Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 132 (2018) 1394–1403

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2018)

DeepAirNet: Applying Recurrent Networks for Air Quality Prediction

Athira V[a], Geetha P[b], Vinayakumar R[ab], Soman K P*

Centre for Computational Engineering and Networking, Amrita School of Engineering, Amrita Vishwa Vidyapeetham,Coimbatore-641112, India

Abstract

With the quick advancement of urbanization and industrialization, air pollution has become a serious issue in developing countries. Governments and natives have raised their increasing concern regarding air contamination since it influences human well-being and economic advancement around the world. Traditional air quality prediction methods depend on numerical data and require more computational power for the estimation of pollutant concentration and thus producing an unsatisfactory result. To tackle this problem, we applied widely used deep learning model. The pollutant considered for this work is Particulate Matter 10 (PM10). In this framework, Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) are used for forecasting, based on the pollution and meteorological time series AirNet data. To figure out best architecture, we examined extensive analysis of different RNN models and its variations with its topologies and model parameters. Every experiment was run up to 1000 epochs by varying the learning rate in the range [0.01, 0.5]. It is observed from the study that all the three models performed comparatively well in prediction.

*Keywords:*Air quality; PM10; feed forwards neural network (FFN); deep learning: recurrent neural network (RNN); long short-term memory (LSTM); Gated Recurrent Unit (GRU)

1. Introduction

Rapid urbanization has resulted in air pollution due to the increase in number of vehicles and industrial activities. It leads to higher mortality and also caused huge economic loss. Particulate matter 10 (PM10) and Particulate matter 2.5 (PM10) are main contributors to air pollution. These particulate matter affects the cardiovascular system and

* *Athira V. Tel.: 9400612040, **Geetha P. Tel.: 9843060402.*
E-mail address:athirav131@gmail.com, p_geetha@cb.amrita.edu

10.1016/j.procs.2018.05.068

other internal organs by ingestion. According to World Health Organization (WHO), there are 51 different cities in the world where the main categories of PM has been identified [1]. Developing countries such as India and China has affected the most due to atmospheric pollution.

It is necessary to know the sources as well as quantity of these pollutants to minimize the adverse health effects of atmospheric pollution. Computing these fine particles serves as indicator for air quality. To control the air pollution and safeguard people from its effect, real-time air quality data is required. Therefore, air quality prediction is mandatory to yield entire information about the air quality in taking major decision for environmental management.

Many studies have been carried out for air quality predictions. Models like Hysplit-4 [2] and KF [3] utilizes the dynamic process in atmosphere and figures out the accumulation and dissipation mechanisms of pollutants in the atmosphere. For measuring the increase in pollution, air quality forecasting techniques has been rapidly upgraded on demand. Models like Recurrent neural network (RNN) and Long short term memory (LSTM) are applied to perform the meteorological forecasting, weather forecasting [4], network traffic prediction [5] and also the estimation of air pollution and precipitation probability [6].

Deep learning uses multiple layer architecture for the extraction of intrinsic features in layers. It extracts the data from bottom to the topmost level and can determine the characteristic pattern in data. The spatial distribution and temporal trends of air quality process are affected by various factors like air pollution emission and depositions, weather conditions, human activities and so on, which makes the process complicated. This condition resulted in an increase in trouble of utilizing conventional trivial models generally for procuring a better pattern. Deep learning can lead to good performance for air quality predictions by extracting the air quality features without knowing any information from the past.

In this paper, deep learning model is applied for predicting the air quality. The architectures used for evaluation are RNN, LSTM and GRU. Here we have used AirNet dataset [7],it includes both meteorological features and air quality data.

2. Related Work

Deterministic and statistical approaches are the two methods performed for air quality predictions. Meteorological emissions and chemical models are employed using deterministic methods for the simulation of pollutant remittance, its exchange, diffusion and expulsion process using dynamic information of fixed number of monitoring stations [8]. Saide et.al.(2011) proposed CMAQ and WRF-Chem for urban air quality forecasting [9]. But it results in low prediction accuracy due to the irregularity in pollutant data, complex underlying atmospheric conditions and insufficient theoretical information.

Statistical methods predict the air quality using statistical modeling approach in a data-driven way. Auto regression moving average [10] and multiple linear regression (MLR) [11] methods are employed in air quality predictions. Accuracy obtained is limited in these methods because of their inability to model non-linear data, thus prediction of air concentration predictions are not exact [12]. One of the alternatives to this model is support vector regression model and artificial neural networks (ANN). A previous study conducted by Prybutol et al. (2000) showed that non-linear models like ANN produce more accurate result than the linear models because a clearer non-linear pattern is present in air quality data [13]. Many studies have carried out using the combination of these models and the outcomes demonstrates that satisfying predictive performance is obtained from hybrid models than single models [14].

Shi et al. (2015) proposed RNN and convolutional LSTM to figure the precipitation in future two hours, which they formalized as aspatio-temporal issue [15]. The air quality estimate is like climate gauge, however two components make air quality conjecture more troublesome and particular from assessing precipitation; 1) the time traverse of air quality conjecture is longer than climate estimate, the previous regularly conjectures in four or five days, here and there even goes past ten days. 2)extra persuasive variables must be considered in air quality gauge, for example, the progression of air poisons and the connection with meteorological conditions.

Ong et al. (2016) connected RNN to anticipate PM2.5 with natural sensor information, which moved forward the outcomes precision [16]. Kurt and Oktay (2010) research on anticipating air contamination with neural systems showed the strategies predominance and achievability [17]. Liang et al. (2015) discharged a dataset containing the estimation of PM2.5 which is just estimated in Beijing [18]. After this, Liang et al. (2016) distributed a bigger dataset to break down the poison factor in five urban areas of China [19]. All datasets utilized above are point-wise information, which don't enable us to demonstrate in a spatially express way.

3. Background

This section discusses about the background information of RNN, LSTM and GRU network architecture.

3.1. Recurrent Neural Network (RNN)

RNN was introduced for modeling the time series data. In RNN, a cyclic loop is added to the feed forward network, which is shown in Fig.1(a). The loop carries information of different time steps and executes the temporal task [20]. Accordingly, RNN learns the temporal pattern and estimates the value at an instant based on the previous and current values.

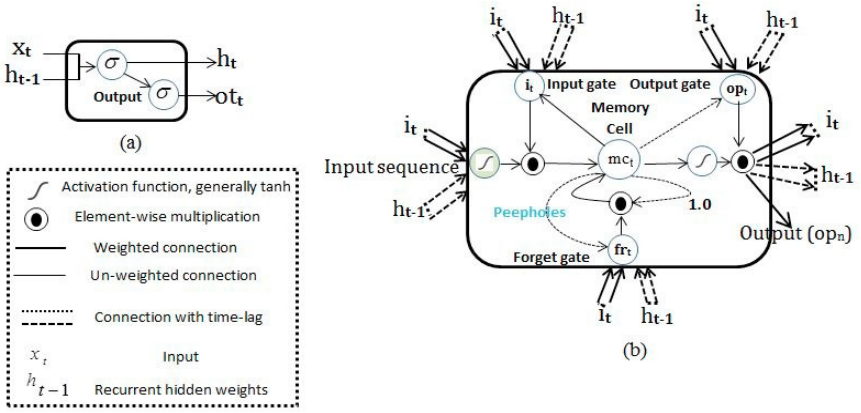

Fig. 1. (a) RNN architecture (b) LSTM architecture

Let $x=(x_1,x_2,x_3\ldots,x_{T-1},x_T)$ be the input time series AirNet data given to RNN network which maps to the hidden vector sequence $h=(h_1,h_2,h_3\ldots,h_{T-1},h_T)$. The output obtained for the time step $t=1$ *to* T is $O=(o_1,o_2,o_3,\ldots,o_{T-1},o_T)$ which is followed by the equations given below:

$$H(x,h) = f(w_{hx}x + w_{hh}h + b) \quad (1)$$

where, $f: R \rightarrow R$, $H: R^d \; X \; R^k \rightarrow R^k$, $w_{hx} \varepsilon R^k \; X \; d$, $w_{hh} \varepsilon R^k \; X \; k$ and $b \; \varepsilon R^k$, f is the *sigmoid* activation function(σ) and $w_{hx}, w_{hh}$are the weight matrices. The output sequence is given by the equation as follows:

$$o\_t = \sigma\,(w_{oh}h_t + b_o) \quad (2)$$

3.2. Long Short-Term Memory

LSTM is an enhanced strategy of traditional RNN. LSTM overcomes the vanishing gradient problem caused by RNN, by introducing a memory block. A memory block is a complex processing unit with a constant error carousel (CEC) value 1, consisting of atleast one memory cell and couple of multiplicative gates as its input and output. The error value will be active across the time step and gets triggered when the memory block does not receive any outside signal values. The multiplicative gates have control over the entire operations in memory block. An input gate controls the cell activation for the flow of input into a memory cell. The output gate controls the flow of output from a memory cell into other nodes [20]. The LSTM architecture is shown in Fig.1(b).

LSTM architecture takes the input time series AirNet data $x=(x_1,x_2,x_3\ldots,x_{T-1},x_T)$ and estimates an output $ot=(ot_1,ot_2,ot_3,\ldots,ot_{T-1},ot_T)$ by updating 3 multiplicative unit (input (i), output (op) and forget gate (fr)) on memory cell with continuous write, read and reset operations on memory cell (mc) from time $t=1$ *to* T. The function for the recurrent layer for the time step T is given as:

$X_t, h_{t-1}, mc_{t-1} \rightarrow h_t, mc_t$

$$i_t = \sigma\,(w_i x_t + w_{hi}\, h_{t-1} + w_{mci} mc_{t-1} + b_i) \quad (3)$$

$$fr_t = \sigma (w_{xfr} x_t + w_{hfr} h_{t-1} + w_{mcfr} mc_{t-1} + b_{fr})(4)$$

$$mc_t = fr_t \cdot mc_{t-1} + i_t \cdot tanh (w_{xmc} x_t + w_{hmc} hi_{t-1} + b_{mc})(5)$$

$$op_t = \sigma (w_{xop} x_t + w_{hmc} hi_{t-1} + b_{op})(6)$$

$$h_t = op \cdot tanh (mc_t)(7)$$

where i_t, op_t, fr_t are the input, output and forget gate respectively. i_t is served for getting the new information and $f_t$is served for remembering the old information. b_i, b_{fr}, b_{mc}, b_{op} are the bias units of the input, forget, memory and output gate respectively. *mc, w*and *h* are memory cell, weight matrix and output of the hidden layer respectively. *sigma* and *tanh* are the activation function in the range [0,1] and [-1,1] respectively.

3.3. Gated Recurrent Network

GRU is an extension to LSTM network. It consists of update and forget gates. They altogether include the balancing of flow of data inside the unit [20]. AirNet time series data is given to the GRU as input. GRU can be defined using the following formula:

X_t, $h_{t-1} \rightarrow h_t$

$$i\_fr_t = \sigma (w_{xi\_fr} x_t + w_{hii\_fr} h_{t-1} + b_{i\_fr})(8)$$

$$fr_t = \sigma (w_{xfr} x_t + w_{hifr} h_{t-1} + b_{fr})(9)$$

$$mc_t = tanh (w_{xmc} x_t + w_{hmc} (fr \, \Theta \, hi_{t-1}) + b_{mc})(10)$$

$$h_t = fr \Theta h_{t-1} + (1-fr) \, \Theta \, mc(11)$$

where*i, fr*and *i_fr* denotes input, forget and update gates respectively. *mc* is the previous state and *h*is the candidate activation. The update gate is formed by combining input and forgets gates, which balances the condition between past activation, candidate activation and output activation in the absence peephole connections. Forget gate resets previous state. GRU requires fewer computations and is simpler than the LSTM networks.

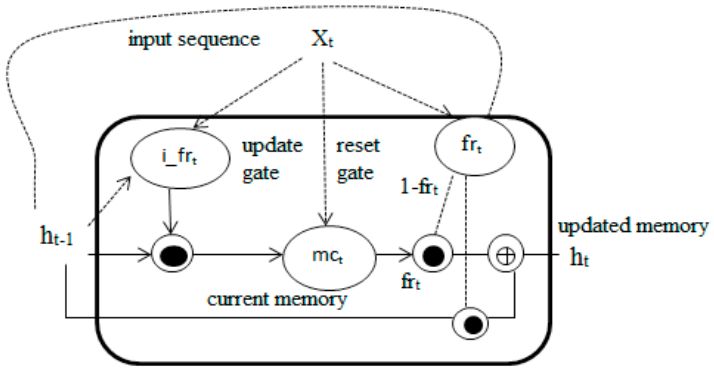

Fig. 2. GRU architecture

4. Experiments

All experiments are run on GPU enabled TensorFlow [21] in conjunction with Keras [22] framework. All recurrent architecture are trained using backpropagation through time (BPTT) [23] with *RMSprop* as an optimizer.

4.1. Description of Dataset

For this work AirNet dataset is used [7]. It consists of 6 indices of air quality collected from 1498 stations from China National Environmental Monitoring Center (CNEMC) and meteorological data gathered from Global Forecasting System (GFS). Each meteorological feature contains 1038240 values at one time point globally. GFS contains a 3 dimensional matrix released by NOAA every 6 hours. This 3 dimensional matrix is interpolated with 2 dimensional data to form a four dimensional matrix to obtain AirNet data. The four dimensional database contains information of (latitude, longitude, timesteps, features). The data was collected from April 1, 2015 to September 1, 2017. For each time frame, there is 6 GFS feature and 7 air quality indices. The total dimension of data is (132, 228, 7072, 13).

Table 1.GFS Features.

Data	Pollutants	Samples
AirNet	$PM_{10}, PM_{2.5}, NO_2, CO, O_3, SO_2, AQI$	10593856

Table 2.GFS Features.

Number	Column A (t)
1	Temperature (K)
2	Relative Humidity (%)
3	U-wind component (m/s)
4	V-wind component (m/s)
5	Precipitation Rate (kg/m^3/s)
6	Total cloud cover (%)

4.2. Selecting Network Parameters

RNN, LSTM and GRU contain parameterized functions which will directly affects the ideal parameters of the data giving better prediction. Different analyses and configurations are employed on the data to obtain the ideal estimations of parameters, for example, learning rate and number of units/memory blocks.

In this paper we have used three architectures such as RNN/LSTM/GRU. It consists of input-layer followed by hidden-layer and then the output-layers. A completely associated organizes: the association between neurons in the contribution to input layers and from hidden layer to output layer; is utilized as a part of the investigation. The train and test data are normalized in the experiment. The range of memory block taken was from four to sixty-four and three trial runs are performed to guarantee the accuracy. Different parameters are used as a part of the experimentation. Loss function is defined by mean square error (*MSE*), *RMSprop* is the optimizer. The batch size is taken as 32, dimension of data is 6 and total number of epochs taken is 100.

In this work, three trials of experiment are run for memory block 4, 8, 32 and 64. As the performance of memory block with 32 and 64 are similar, the memory block size is set to be 32 for rest of the experiments. Three trial runs is conducted in the experiment with learning rates ranging from 0.01 to 0.5 to obtain the satisfactory learning rate.

Optimal prediction is obtained for lower learning rates, so we considered the learning rate to be 0.01 for the experimentation based on different factors such as training time and prediction rate. Each units/memory blockscontains four layers. Three set of experiments are run for each of the topologies till 100 epochs. Among that the performance of network with four layers gives the optimal result. Based on this, four layers are set for each network.

4.3. Network Topologies

The following network topologies have been used to select the optimal network for RNN/LSTM/GRU:

- RNN/LSTM/GRU 1 layer with 32 units/memory blocks
- RNN/LSTM/GRU 2 layer with 32 units/memory blocks
- RNN/LSTM/GRU 3 layer with 32 units/memory blocks
- RNN/LSTM/GRU 4 layer with 32 units/memory blocks

Each units/memory blocks contains four layers. Three set of experiments are run for each of the topologies till 100 epochs. Among that the performance of network with four layers gives the optimal result. Based on this, four layers are set for each network.

4.4. Proposed Architecture

Deep learning learns the optimal features by taking raw time series data as input. The hidden layer learns the abstract input features of data. Fig.3.shows the proposed architecture.

- Input: Normalization process is applied to the dataset to bring together the data in the range of [0, 1]. 6 dimension data is given as the input the recurrent structure. From the available data, the data for the month September 2017 is considered for evaluation. The data points taken to train the model is 17,00016. The data given for testing is 877632.
- Recurrent Structure: Three distinctive neural network structures are used in this study: RNN, LSTM and GRU. 32 units/memory blocks are contained in each recurrent structure. Existence of long-term dependencies among the data is concentrated in this study. All the three architecture were effective in determining the long-term dependencies and models can be utilized for future prediction. The train and test information were normalized utilizing the similar strategy.
- Prediction: The extracted features are used for the future prediction. Dense neural network is used, which consists two layers. The first layer is Dense (1) unit, which is followed by DenseLinear layer with 10 units. Given the recurrent features, the dense layer learns a non-linear kernel. The linear activation function yield gives the 10 times step predicted esteems given the yield of the last dense layer.

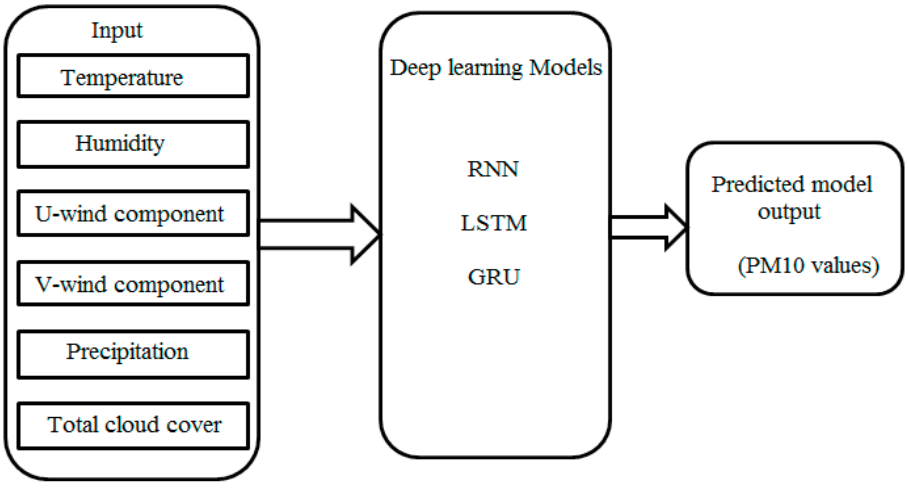

Fig. 3. Proposed Architecture

Models were trained for 1000 epochs by shifting the layer size to enhance the efficiency. If the Mean Square Error (*MSE*) of the past epoch is greater than that of the current epoch, then the weight matrices are stored. After

obtaining the trained models, each data points are tested and, Root Mean Square Error (*RMSE*) is calculated. Model with small *RMSE* is taken as the final prediction model. *MSE* is given by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 (12)$$

where n is the total number of predicted value and y is the predicted value. All the experiments related to different configurations of RNN architectures are run till 1000 epochs. *RMSprop* optimizer is used here. De-normalization is applied to the predicted data output. Mean Absolute Percentage Error (*MAPE*) is formulated as follows:

$$MAPE = abs\left(\frac{predictedvalue - realvalue}{realvalue}\right) \times 100 (13)$$

5. Results

The mean square error (*MSE*) on training data for one epoch is given in Table 3. It can be seen that GRU layer 1 is having less mean square error compared to all other proposed models.

Table 3. MSE obtained for different models for one epoch

Model	MSE
RNN layer 1	0.00025
RNN layer 2	0.00038
RNN layer 3	0.00047
RNN layer 4	0.00056
LSTM layer 1	0.00058
LSTM layer 2	0.00048
LSTM layer 3	0.00057
LSTM layer 4	0.00066
GRU layer 1	0.00023
GRU layer 2	0.00039
GRU layer 3	0.00043
GRU layer 4	0.00042

The details of RNN, LSTM and GRU architecture used for this work is given in Table 4, 5, 6 respectively.

Table 4.Details of RNN architecture.

Layer (type)	Output shape	Parameters
RNN layer 1	(None , 32)	1248
Dense layer 1	(None, 1)	33
Activation layer 1	(None, 1)	0

Table 5.Details ofLSTM architecture.

Layer (type)	Output shape	Parameters
LSTM layer 1	(None , None, 32)	4992
LSTM layer 2	(None, None, 32)	8320
Dense layer 1	(None, 1)	33
Activation layer 1	(None, 1)	0

Table 6.Details of GRU architecture.

Layer (type)	Output shape	Parameters
GRU layer 1	(None , None, 32)	3744
Dense layer 1	(None, 1)	33
Activation layer 1	(None, 1)	0

In this paper, we analyzed the air pollution data using three deep learning models: RNN, LSTM and GRU. *RMSE* and *MAPE*is estimated for all the three models. The results obtained are comparable. Among the three models GRU gives the best performance with *MAPE*= 0.4525. The obtained results are tabulated in the Table 7.

It is clear from the table that RNN layer 1 gives lesser *MAPE* value when compared with all other RNN layers performed in this model. Similarly, LSTM layer 2 gives lesser *MAPE* value compared with all other LSTM layers in the model. Also, GRU layer 1 is having lesser *MAPE* value compared with all the GRU layers, LSTM layers and RNN layers. Thus GRU gives the better performance among the three models performed.

Table 7. RMSE and MAPE obtained for various Networks

Model	RMSE	MAPE
RNN layer 1	0.4072	0.5729
RNN layer 2	0.4131	1.2823
RNN layer 3	0.5073	7.8391
RNN layer 4	0.4196	1.3503
LSTM layer 1	0.4075	0.5471
LSTM layer 2	0.4095	0.5231
LSTM layer 3	0.4087	0.8004
LSTM layer 4	0.4092	0.6947
GRU layer 1	0.4077	0.4525
GRU layer 2	0.5648	7.0150
GRU layer 3	0.8105	6.6292
GRU layer 4	0.4111	0.7489

Fig.4., Fig.5.and Fig.6.shows the real value and predicted value for test data with minimum $MAPE$ error. The best obtained result for each model is plotted. Fig.4 and Fig.5 shows the RNN layer 1 and LSTM layer 2 plot respectively. From the plot we can understand that initially both LSTM and RNN are not capturing the trend accurately, but towards the end a clear pattern is obtained. From Fig.6, we can see that GRU layer 1 is capable of capturing the trend clearly. Hence GRU has better performance than RNN and LSTM as the predicted pattern is similar to the original pattern.

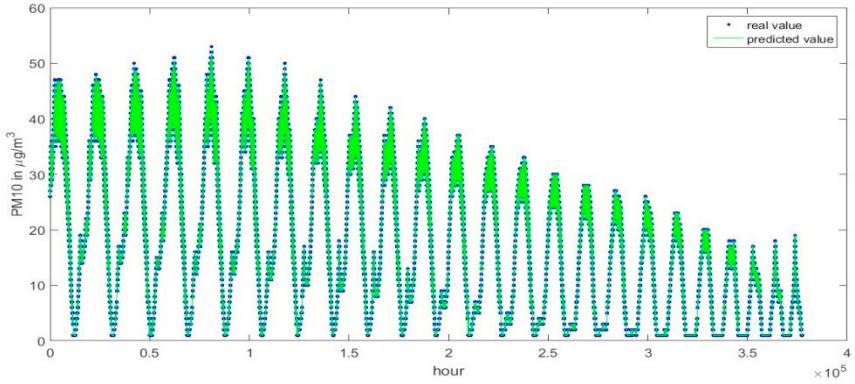

Fig. 4. RNN layer 1 plot for real vs predicted value with MAPE= 0.5729

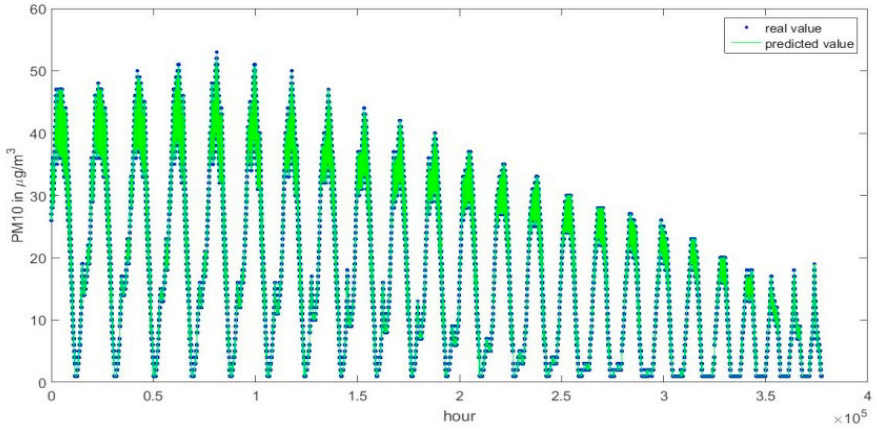

Fig. 5. LSTM layer 2 plot for real vs predicted value with MAPE= 0.5231

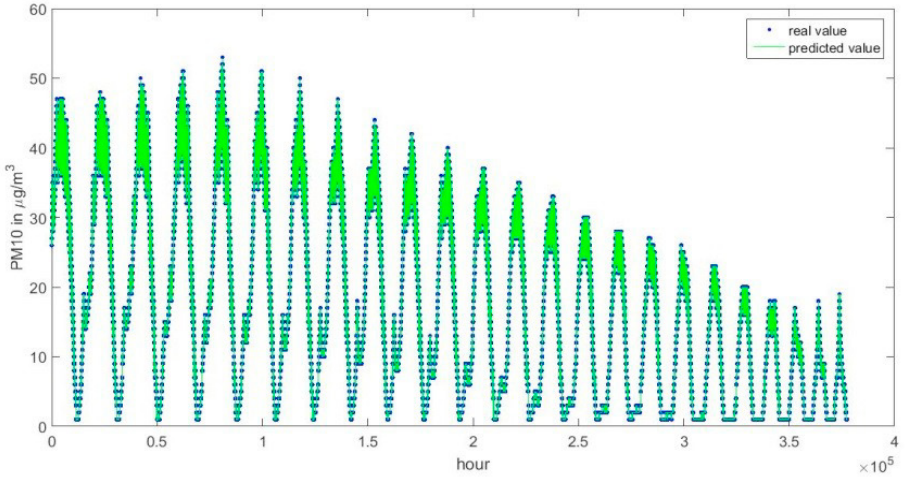

Fig. 6. GRU layer 1 plot for real vs predicted value with MAPE= 0.4525

6. Conclusion

Deep learning method is gradually developing as a promising technique for forecasting non-linear time series information like meteorological and pollution data. In this paper, we used different deep learning models for the prediction of air quality. Here we trained RNN, LSTM and GRU networks on AirNet data to predict the future PM10 value. From the results it can be found that the performance of GRU network is slightly higher than the RNN and LSTM networks. Also, we are able to identify the pattern of PM10 concentration.

Since AirNet is a huge dataset, resources like GPU is required for better performance and faster computation. Due to this, we have considered only a subset of AirNet data. We were able to obtain a better prediction with minimal error eventhough, only the subset of dataset is used.

The analysis can be further extended by utilizing methods like Convolution Neural Network (CNN) to catch the uneven changes happening in the air pollution data. The connection between different features can likewise be assessed hence enabling us to see whether there is any hidden parameter which will correlate the performance of features that appears to be different from the first peek.

References

[1] World Health Organization. WHO Air quality guidelines for particulate matter, ozone, nitrogen dioxide and sulphur dioxide.Global Update 2005.Summary of risk assessment. Google Scholar. 2005.

[2] Geetha P, Kokila M. (2015) "Estimation of air pollution using remote sensing technique in coimbatore-a case study." In Communications and Signal Processing (ICCSP), 2015 International Conference, IEEE: 0794-0798

[3] Djalalova I, DelleMonache L, Wilczak J. (2015) "PM2.5 analog forecast and Kalman filter post-processing for the Community Multiscale Air Quality (CMAQ) model." Atmospheric Environment**108**: 76-87

[4] Ghaderi A, Sanandaji B. M., Ghaderi, F. (2017) "Deep Forecast: Deep Learning-based Spatio-Temporal Forecasting." arXiv preprint arXiv: 1707.08110.

[5] Vinayakumar R, Soman K P, PoornachandranPrabaharan. (2017) "Applying deep learning approaches for network traffic prediction."InAdvances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on 2017 Sep 13 (pp. 2353-2358). IEEE: 1677-1683.

[6] Xingjian S. H. I, Chen Z, Wang H, Yeung D. Y, Wong W. K, Woo W. C. (2015) "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." In Advances in neural information processing systems: 802-810.

[7] Songgang Zhao, Xingyuan Yuan, Da Xiao, Jianyuan Zhang, Zhouyuan Li. (2018) "AirNet:a machine learning dataset for air quality forecasting"

[8] Baklanov A, Mestayer P, Clappier A, Zilitinkevich S, Joffre S, Mahura A, Nielsen N. (2008) "Towards improving the simulation of meteorological fields in urban areas through updated/advanced surface fluxes description." Atmospheric Chemistry and Physics **8**: 523-543

[9] Saide P, Carmichael G, Spak S, Gallardo L, Osses A, Mena-Carrasco M, Pagowski M. (2011) "Forecasting urban PM10 and PM2.5 pollution episodes in very stable nocturnal conditions and complex terrain using WRF–Chem CO tracer model." Atmospheric Environment **45**: 2769-2780

[10] Box G. E, Jenkins G. M, Reinsel G. C, Ljung G. M. (2015) "Time series analysis:forecasting and control." John Wiley &Sons.

[11] Li C, Hsu N, Tsay S. (2011) "A study on the potential applications of satellite data in air quality monitoring and forecasting." Atmospheric Environment **45**: 3663-3675

[12]Goyal P, Chan A, Jaiswal N. (2006) "Statistical models for the prediction of respirable suspended particulate matter in urban cities." Atmospheric Environment **40**: 2068-2077

[13] Prybutok V, Yi J, Mitchell D. (2000) "Comparison of neural network models with ARIMA and regression models for prediction of Houston's daily maximum ozone concentrations." European Journal of Operational Research **122**: 31-40

[14] Díaz-Robles L, Ortega J, Fu J, Reed G, Chow J, Watson J, Moncada-Herrera J. (2008) "A hybrid ARIMA and artificial neural networks model to forecast particulate matter in urban areas: The case of Temuco, Chile." Atmospheric Environment **42**: 8331-8340

[15] Xingjian, S. H. I., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., Woo, W. C. (2015) "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." In Advances in neural information processing systems: 802-810.

[16] Ong B, Sugiura K, Zettsu K. (2015) "Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2.5." Neural Computing and Applications **27**: 1553-1566

[17] Kurt A, Oktay A. (2010) "Forecasting air pollutant indicator levels with geographic models 3days in advance using neural networks." Expert Systems with Applications **37**: 7986-7992

[18] Liang X, Zou T, Guo B, Li S, Zhang H, Zhang S, Huang H, Chen S. (2015) "Assessing Beijing's PM2.5pollution: severity, weather impact, APEC and winter heating." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science **471**: 20150257

[19] Liang X, Li S, Zhang S, Huang H, Chen S. (2016) "PM2.5data reliability, consistency, and air quality assessment in five Chinese cities." Journal of Geophysical Research: Atmospheres **121**: 10,220-10,236

[20] LeCun Y, Bengio Y, Hinton G. (2015) "Deep learning. nature." **521**: 436.

[21] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, Kudlur M. (2016) "TensorFlow: A System for Large-Scale Machine Learning." InOSDI**16**: 265-283.

[22] Chollet F. "Keras."

[23] Werbos PJ. (1990) "Backpropagation through time: what it does and how to do it." Proceedings of the IEEE **78**: 1550-60.