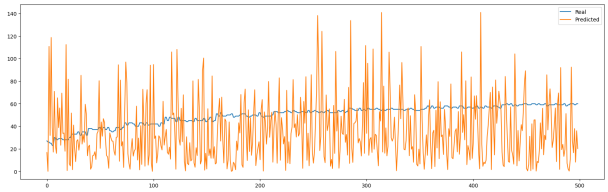

FIGURE 15 Autoencoder LSTM Test Data - Actual vs. Forecasted Values

Tensorflow stack to store the output after it passed through the loop. We set the window size to 24 so it will predict the next 24 values for the time series.

Figure 14 shows the prediction values and original input values for the window size of next 24 values using the Autoregressive LSTM model. The predicted values are close to the original values.

4.4.7 | Experiment 7

The selected parameters for the Autoencoder LSTM Autoencoder (comprises of LSTM Encoder and LSTM Decoder) and the Forecast model were the following:

- Units = 128
- Dropout = 0.5
- Optimizer = Adam
- Loss value = MAE and RMSE
- Epochs = 300
- Batch Size = 128

Figure 15 shows the test results of the time series analysis in comparison to the original time series of the AQI values using the Autoencoder LSTM model.

4.4.8 | Experiment 8

The Evolving Clustering Method (ECM) is a maximum distance clustering method used for partitioning the input space and creating the fuzzy inference rules (FIS). For this experiment, we used a type of ECM called online ECM, which is a one-pass algorithm for the dynamic estimation of the number of clusters to find the centers in the input data space. We implemented this ECM using python[20]. We use DENFIS as a fuzzy inference engine on top of ECM, and DENFIS works on the basis of Takagi–Sugeno type fuzzy inference engine. The fuzzy rules are created and updated at the same time with the online ECM input space partitioning.

The above DENFIS setup is used for modeling the data and predicting the future values, and the following experiment was conducted: The training data is used for online processes, and the testing for recalling procedure. We calculated RMSE and MAE for the model as a performance measurement.

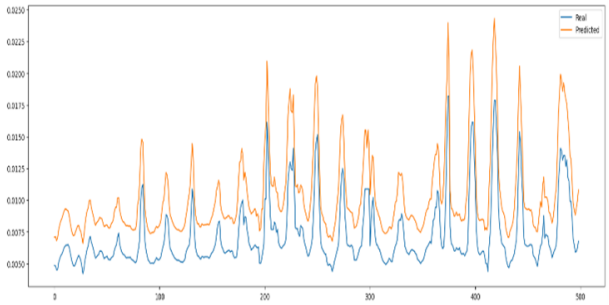

FIGURE 16 ANFIS+ABC Actual vs. Forecasted Values for Sample of 500 Values

4.4.9 | Experiment 9

In the first step for the ANFIS+ABC experiment, the normalized dataset's training portion is used to build the model, and then the testing set is used for the performance evaluation using the performance indices like RMSE and MAE values. The number of employed bees is initialized to 50, and the number of iterations is fixed to 500.

To predict the AQI values of the weather data using ANFIS, two fuzzy input variables are applied – NowCast concentration and raw concentration values of the $PM_{2.5}$ parameter. The bell membership function is used which is defined as in Equation 10:

$$\frac{1}{1+|\frac{x-c}{a}|^{2b}} \tag{10}$$

where x is a one-dimensional array of values; a, b and c are used to control width, center and slope, respectively.

At the beginning of the experiment, the ANFIS and ABC parameters are initialized before starting. The employed bees are fixed to 50 and the maximum iterations on the data is set to 500. To avoid the overfitting problem, we employed an early stopping strategy, such that the training process stops when the loss value on the validation data is not changed for several epochs.

Figure 16 shows the training results of the time series analysis in comparison to the original time series of the AQI values using the ANFIS+ABC model.

4.4.10 | Experiment 10

The transfer function type of high order fuzzy cognitive maps controls the output value range. The Tanh function is adopted as the transfer function because the wavelet coefficients can be negative values, and thus, our predicted value should be in $[-1, 1]$. Normalizing the time series data into the range of $[-1, 1]$ is necessary before feeding it to the Wavelet-HFCM for modeling and prediction. As we did in the previous experiment (ANFIS+ABC), we adopt the min-max normalization to normalize the data. After normalization, we split the data into training, validation, and testing datasets. The training dataset is used to learn the weight matrix of the Wavelet-HFCM prediction model, the validation data set is used to select the best model, and the testing data set is used for forecasting accuracy evaluation.

Wavelet-HFCM's forecasting performance depends on 3 hyperparameters - order k, number of nodes Nc, and regularization factor α. The default values of order 2 and 5 nodes work for most of the cases, but we use grid search as the validation for performing the model selection[30].

The following process is used to gather the optimal hyperparameters.

- Step 1: Let the order k be 1; find the best α by increasing the value of Nc from 2 to 7. Then, that best α will have both the validation dataset's highest prediction accuracy and also test dataset's record RMSE.
- Step 2: Increase order k by 1; then, go to Step 1 until k is 6.

The total HFCM nodes are equal to the wavelet decomposition level plus one. The HFCM parameter optimization is done using the proposed learning method that is based on ridge regression. Then, the learned HFCM is used for each wavelet coefficient scale's forecasting.

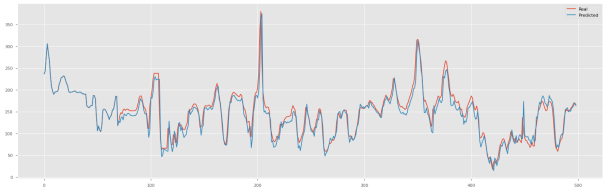

FIGURE 17 Wavelet-HFCM Actual vs. Forecasted AQI Values of Weather Data

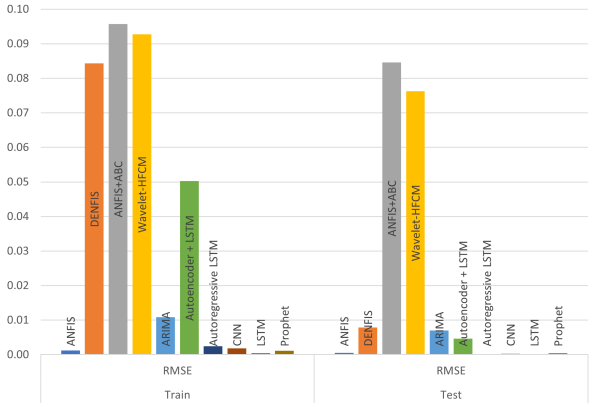

FIGURE 18 RMSE Comparison for all Experiments

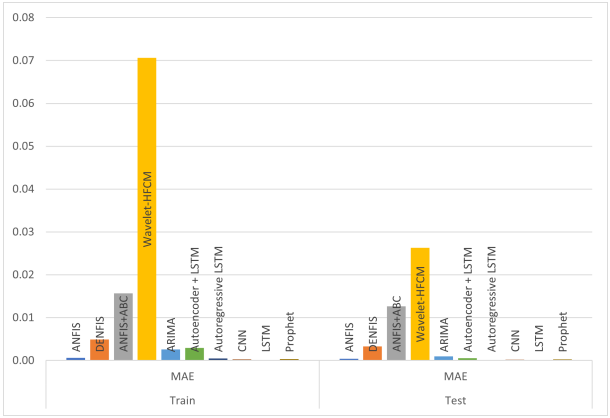

FIGURE 19 MAE Comparison for all Experiments

The wavelet coefficients predictions at each scale and at each point is summed to determine the final AQI prediction value of the dataset.

Figure 17 shows the training results of the time series analysis in comparison to the original time series using the Wavelet-HFCM model.

4.4.11 | Summary of Results

Figures 18 and 19 show the charts for RMSE and MAE, respectively, for all the models used for this research investigation.

TABLE 1 RMSE and MAE on train and test portion of forecasting models

	Train		Test	
	RMSE	**MAE**	**RMSE**	**MAE**
ANFIS	0.001263	0.000620	0.000522	0.000375
DENFIS	0.084306	0.004907	0.007899	0.003259
ANFIS+ABC	0.095738	0.015652	0.084634	0.012614
Wavelet-HFCM	0.092765	0.070620	0.076305	0.026330
ARIMA	0.01083	0.002611	0.006978	0.000982
Autoencoder+LSTM	0.050231	0.002967	0.004673	0.000565
Autoregressive LSTM	0.002467	0.000459	0.000198	0.000084
CNN	0.001842	0.000238	0.000295	0.000166
LSTM	0.00046	0.000035	0.000024	0.000002
Prophet	0.001097	0.00036	0.000439	0.000238

TABLE 2 Results obtained by Friedman Test

Observation value	χ^2	**Asymptotic P-value**
RMSE	16.9090	0.05016
MAE	17.5636	0.040596

Table 1 shows the comparison of all the models investigated in this paper for both the train and test portions of the data set. RMSE and MAE are used to evaluate the accuracy of the models.

4.4.12 | Relative Performance of Experiments

The Friedman test is a non-parametric statistical test that used to identify the differences in experiments across multiple test attempts[45]. We applied the Friedman's test to all the 10 experiments in this paper across the train dataset and test dataset. Test data and train data are considered as blocks for the purpose of this statistical analysis, all the 10 experiments are considered as the treatments. We applied the Friedman's statistical approach by assuming RMSE as observation value in one approach and in another approach MAE as observation value.

In this analysis the null and alternative hypothesis are formed.

H_0: There is no statistical difference between the performances of the experiments

H_a: There is a statistical difference between the performances of the experiments

The Friedman's test first calculates the relative ranks of all the experiments based on the RMSE and MAE values of test and train data. This ranking process is used to calculate the Friedman statistic value χ^2, and this statistic value further used to calculate the p value.

The Friedman statistic value is calculated by using Equation 11[46]:

$$\chi^2 = \frac{12}{nk(k+1)} [\sum_{i=1}^{k} R_i^2] - 3n(k+1) \tag{11}$$

where R_i is the individual rank total of the treatment, n is the number of data instances, and K is number of treatments. The Friedman statistic value χ^2 is distributed over 'k-1' degrees of freedom. The results for the Friedman's test are tabulated in Table 2.

For the Friedman test, where RMSE is considered as observation value, the calculated value of χ^2 is less than the tabulated value (obtained from chi-squared table with 0.05 level of significance and degree of freedom), we do not reject the null hypothesis. Thus, we conclude that the performance of ten experiments do not differ significantly.

For the Friedman test, where MAE considered as observation value, the calculated value of χ^2 is greater than the tabulated value, we reject the null hypothesis. Thus, we conclude that the performance of ten experiments are statistically significantly different.

5 | CONCLUSION

Predicting the air quality is difficult because the contaminants are variable and highly volatile. The accurate prediction of air quality becoming crucial as the air pollution impacts the environment and health of the living organisms particularly in the urban areas. The dataset for this paper is extracted from the Airnow website maintained by the Office of the Air Quality Planning and Standards (OAQPS), USA to monitor the pollution all over the world where the USA consulates reside. For this paper, experiments we used the air pollution data for Delhi, a city in India which is currently ranked as one of the highly polluted places in the world. The data gathered has undergone pre-processing to add missing values and eliminate any extraneous information.

Forecasting the time series is a complex issue due to its sequence characteristics. There are many different algorithms that contribute to the prediction. Good prediction is defined as a minimum of the cost or higher accuracy. We investigated various deep learning and fuzzy based time series forecasting methods to forecast the AQI values in the dataset. As shown by the results of the experiments, the LSTM technique outperformed by a great margin with RMSE values close to zero - 0.00046 and 0.000024 for train and test data, respectively, compared to all the other models.

The Wavelet-HFCM error results are higher compared to the other techniques, and therefore, it is the weakest model for this time series data set but it might work for other types of data sets. Hence, we can conclude that the LSTM model is the best technique to use when better accuracy is concerned.

REFERENCES

1. D. Boudreau, M. McDaniel, E. Sprout and A. Turgeon, "pollution," National Geographic Society, 15 12 2022. [Online]. Available: https://education.nationalgeographic.org/ resource/pollution/. [Accessed 9 1 2023].
2. S. Singh and C. Srivastava, "Estimation of Air Pollution in Delhi Using Machine Learning Techniques," International Conference on Computing, Power and Communication Technologies, 9 2018.
3. World's most polluted cities, IQAir, [Online]. Available: https://www.iqair.com/in-en/world-most-polluted-cities. [Accessed 9 1 2023]
4. C. Textor, "China Annual PM2.5 Particle Levels Beijing," Statista, 12 1 2023. [Online]. Available: https://www.statista.com/statistics/690823/china-annual-pm25-particle-levels-beijing/. [Accessed 13 2 2023].
5. H. Qin and M. Whitney, "How China is tackling air pollution with big data," World Economic Forum, 21 2 2021. [Online]. Available: https://www.weforum.org/agenda/2021/02/ china-tackling-air-pollution-big-data/. [Accessed 24 12 2022].
6. V. Thomas and C. Tiwari, "Delhi, the world's most air polluted capital fights back," BROOKINGS, 25 11 2020. [Online]. Available: https://www.brookings.edu/blog/future-development/2020/11/25/delhi-the-worlds-most-air-polluted-capital-fights-back/. [Accessed 2022 29 12].
7. AirNow, [Online]. Available: https://www.airnow.gov/international/us-embassies-and-consulates/India, New Delhi. [Accessed 25 4 2023].
8. C. Srivastava, S. Singh and A. P. Singh, "Estimation of Air Pollution in Delhi Using Machine Learning Techniques," in International Conference on Computing, Power and Communication Technologies, Greater Noida, 2018.
9. M. S. Baawain and A. S. Al-Serihi, "Systematic Approach for the Prediction of Ground-Level Air Pollution (around an Industrial Port) Using an Artificial Neural Network," Aerosol and Air Quality Research, vol. 14, no. 1, pp. 124-134, 2014.
10. S. Roy, S. P. Biswas, S. Mahata and R. Bose, "Time Series Forecasting using Exponential Smoothing to Predict the Major Atmospheric Pollutants," in International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, 2018.
11. W. Wang, W.-g. Shen, B. Chen, R. Zhu and Y.-x. Sun, "Air Quality Index Forecasting Based on SVM and Moments," in International Conference on Systems and Informatics, Nanjing, 2018.
12. A. Barve, V. Mohan Singh, S. Shrirao and M. Bedekar, "Air Quality Index forecasting using parallel Dense Neural Network and LSTM cell," in International Conference for Emerging Technology, Belgaum, 2020.
13. L. Yan, Y. Wu, L. Yan and M. Zhou, "Encoder-Decoder Model for Forecast of PM2.5 Concentration per Hour," in International Cognitive Cities Conference, Okinawa, 2018.
14. Y. Zhu and X. Zhou, "Prediction of Air Quality Index Based on Wavelet Transform Combination Model," in International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, 2019.
15. U. Kiftiyani and S. A. Nazhifah, "Deep Learning Models for Air Pollution Forecasting in Seoul South Korea," in International Conference on Software Engineering & Computer Systems and International Conference on Computational Science and Information Management, Pekan, 2021.
16. S. F. Mihalache, M. Popescu and M. Oprea, "Particulate matter prediction using ANFIS modelling techniques," in International Conference on System Theory, Control and Computing, Cheile Gradistei, 2015.
17. S. F. Mihalache and M. Popescu, "Development of ANFIS models for PM short-term prediction. case study," in International Conference on Electronics, Computers and Artificial Intelligence, Ploiesti, 2016.
18. M. Parsajoo, A. S. Mohammed, S. Yagiz, D. J. Armaghani and M. Khandelwal, "An evolutionary adaptive neuro-fuzzy inference system for estimating field penetration index of tunnel boring machine in rock mass," Journal of Rock Mechanics and Geotechnical Engineering, vol. 13, no. 6, pp. 1290-1299, 2021.
19. N. Kasabov and Q. Song, "DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction," IEEE Transactions on Fuzzy Systems, vol. 10, no. 2, pp. 144-154, 2002.
20. S. Yang and J. Liu, "Time-Series Forecasting Based on High-Order Fuzzy Cognitive Maps and Wavelet Transform," IEEE Transactions on Fuzzy Systems, vol. 26, no. 6, pp. 3391-3402, 2018.
21. G. Singhal, "Introduction to LSTM Units in RNN," Pluralsight, 20 9 2020. [Online]. Available: https://www.pluralsight.com/guides/ introduction-to-lstm-units-in-rnn. [Accessed 12 12 2022].
22. R. Nau, "Introduction to ARIMA: nonseasonal models," 18 8 2020. [Online]. Available: https://people.duke.edu/ rnau/411arim.htm. [Accessed 16 1 2023].

23. C. K. Babulal and P. S. Kannan, "A Novel Approach for ATC Computation in Deregulated Environment," Journal of Electrical Systems, vol. 2, no. 3, p. 146 – 161, 2006.
24. M. S. Kıran and O. Fındık, "A directed artificial bee colony algorithm," Applied Soft Computing, vol. 26, pp. 454-462, 2015.
25. D. Karaboga and B. Basturk, "Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems," in International Fuzzy Systems Association World Congress, 2007.
26. D. Cai, Y. Yu and J. Wei, "A Modified Artificial Bee Colony Algorithm for Parameter Estimation of Fractional-Order Nonlinear Systems," IEEE Access, vol. 6, pp. 48600-48610, 2018.
27. R. Ye, P. N. Suganthan, N. Srikanth and S. Sarkar, "A hybrid ARIMA-DENFIS method for wind speed forecasting," in IEEE International Conference on Fuzzy Systems, Hyderabad, 2013.
28. F. Z. Xing, E. Cambria and X. Zou, "Predicting evolving chaotic time series with fuzzy neural networks," in International Joint Conference on Neural Networks, Anchorage, 2017.
29. B. Kosko, "Fuzzy Cognitive Maps," International Journal of Man-Machine Studies, vol. 24, pp. 65-75, 1986.
30. M. Rhif, A. A. Ben, I. R. Farah, B. Martínez and Y. Sang, "Wavelet Transform Application for/in Non-Stationary Time-Series Analysis: A Review," Applied Sciences, vol. 9, no. 7, 2019.
31. I. Koprinska, D. Wu and Z. Wang, "Convolutional Neural Networks for Energy Time Series Forecasting," in International Joint Conference on Neural Networks, Rio de Janeiro, 2018.
32. J. Brownlee, "How to Develop Convolutional Neural Network Models for Time Series Forecasting," Machine Learning Mastery, 12 11 2018. [Online]. Available: https://machinelearningmastery.com/how-to-develop-convolutional-neural-network-models-for-time-series-forecasting/. [Accessed 12 12 2022].
33. I. Yenidoğan, A. Çayir, O. Kozan, T. Dağ and Ç. Arslan, "Bitcoin Forecasting Using ARIMA and PROPHET," in International Conference on Computer Science and Engineering, Sarajevo, 2018.
34. J. T. Rasmussen, "How to Use an Autoregressive (AR) Model For Time Series Analysis," Towards Data Science, 22 12 2021. [Online]. Available: https://towardsdatascience.com/how-to-use-an-autoregressive-ar-model-for-time-series-analysis-bb12b7831024. [Accessed 28 1 2023].
35. L. Zhu and N. Laptev, "Deep and Confident Prediction for Time Series at Uber," in IEEE International Conference on Data Mining Workshops, 2017.
36. M. Cerliani, "Extreme Event Forecasting with LSTM Autoencoders," Towards Datascience, 22 5 2019. [Online]. Available: https://towardsdatascience.com/extreme-event-forecasting-with-lstm-autoencoders-297492485037. [Accessed 26 1 2023].
37. T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning Data Mining, Inference, and Prediction, New York, Springer, 2008, pp. 219-257.
38. What-is-time-series-data, influxdata, [Online]. Available: https://www.influxdata.com/what-is-time-series-data/. [Accessed 23 1 2023].
39. J. Yang and A. W. Ismail, "Air Quality Forecasting Using Deep Learning and Transfer Learning: A survey," in Global Conference on Computing, Power and Communication Technologies, New Delhi, 2022.
40. Forecasting algorithms, IBM, 21 6 2023. [Online]. Available: https://www.ibm.com/docs/en/planning-analytics/2.0.0?topic=models-forecasting-algorithms. [Accessed 22 6 2023].
41. R. J. Kuo, B. Prasetyo and B. S. Wibowo, "Deep Learning-Based Approach for Air Quality Forecasting by Using Recurrent Neural Network with Gaussian Process in Taiwan," in International Conference on Industrial Engineering and Applications, Tokyo, 2019.
42. J. Tan, W. Xiong and Z. Tu, "The Air Quality Prediction on Deep Spatiotemporal Feature Extraction with a Transductive Kernel Extreme Learning Machine," in International Symposium on Industrial Electronics, Anchorage, 2022.
43. N. Jin, Y. Zeng, K. Yan and Z. Ji, "Multivariate Air Quality Forecasting With Nested Long Short Term Memory Neural Network," in IEEE Transactions on Industrial Informatics, vol. 17, no. 12, pp. 8514-8522, Dec. 2021.
44. J. -C. Cheng and H. -C. Peng, "Air Quality Forecast and Evaluation Based on Long Short-Term Memory Network and Fuzzy Algorithm," 2021 IEEE 4th International Conference on Knowledge Innovation and Invention (ICKII), Taichung, Taiwan, 2021.
45. "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Friedmantest. [Accessed 5 8 2023].
46. R. Malhotra, Empirical Research in Software Engineering, Routledge, 2015.