Vol. 02, No. 4 (2020) 431-440, doi: 10.24874/PES0204.010

Proceedings on Engineering Sciences

www.pesjournal.net

AIR QUALITY INDEX FORECASTING USING HYBRID NEURAL NETWORK MODEL WITH LSTM ON AQI SEQUENCES

Shirshendu Roy[1]
Pratyay Mukherjee

Keywords:

LSTM; Hybrid model; AQI, forecast; Time series.

A B S T R A C T

This paper presents an approach to forecasting air pollution levels measured as Air Quality Index (AQI) metric using hybrid Long Short-Term Memory (LSTM) models. The pollution levels have been found to vary in a particular pattern that depends on both the overall climate or season as well as the hour of the day. The hybrid model captures these 2 patterns and makes the prediction of AQI of some future hour. It employs 2 separate LSTM models that are trained on time-series data of AQI gathered at different time lags i.e. hourly and daily. The final output is given as a weighted sum of the 2 outputs produced by LSTM model. Upon comparing the performance of the standalone hour-wise forecasting LSTM model and the hybrid model it was found the latter gives the minimum error metric given an appropriate weight is chosen.

1. INTRODUCTION

A detailed and comprehensive overview of the cause of air pollution and its adverse effects on the environment especially human health is found in (Bai et al., 2018). The paper asserts the importance of research in preventing air pollution using artificial intelligence. It also shows that statistical and hybrid models have the potential to address the problem by forecasting the Air Quality Index (AQI) accurately. This paper proposes a hybrid model to identify the air quality index accurately.
Central Pollution Control Board (CPCB) and Open Government Data (OGD) provide real-time as well as historical data on air pollutants level in major cities and towns of India. This model proposes to use that data to build reliable deep learning models. The model presented in this paper has been trained using the data gathered from CPCB. The terrible situation in Indian cities (Greenstone et al., 2017; Garg et al., 2018; Mishra, 2019)due to increase in the level of air pollution makes it a good platform on which to build and test the hybrid model. CPCB provides the standards and procedures to calculate the AQI along with the consequential health hazards and their respective precautionary steps shown in Table 1. Monitoring stations have been set up across the country in major cities, towns and industrial areas where air pollution is concentrated.

LSTM (Hochreiter and Schmidhuber, 1997), the abbreviation of Long Short-Term Memory is a variant of Recurrent Neural Network (RNN) that has found an increasing number of applications in the field of forecasting problems using time-series data. It can be considered as the state-of-art deep learning algorithm that can easily identify long-term dependencies in sequential data. This feature of LSTM can be put to advantage to

[1] Corresponding author: Shirshendu Roy
Email: mtq0306@yahoo.com

make predictions of air pollution levels or AQI by appropriately using its pattern of variation.

Hybrid neural network models consist of a number of artificial neural network models (Agarwal, 2018, pp. 1-52)that may be of the same or different types that are trained independently and influence the output independently. In (Miškovic, 2014)we can find a comparative study of explicit, implicit and hybrid machine learning models in decision-making problems in the field of business, medicine, diagnosis, etc. that shows hybrid models perform uniformly well and sometimes better than the traditional models in the mentioned fields of application. A good example of the use of a hybrid model in the deep learning paradigm can be found in (Banihashemi, Ding and Wang, 2017) , where an Artificial Neural Network for prediction and a decision-tree for classification have been employed to predict building energy consumption with minimum error. Hence, hybrid models have been found to outperform traditional standalone machine learning and deep learning models in most of the cases by a significant margin.

Table 1. AQI Levels and impact described by CPCB, India (“National Air Quality Index”, 2020)

AQI Category	Associated Health Impact
Good (<50)	Minimal impact
Satisfactory (51-100)	Minor breathing discomfort in sensitive people
Moderately polluted (101-200)	Breathing discomfort to people with lung disease such as asthma and discomfort to people with heart disease, children and older adults.
Poor (201-300)	Breathing discomfort to people on prolonged exposure and discomfort to people with heart disease.
Very Poor (301-400)	Respiratory illness to the people on prolonged exposure. People with lung and heart diseases may suffer severely.
Severe (≥401)	Respiratory illness even on healthy people and serious health impacts on people with lung/heart diseases. The health impacts may be experienced even during light physical activity.

To utilize the ability of LSTM to capture long-term dependencies in data and that of hybrid models to combine strengths of knowledge representation of both models we present a hybrid model that consists of 2 LSTM models that are used to train on time-series data of AQI recorded at different time lags, hour-wise and daily. The final hour-wise prediction of a future hour is a weighted sum of the outputs obtained from 2 models.

On comparison of the RMSE (Root Mean Square Error)on training and test data of the standalone LSTM model trained on hour-wise data and the hybrid model, it has been found that the latter approach gives a significantly lower error metric. This result can be credited to the ability of the hybrid model to capture the variations in AQI on a seasonal (monthly changes) and daily(hourly changes) basis.

2. RELATED WORK

The hybrid model proposed in this paper is largely influenced by the one described in (Le and Cha, 2018). Although the given model closely resembles the proposed one in terms of the field of application, the main difference is in the composition of the hybrid model and most importantly the objective. The objective of the mentioned paper is to use spatiotemporal big data for predicting the real-time air pollution levels. To do that a hybrid model consisting of a neural network with a LSTM layer and a simple multilayer artificial neural network has been employed. The LSTM model has been trained using unit-step time-series data to predict real-time air pollution based on the hour-wise variational patterns that can be observed in pollution levels. Artificial neural networks (Agarwal, 2018, pp. 1-52) are trained computational models that are meant to simulate the learning mechanism of human brain. In the real-time prediction model, a simple artificial neural network has been used to predict the air pollution levels based on various factors that influence the pollution level such as weather, wind direction, wind speed, etc.

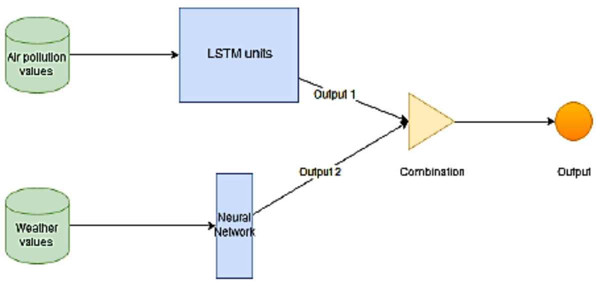


Figure 1. A hybrid model consisting of LSTM and artificial neural network and the corresponding data used to train each model (Le and Cha, 2018, Fig. 6)

The weighted sum used to compute the final output is

$$O = \alpha L + (1 - \alpha)N \quad (1)$$

where O is the final output, α is the weight used to combine the two outputs, L is the output of the LSTM and N is the output of the neural network.
This model gives a real-time prediction of the AQI levels. However, the objective of the proposed model is to use a hybrid model to forecast AQI. The performance of the hybrid model on real-time prediction in comparison to those of standalone neural networks or LSTM model is considerably better. A comparative study of the validation error between the hybrid model and a standalone model is shown in Table 2 for the prediction problem that justifies the choice of using a hybrid model for the forecasting problem.

Table 2. Comparative study of hybrid and standalone neural network models. (Le and Cha, 2018, TABLE II)

α	Standalone RNN model	Standalone LSTM model	RNN + NN model	Hybrid model (LSTM + NN)
1.0	**5.010952**	5.775150	**5.010952**	5.775150
0.9			5.207657	**4.656111**
0.8			**4.574787**	4.880526
0.7			6.302503	**4.895847**
0.6			4.946835	**4.705889**
0.5			4.834772	**4.257993**
0.4			**4.265581**	5.341873
0.3			**6.811509**	7.887727
0.2			5.673909	**4.966643**
0.1			5.198360	**4.546102**
0.0			6.597609	**6.476847**

3. LITERATURE REVIEW

3.1Time series data

Time-series data is sampled at discrete time points with a uniform time-interval. In other words, data that is ordered chronologically can be regarded as time-series data. The analysis of such data involves examining and determining the dependency of some variables on changing another variable over time. As defined in (Ullah, 2020), a sequence of random variables indexed by time is called a stochastic process (stochastic means random) or time series.

We can think of a time-series variable as a unit of time denoted by Y_ti.e. value of variable Yat time $\boldsymbol{t}$. The values observed prior to $t$are called lag variables. The j^{th}lag variable in Y_{t-j}. A simple bivariate regression equation on time-series data can be given by (Ullah, 2020):

$$y_t = \beta_0 + \beta * x_t + u_t (2)$$

Time-series data can be of 2 types. Continuous time-series data is sampled continuously while discrete time series data is sampled at discrete timestamps.

Time series analysis (Kantz & Schreiber, 2014) is the extraction of characteristics and meaningful statistics (Warren Liao, 2005) from the data. On the other hand, time series forecasting is done using a model to forecast future values of a variable based on the observations at previously sampled timestamps. There are real-world scenarios where time-series analysis and forecasting can play a major role. For example, in (Lin et al., 2003) time-series data (real-values) is used to represent streaming data (discrete-valued) for efficient data mining and analysis purposes.

3.2Recurrent Neural Networks

RNN (Olah, 2015) is a variant of the traditional feed forward neural networks that exhibit dynamic behavior with respect to time. This helps the network to capture the context of the data for performing various operations such as analysis, regression, classification and forecasting.

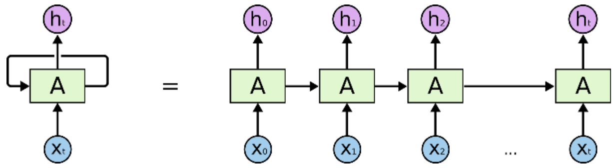


Figure 2. An RNN with loops and its equivalent unrolled structure. The sequential input and output with the transfer of ‘information’ from one state to another depicts the capability of RNNs to model context of data (Olah, 2015, Recurrent Neural Networks)

In Fig. 2, RNN is shown as a chunk of neural network (**A**) that takes an input ($\mathbf{x_t}$) and gives an output($\mathbf{h_t}$), by using both the current input and information propagated through previous time steps or context. Thus, intuitively RNNs can be regarded as a copy of the same network that gives an output upon giving some input and passes the contextual information to its successor.

The term ‘recurrent neural network’ is sometimes assumed to cover two broad types of networks that have the same general structure and behavior – finite impulse and infinite impulse. The major difference between these 2 networks is that the former can be unrolled and replaced by a strictly feed forward neural network which is not possible for the latter. A detailed comparative study of RNNs and FIRNN (Finite Impulse Response Neural Networks) can be found in (Miljanović, 2012). This comparative study, performed on time-series datasets namely, Mackay-Glass, Sunspots, S&P 500 shows that RNN outperforms its counterpart in most of the cases even when FIRNN hyper parameters are tuned to give optimal performance.

Due to its efficiency in modeling the dependence of sequential data, RNNs have been majorly used in fields such as handwriting recognition (Graves & Schmidhuber, 1995; Graves et al., 2009) and speech recognition (Li and Wu, 2015).

3.3 Long Short-Term Memory

As stated in (Bengio, Simard and Frasconi, 1994) the requirements for a model exhibiting temporal dynamic behavior are the ability to store information for an arbitrary duration, resiliency to noise in data and trainable parameters. Although RNNs meet these requirements in theory, in practice they find it hard to model long-term dependencies.

Long short-term memory network (Hochreiter & Schmidhuber, 1997) is a variant of the RNN model that has been specifically built to solve the long-term dependency problem. LSTM networks are increasingly being used in complex sequence and pattern analysis fields such as language modeling (Sundermeyer, Schluter & Ney, 2012) and large-scale acoustic modeling (Sak, Senior & Beaufays, 2014).

The basic difference in the structure between RNN and LSTM networks is in the repeating module seen in the unrolled view. LSTM networks make use of **gate** structures to add or remove information to the cell state. Fig. 3 shows a basic LSTM network. Each directional line carries a vector. A junction point of these lines denotes concatenation while a fork point denotes split of the vector into 2 different directions. The pink circles represent operation – vector addition, multiplication or tanh operation while the yellow rectangles are neural networks. The horizontal line passing through the top of the cell contains cell state information while the bottom horizontal line carries the hidden state and input information.

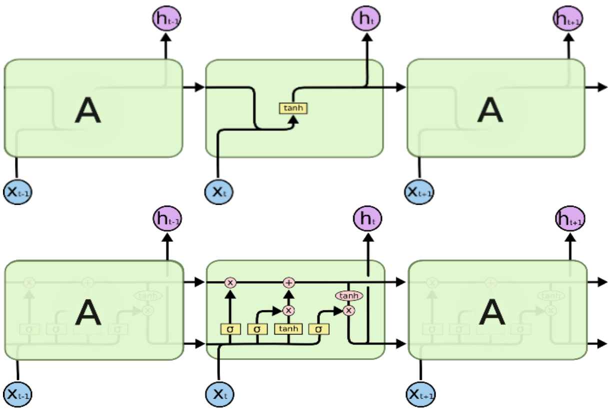


Figure 3. A simple RNN (top) contains a single layer while a LSTM network contains 4 intersecting layers. (Olah, 2015, LSTM Networks)

The mathematical operations carried out in each layer is as follows:

i. The leftmost sigmoid layer called **forget gate** decides what information regarding the cell state to keep.

$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$ (3)

where f is the output of the sigmoid network, h is the output of the previous layer, x is the input and b is the bias.

ii. A two-step process to decide what information needs to be added to the cell state. This is done by another sigmoid layer **called the input gate and** a tanh layer.

$i_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$ (4)

where i is the output of the input gate.

After the input gate computes the values that need to be updated the tanh layer creates a vector of new values to be introduced into the cell state.

$C_t' = tanh(W_f[h_{t-1}, x_t] + b_c)$(5)

where C refers to the cell state.

iii. Update the cell state by forgetting the amount of information we had computed using the forget gate i.e. $f_t$and adding the new values i.e. C_t' by the amount we had computed using the input gate i.e. i_t.

$C_t = f_t * C_{t-1} + i_t * C_t'$(6)

iv. Finally, we compute the output by filtering the cell state and keeping only the relevant and desired information. This is a two-step process. First, we use a sigmoid layer called **output gate** to decide which parts of the cell state we are going to produce as output and then multiply this vector with the cell state after passing it through a tanh layer.

$O_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$ (7)

where o is the output of the output gate.

$h_t = O_t * \tanh(C_t)$(8)

The basic LSTM structure and working have been upgraded further in recent years to enhance its capabilities make it viable to be used in everyday applications. A very concise and comprehensive study of these variants of the LSTM network can be found in (Greff et al., 2017).

3.4Hybrid Models

Many real-world problems have various perspectives or views. These views can be regarded as the different types of features or the domain of the features altogether. Sometimes these problems need to be viewed from all these perspectives to build an efficient model to perform various operations such as classification, regression, analysis, etc. Traditional machine learning models and neural networks are mostly capable of modeling these views partially which often leads to results that are below the desired standard. For example, in our problem of forecasting the AQI the air quality changes differently according to season and also the hour of the day. Hybrid models are an alternative for standalone models that can capture these perspectives efficiently and are being used increasingly in various real-world applications.

4. METHODOLOGY

4.1 Data Acquisition

The Central Pollution Control Board of India collects the data of air quality from stations posted at major cities and industrial areas in India. The model has been trained on the data collected from the Anand Vihar station in New Delhi. Two data sets are formed by collecting the data at different time lags, hour-wise and day-wise.

The process of gathering the data has been automated to query the calendar API of the website. Data has been collected hour-wise from 10th May 2019 to 4th July 2019 till 23:59:59. This gave a substantial amount of data consisting of 626 data points. The AQI of a single day has been calculated by taking the average over the 24 hours of that day. Thus, the data collected on a daily basis range from 14th August 2018 to 13th July 2019 and consists of 300 data points.

The time frame for the data collected hour-wise has been chosen in such a manner that it can capture the pattern in variation of the air pollution levels affected by factors that change frequently viz. wind direction, wind speed, traffic density, etc. Similarly, the collected daily data is expected to capture the variations in air pollution levels that are observable over a long period of time affected by factors such as climate, smog, humidity, etc.

4.2 The validity of LSTM and Preprocessing of Data

A good check to find the dependence of the value of a particular variable on the values it had acquired at previous time steps is to find the autocorrelation i.e. the correlation of the variable with itself. The autocorrelation plot also gives us a fair estimate of which lag variable(s) influences the value at a given timestamp significantly.
The autocorrelation plot provided by the 'Pandas' library in Python has been used for plotting the figures. In Fig.4 we see dashed and solid lines parallel to the x-axis which indicates the 95-99% confidence intervals. The portion of the plot lying above these lines is considered significant and thus the lag value can be inferred to be between 1 to 90 for the hour-wise AQI and between 1 to 40 for daily AQI. Hence, we take a vector of lag variables for each of the constituent LSTM models of the hybrid model – 12 for training the model on hour-wise data and 7 for the one on daily data.

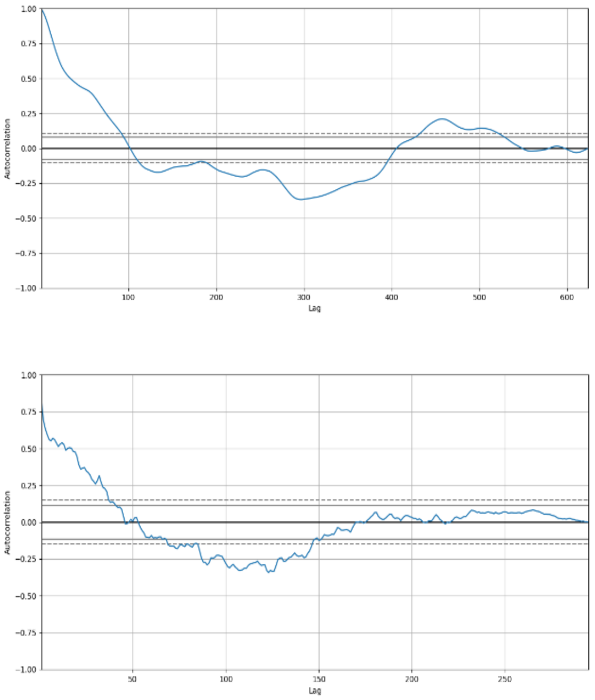


Figure 4. Autocorrelation plot (autocorrelation vs lag) for AQI collected hour-wise(top) and daily(bottom)

For training time-series models for regression, it is essential to normalize data. Normalized inputs and outputs ensure faster convergence of the learning algorithm and makes the learning process stable thereby preventing exploding gradients that render the learning process a failure. Since we are using only one variable i.e. AQI we can normalize the entire dataset together taking values ranging from 0 to 1. We also need to compute the reverse normalization so that we can get the true values within the actual range during the validation process and the actual forecasting application.

The LSTM architecture has been built using Keras which is a neural network API for Python running on top of TensorFlow. Hence, this also necessitates the conversion of our time-series data into the form of data used in supervised learning i.e. feature vector and the corresponding label. In this problem, the label will be the value of AQI at a given timestamp while the feature vector will consist of the lag variables. Thus, the general form of each data point in the dataset will appear as shown below.

$$[x_{t-k}, x_{t-k+1}, \dots . x_{t-1}] \rightarrow x_t \qquad (9)$$

In the above representation, the LHS represents the feature vector with k lag variables and t represents the value of the variable at a given timestamp.

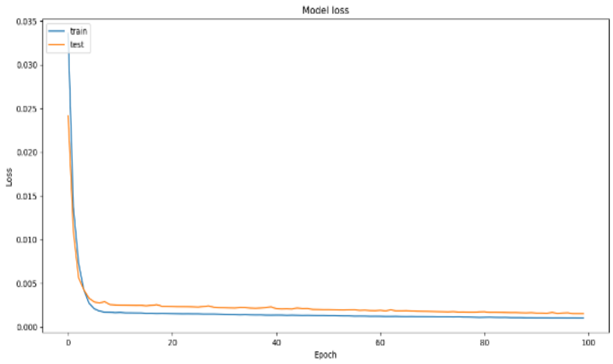
4.3Training and Validation of Neural Network models

The processed time-series data generated cannot be split randomly into the training and validation sets. There are 2 reasons.

- Time-series data is ordered with respect to time and randomized splitting of the dataset does not preserve this order. This may lead us to predict or forecast for some samples using a model trained on posterior samples.
- Time series data is often very highly correlated in time and hence a random validation split may include samples in it that are likely to be strongly correlated to the training set. Thus, the very purpose of a validation set is not met and the validation process may give alarmingly good metrics.

This is why the traditional approach of splitting a time series data set is to take contiguous blocks of training and validation set.Here we have taken the first three-fourth fraction of both the datasets as the training set and the rest as the validation set.

The LSTM layer configuration provided by the Keras API for Python has been used in the neural network models. The neural network architecture for hour-wise forecasting consists of an LSTM input layer of 12 nodes corresponding to AQI values of the previous 12 hours (lag variables) and 100 hidden units. The output layer has one node for a single output AQI value of the next hour. The layers are dense. The model has been trained using stochastic gradient descent as an optimizer with a learning rate of 0.01, the momentum of 0.5 and mean-squared error loss over 100 epochs with a batch size of 5. The neural network architecture for day-wise forecasting consists of an LSTM layer of 7 input nodes corresponding to AQI values of the previous week and 80 hidden units. The output layer has one node for a single output AQI value of the next day. The layers are fully connected. The model has been trained using stochastic gradient descent (Ruder, 2016) as an optimizer with a learning rate of 0.01, a momentum of 0.5, decay of 0.001 and mean squared error loss over 500 epochs with a batch size of 10. The values of the hyper parameters of the neural network and LSTM layer have been chosen empirically.



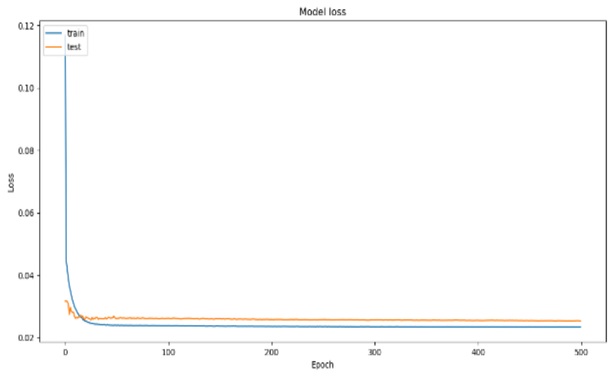


Figure 5. Loss plot (Loss vs epoch) of training of both models (hour-wise above, daily below). Convergence indicates good fit

The loss plot in Fig. 5 indicates a good fit. The final training loss of the model trained on hour-wise data after 100 epochs is 0.001 and that of the model trained on daily data after 500 epochs is 0.0234.

The validation score and loss or mean squared error (MSE) is calculated at each training epoch on the validation dataset previously separated. The MSE train and test scores of the former model are 13.43 and 16.57, respectively while that of the latter is 64.38 and 67.11, respectively.

4.4Estimating coefficients for combining forecasted values

To select the appropriate weights for combining the two AQI values in the datasets to produce the target value with minimum error we can employ curve-fitting (O'Neil et al., 2015; Arlinghaus, 1994) algorithm. The equation hence formed is

$$O = \alpha * A_1 + \beta * A_2 (10)$$

where α and β are the parameters to be estimated, A_1 and $A_2$are the average daily AQI and hour-wise AQI, respectively and O is the output or true value.

For the purpose of curve-fitting, the trained models are used to forecast values at timestamps for which the AQI values are already known i.e. the values can be obtained from the source mentioned. Thus, another dataset is created with the daily average AQI and hour-wise AQI as two features and the true value of the AQI that is obtained from the source as the target or output. Also, to prevent the LSTM models from stagnating over a long period of forecasting we add a little noise to the AQI output of each of the models to represent random unprecedented variations in the climate and weather that affects air pollution.

We use non-linear least squares to fit a function to the data. 2 separate parameters can be used as shown in the equation or use complemented parameters. The RMSE of the hour-wise forecastusing only standalone hour-wise trained model is 43.09 when compared to the true AQI value. Table 3 shows the improvement in error on employing the curve-fitting algorithm.

Table 3. RMSE Scores for combining the AQI values with different types of parameters and randomization.

Parameter type	**Max Score improvement (No randomization)**	**Max score improvement (With randomization)**
Complemented	0.31	0.33
Independent	0.26	0.42

Curve-fitting does improve the performance of the hybrid model but the difference is not significant. This is due to the fact that the dependence of the target variable on the features is not linear. An alternative method that gives better results is to combine the hour-wise and daily AQI forecasts using an artificial neural network (ANN)model with the hour-wise and daily average AQI as inputs. This model is trained on the same dataset as curve-fitting to give the output of the combined AQI. The use of ANN model also has an additional advantage – it is more generalized than the curve-fitting method. This is necessary to ensure that the hybrid model performs well on unknown data in the real world.Although this increases the space and time (for training) complexity, the improvement in the forecast is noticeably greater. A simple ANN was built with 4 units in the input layer with a dropout of 50% and 10 units in the hidden layer with a dropout of 20% and ReLU (Glorot, Bordes and Bengio, 2011) activation function in both layers. The model gives a real-valued output i.e. the forecasted AQI. The model was compiled with mean-squared logarithmic error (MSLE) as the loss function and ADAM (Kingma & Ba, 2014) optimizer. MSLE has been used as the output values are scattered and it is desired to reduce the penalty in loss function for large differences in target and predicted values. The RMSE improvement is **10.67** after 2000 epochs with a batch size of 10. Fig. 6 shows the values forecasted by the ANN in comparison to the true values of AQI and the loss plot that shows a good fit to data.

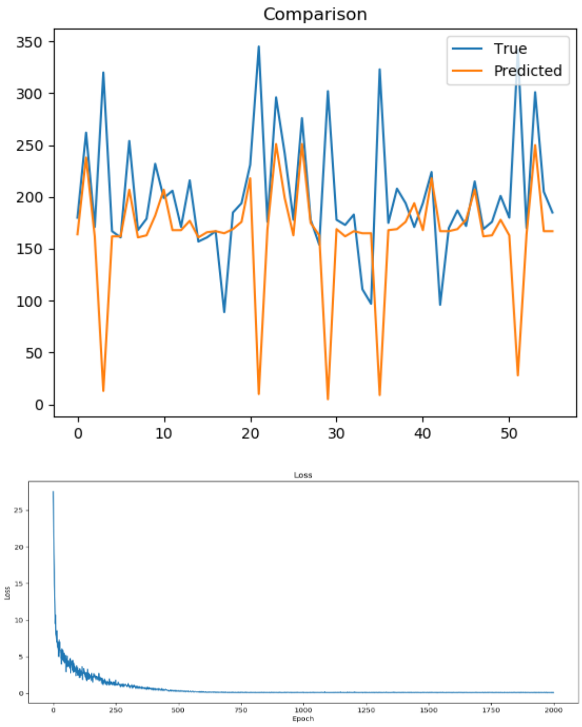


Figure 6(a) True and predicted values of the trained ANN mode over 60 days. **(b)** Training loss (Loss vs Epoch) plot over 2000 epochs. Convergence indicates good fit

4.5 Using the final model

After both neural networks have been trained for forecasting and a choice for combining the outputs has been made, we need to maintain the datasets of AQI used for training and include the computed values in these data sets in chronological order. This will help to maintain the model's performance in the long term. The walkthrough for using the model for forecasting AQI is as follows:

- Take input of date and hour from the user.
- Use the date and hour given to check if the AQI for the given time is already present in the dataset or not.
- If present, the output is given then exit.
- Else check if the AQI values of the previous timestamps are present in both datasets.
- If there are no blanks in the datasets i.e. all the lag variables are available to be used by the hybrid model for forecasting use the model to compute the output AQI.
- If not, compute or forecast the values of the previous missing timestamps first before forecasting the value of the given time. This is a basic dynamic programming approach: we forecast the AQI of the future timestamps before the given input timestamp that are not available in the datasets and use this to compute the AQI of the input time.
- Give the output of the AQI value of the given time.
- Finally, update the AQI values of all the timestamps that have been computed in the process so far in the datasets and exit.
- Periodically, update the database of average daily AQI values and hour-wise AQI values to maintain the performance of the hybrid model.

In Fig. 7 there is a tradeoff between using a weighted linear combination of the individual forecasts and an ANN for the purpose. The first approach is simple but is limited in its capability to model the observed non-linear dependency of the true output value on the features. The second approach is complex in terms of training time and space. However, once the ANN has been trained the computation of output is fast and also the results are significantly better than the previous approach.

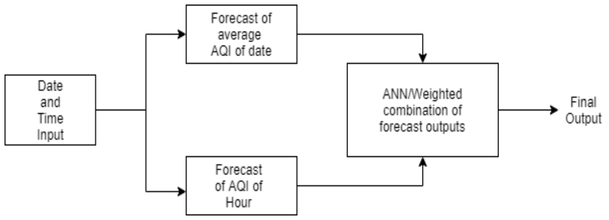


Figure 7. Flowchart of using the model to forecast future AQI

The step of updating the database is of vital importance as the output of the LSTM neural network may get stranded at a fixed value if the dataset is not able to represent the effect of external factors that cause the variations in the time-series variable in our problem the AQI value. Since we are using only one time-series variable and the challenge of procuring a large dataset this step is indispensable. The AQI data for a particular timestamp that need not be forecasted and is available in the source can be updated with the true value to avoid this problem.

5. CONCLUSION AND DISCUSSION

In this paper, we have built, examined and carried out a comparative study of a hybrid neural network model with LSTM to forecast Air Quality Index. The results of our model have been compared to standalone neural network models to show the enhancement of the accuracy of the forecast. Some variations of the hybrid model have also been suggested to improve the overall performance of the model and reduce the computation time of the forecast. The model has been tested with data gathered from a single station however, the same procedure can be extended to forecast AQI of other cities and regions as well. As a part of further development in this work, the model is desired to be included in an application in mobile devices for use in everyday lives.

References:

Agarwal, C. (2018). *Neural Networks and Deep Learning* [Ebook] (pp. 1-52). Springer, Cham. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-94463-0_1

Arlinghaus, S. (1994). *Practical handbook of curve fitting*. Boca Raton, Fla.: CRC Press.

Bai, L., Wang, J., Ma, X., & Lu, H., (2018). Air Pollution Forecasts: An Overview. *International Journal of Environmental Research and Public Health*, *15*(4), 780.

Banihashemi, S., Ding, G., & Wang, J., (2017). Developing a Hybrid Model of Prediction and Classification Algorithms for Building Energy Consumption. *Energy Procedia*, *110*, 371-376.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions On Neural Networks*, *5*(2), 157-166. doi: 10.1109/72.279181

Garg, S., Maharana, S., Paul, B., Dasgupta, A., & Bandyopadhyay, L. (2018). Exposure to indoor air pollution and its perceived impact on health of women and their children: A household survey in a slum of Kolkata, India. *Indian Journal Of Public Health*, *62*(3), 182. doi: 10.4103/ijph.ijph_259_18

Glorot, X., Bordes, A., & Bengio, Y., (2011). Deep Sparse Rectifier Neural Networks. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. PMLR, 315-323. Retrieved 15 November 2019, from http://proceedings.mlr.press/v15/glorot11a.html.

Graves, A., & Schmidhuber, J. (1995). Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks. In *NIPS'95: Proceedings of the 8th International Conference on Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press.

Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2009). A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, *31*(5), 855-868. doi: 10.1109/tpami.2008.137

Greenstone, M., Harish, S., Pande, R., & Sudarshan, A. (2017). The Solvable Challenge of Air Pollution in India. In *Indian Policy Forum*. New Delhi: National Council of Applied Economic Research. Retrieved from http://www.ncaer.org/publication_details.php?pID=301

Greff, K., Srivastava, R., Koutnik, J., Steunebrink, B., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions On Neural Networks And Learning Systems*, *28*(10), 2222-2232. doi: 10.1109/tnnls.2016.2582924

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735-1780. doi: 10.1162/neco.1997.9.8.1735

Kantz, H., & Schreiber, T. (2010). *Nonlinear time series analysis*. Cambridge: Cambridge Univ. Press.

Kingma, D., & Ba, J., (2015). Adam: A Method for Stochastic Optimization. *ICLR*

Le, V. and Cha, S., (2018). Real-time Air Pollution prediction model based on Spatiotemporal Big data. *The International Conference on Big data, IoT, and Cloud Computing (BIC 2018)*, Retrieved 27 May 2020, from https://arxiv.org/abs/1805.00432.

Li, X., & Wu, X. (2015). Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition. *2015 IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP)*. doi: 10.1109/icassp.2015.7178826

Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. *Proceedings Of The 8Th ACM SIGMOD Workshop On Research Issues In Data Mining And Knowledge Discovery - DMKD '03*, 2-11. doi: 10.1145/882082.882086

Miljanović, M. (2012). Comparative analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction. *Indian Journal Of Computer Science And Engineering*, *3*(1), 180-191. Retrieved from http://www.ijcse.com/docs/INDJCSE12-03-01-028.pdf

Mishra, M. (2019). Poison in the air: Declining air quality in India. *Lung India*, *36*(2), 160. doi: 10.4103/lungindia.lungindia_17_18

Miškovic, V. (2014). Machine Learning of Hybrid Classification Models for Decision Support. *Proceedings Of The 1St International Scientific Conference - Sinteza 2014*, 318-323. doi: 10.15308/sinteza-2014-318-323

Olah, C. (2015). Understanding LSTM Networks. Retrieved 10 May 2019, from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

O'Neil, S., Mac, A., Rhodes, G., & Webster, M. (2015). Model Fitting Versus Curve Fitting: A Model of Renormalization Provides a Better Account of Age Aftereffects Than a Model of Local Repulsion. *I-Perception*, *6*(6), 204166951561366. doi: 10.1177/2041669515613669

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *Arxiv*, *1609.04747*. Retrieved from https://arxiv.org/pdf/1609.04747.pdf

Sak, H., Senior, A., & Beaufays, F. (2014). Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling. *15Th Annual Conference Of The International Speech Communication Association, INTERSPEECH-2014*, 338-342. Retrieved from https://www.isca-speech.org/archive/interspeech_2014/i14_0338.html

National Air Quality Index. (2020). Retrieved 10 May 2019, from https://app.cpcbccr.com/AQI_India/

Sundermeyer, M., Schluter, R., & Ney, H. (2012). LSTM neural networks for language modeling. *Proceedings Of The International Speech Communication Association, INTERSPEECH*, 194-197.

Ullah, M. (2020). Time Series Analysis and Forecasting. Retrieved 12 December 2019, from http://itfeature.com/time-series-analysis-and-forecasting/time-series-analysis-forecasting

Warren Liao, T. (2005). Clustering of time series data—a survey. *Pattern Recognition*, *38*(11), 1857-1874. doi: 10.1016/j.patcog.2005.01.025

Dr. Shirshendu Roy
Jadavpur University
Kolkata
India
mtq0306@yahoo.com

Pratyay Mukherjee
SRM Institute of Science and Technology
Kattankulathur
India
tintingetgoing@gmail.com