Earth Science Informatics (2020) 13:821–834
https://doi.org/10.1007/s12145-020-00462-9

RESEARCH ARTICLE

Airborne particle pollution predictive model using Gated Recurrent Unit (GRU) deep neural networks

Josue Becerra-Rico[1] · Marco A. Aceves-Fernández[1] · Karen Esquivel-Escalante[1] · Jesús Carlos Pedraza-Ortega[1]

Received: 13 November 2019 / Accepted: 6 April 2020 / Published online: 24 May 2020

Abstract

Developments in deep learning for time-series problems have shown promising results for data prediction. Particulate Matter equal or smaller than 10 μm (PM_{10}) have increased importance in the research field due to the negative impact in the respiratory system. PM_{10} particles show non-linear behavior, hence it is not an easy task to implement techniques to predict subsequent concentration of the particles in the atmosphere. This paper presents a forecasting model using gated Recurrent unit (GRU) and Long-Short Term Memory (LSTM) networks, which are types of a deep recurrent neural network (RNN). The predicted results of PM_{10} are presented using data of Mexico City as a case study, showing that this type of deep network is feasible for predicting the non-linearities of this type of data. Several experiments were carried out for 12, 24, 48, and 120 h prediction, showing that this method may be applied to accurately forecast the behavior of PM10.

*K*eywords Airborne pollution · Gated recurrent unit · Machine learning · Predictive models

Introduction

The atmospheric system that surrounds the earth is a complex fluidic system which can be easily altered by the presence of physical, chemical and biologic compounds. The regular composition of the atmosphere is nitrogen (N_2), oxygen (O_2), carbon dioxide (CO_2), water vapor (H_2O), methane (CH_4), sulfur dioxide (SO_2) and carbon monoxide (CO) as main components among others (Akbarzadeh et al. 2018).

The daily activity of a city generates a huge amount of substances that modify the natural composition of the air. For example, burning fossil fuels to generate energy and transportation produces tons of pollutants which are emitted into the atmosphere (Akbarzadeh et al. 2018).

A few components in minor concentrations such as the greenhouse gases (nitrogen oxides, NOx, Chlorofluorocarbon Compounds, CFC, organic volatile compounds, OVC, and ozone, O_3) along with the CO_2 and CH_4 can produce a negative effect over all the ecosystem (Goglio et al. 2019). The biologic compounds that can be found in the air, is due to the presence of virus and bacteria from animal excreta, or even from respiratory tract diseases of human beings, which represents a human being health concern (Schraufnagel et al. 2018).

Other important pollutants that are rise some alerts in overcrowded cities, and in places where explosive growths in industrial activity is present, all around the world, are the suspended particles (TSP) or particulate matter (PM) (Oh and et al. 2015; Bartzis et.al. 2019; Bos et al. 2019). The toxicity of the particles is determined by their physical and chemical characteristics. For instance, size, which is measured in aerodynamic diameter, is an important parameter to characterize its behavior. This parameter is important due to its capacity of penetration and retention in the human respiratory system. Also, size may determine its residence time in the atmosphere (Bartzis et.al. 2019). Because of their small size, particles on the order of 10 μm or less (PM_{10}) can penetrate into the lungs. It has been widely documented the risk in morbidity and mortality in cardiac and respiratory problems due to high concentration exposure of these pollutants. And the effects can be noticed immediate or felt after several days of exposure (Schraufnagel et al. 2018).

In order to understand and get a better observation of this problem magnitude, several papers deals with the fact that the

Communicated by: H. Babaie

✉ Marco A. Aceves-Fernández
marco.aceves@uaq.mx

[1] Facultad de Ingeniería, Universidad Autónoma de Querétaro, Querétaro, Mexico

task of contaminants predicting behavior is not an easy task as does not present a linear behavior (Bos et al. 2019).

PM_{10} particle concentration sequence modeling has shown good results using statistic and machine learning algorithms (Akbarzadeh et al. 2018; Goglio et al. 2019). Forecasting of airborne pollution is one topic of research due to the negative effects in human health, which problems are not limited to the respiratory system, but also can lead to major and more chronic diseases. The effects of these pollutants, as was mentioned before, could be immediately notices or even few years later after exposure.

Studies on the particle size distribution (PSD) of fine particles in the Mexico City Metropolitan Area (MCMA) are very scarce (Caudillo et al. 2020), and as can be seen in Fig. 1, Mexico City emissions has an important contribution to the air quality in this same place. That is the main reason of this research, which will give a novelty option of particle distribution description, in order to get more information about it, and be able to build in the future, a better understanding behavior of air particles systems (Aceves-Fernandez et al. 2015).

Background

PM10 Modelling

There are many available methods to model the behavior of PM10 particles. These methods include for instance, fuzzy logic, where Yetilmezsoy (2012) implemented a neurofuzzy algorithm for PM10 prediction. In terms of swarm intelligence, Ordonez de Leon (Ordonez de León et al. 2019) implemented a particle swarm optimization (PSO) algorithm for PM10 prediction, whilst Cabrera-Hernández (Cabrera-Hernandez et al. 2019) used a Neuro-fuzzy algorithm optimized with bacteria foraging and Aceves-Fernández (Aceves-Fernandez et al. 2018) used Ant Colony Optimization for the same purpose.

Furthermore, Grivas (Grivas et al. 2006) used a genetic algorithm to predict PM10 on an hourly basis. Also, Franschesci (Franceschi et al. 2018) used a hybrid algorithm based upon k-means classification and PSO. Lastly, Skrzypski (Skrzypski et al. 2009) implemented a recurrent neural network, whilst Domínguez (Domínguez-Guevara 2019) implemented a convolutional neural network.

In this contribution, two approaches were considered to predict the non-linear behavior of PM10 airborne pollution: Long-Short Term Memory (LSTM) Networks and Gated Recurrent Unit (GRU) Networks.

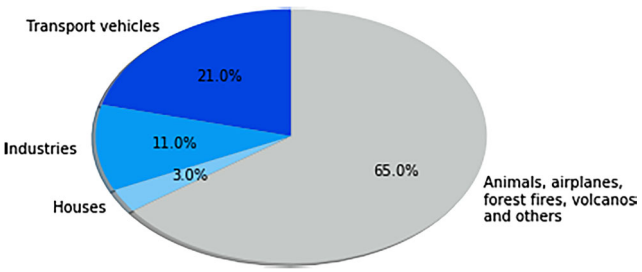

Fig. 1 Contribution of PM10 pollution emitters in Mexico City (data obtained from *Red Automática de Monitoreo Ambiental (*SEDEMA 2018*)*)

Recurrent neural networks

Several multi-layer neural networks have been developed, one of them are Recurrent Neural Networks (RNN), which have as main feature feedback connections that allows them to model time series problems.

It can be observed in Fig. 2, where x_t are the inputs and h_t the output signals. This topology reveals that recurrent neural networks are intimately related to sequences and lists, which makes this method in the past years they have been applied to voice recognition problems, language modeling, and time-series (Längkvist et al. 2014; Montañez et al. 2019).

We can think of simple RNN as a NN with single layer that is repeated and pass the information to the next NN (as shown on Fig. 3). In general, an RNN uses the hyperbolic tangent activation function, which is also used in this study.

Long short term memory

The Long Short Term Memory (LSTM) is a type of RNN. They were introduced in 1997 by Hochreiter and Schmidhuber (Hochreiter and Schmidhuber 1997) designed to avoid the long term dependency problems. They can remember the learned information for long periods of time without any learning problem.

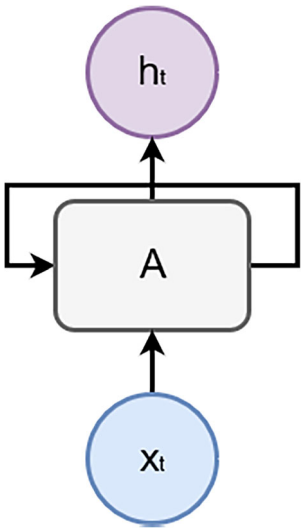

Fig. 2 Recurrent neural network model (Hochreiter and Schmidhuber 1997)

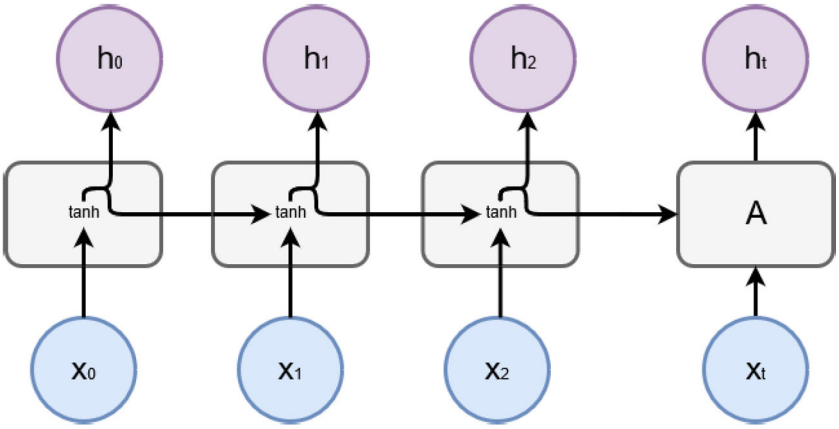

Fig. 3 Interpretation of a recurrent neural as a conventional neural network (Längkvist et al. 2014; Montañez et al. 2019)

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure (see Fig. 2), such as a single tanh layer, LSTM networks also have this chain-like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting as shown on Fig. 3 (Hochreiter and Schmidhuber 1997).

The cell state $C_{t\text{-}1}$ C_t is similar as a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. This way is convenient for information to just flow along it unchanged. The LSTM networks have the ability to remove or add information to the cell state, carefully regulated by structures called gates.

Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer denoted by σ and a point-wise multiplication operation x.

The sigmoid function outputs 1 when the data needed to be kept entirely while 0 erases the data inside the RNN (Hochreiter and Schmidhuber 1997).

The LSTM architecture work as follows:

1. The *forget gate layer* takes the input values x_t and $h_t - 1$, applies the sigmoid function to determine what to preserve and discard, then is multiplied by C_{t-1}.

$$f_t = \sigma\left(W \cdot [h_{t-1}, x_t] + b_f\right) \tag{1}$$

Where W is a weight and b_f the bias.

2. The next step is to decide what information is going to store in the cell state. First, a sigmoid layer called the *input gate layer* decides which values are going to be updated. Next, a tanh layer creates a vector of new candidate values, $\tilde{C}t$, that could be added to the state. In the next step, these two steps are combined to create an update to the state.

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{2}$$

$$\tilde{C}t = tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{3}$$

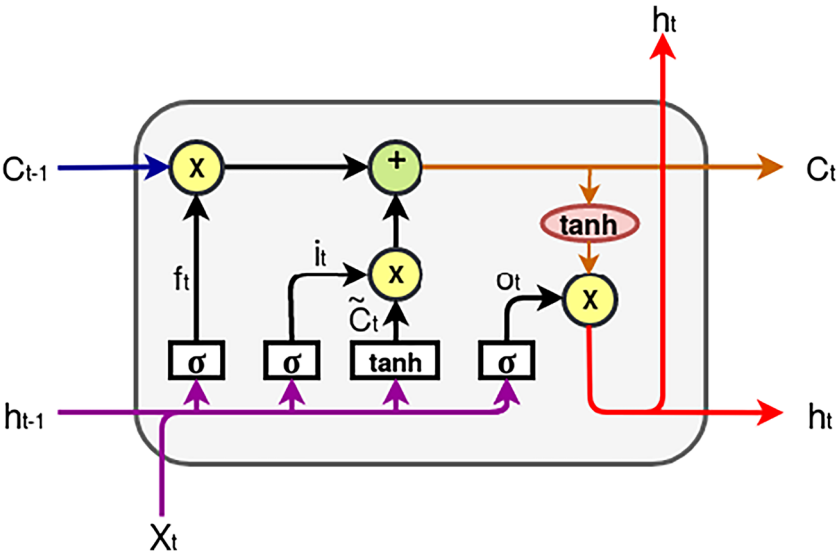

Fig. 4 Architecture of the LSTM. The repeating module in an LSTM contains four interacting layers (Hochreiter and Schmidhuber 1997)

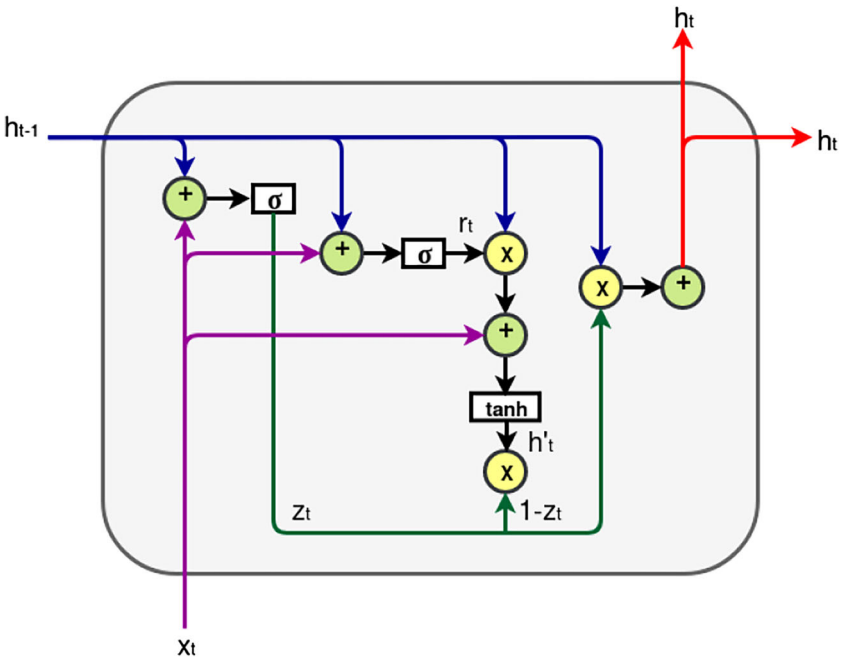

Fig. 5 Gated recurrent unit architecture (Zhou et al. 2016)

3. The old cell state C_{t-1} unto the new C_t is updated

$$C_t = f_t{*}C_{t-1} + i_t{*}\mathrm{C}t \tag{4}$$

4. Finally, an output is chosen (Fig. 4)

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{5}$$

$$h_t = o_t{*}tanh\left(C_t\right) \tag{6}$$

Gated recurrent unit

The Gated Recurrent Unit (GRU) architecture was introduced in 2014 by Cho et al (2014). It is considered a variation of the

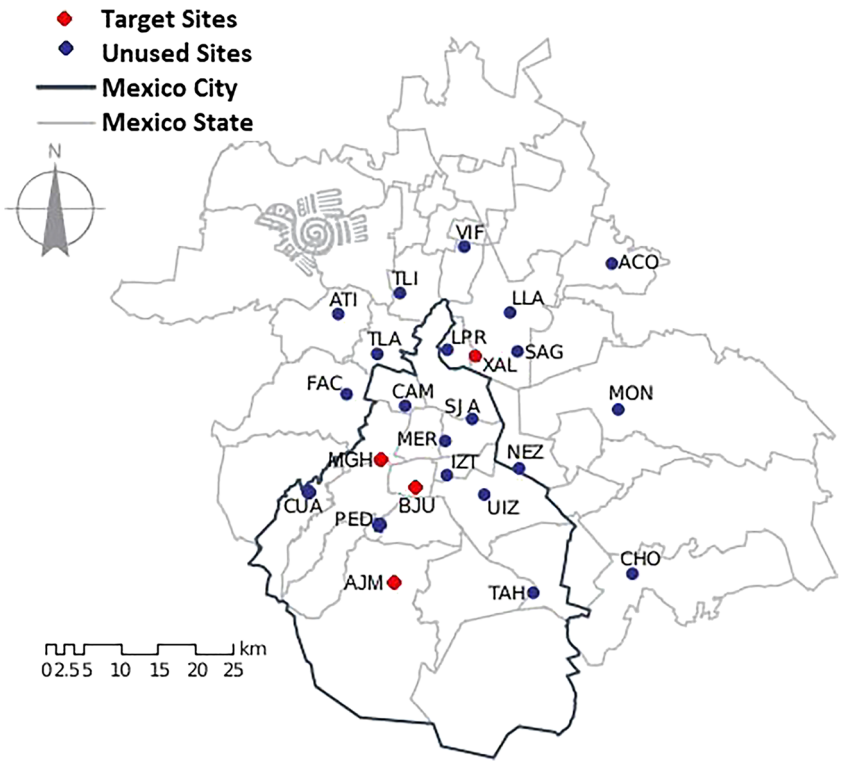

Fig. 6 Geographic location of the monitoring stations where the data was obtained (SEDEMA 2018)

Table 1 Raw data statistics used to train the model

Site	Min(µg/m^3)	Max($\mu g/m^3$)	Mean ($\mu g/m^3$)	$\sigma(\mu g/m^3)$	Missing Values
AJM	1	150	31.48	20.12	472
BJU	1	281	39.44	23.50	187
MGH	2	177	34.13	20.14	124
XAL	2	340	66.38	43.29	213
Total Mean	1.5	237	42.86	26.76	249
Total Absolute	1	340	66.38	43.29	996

LSTM, in the sense that uses the same functions, but they are organized in a different way. Recent studies trend towards simpler recurrent units suggests a need for RNNs with smaller memory footprints and lower training computational load (JHeck and Salem 2017).

To solve the vanishing gradient problem of a standard RNN, GRU use the so called update gate and reset gate. Basically, these are two vectors which decide what information should be passed to the output. The special feature about them is that they can be trained to keep long-term information, without forgetting the information it through time or remove information which is irrelevant to the prediction (Chung et al. 2014), as shown on Fig. 5.

In this contribution, the gated recurrent unit works as follows:

1. The algorithm starts by calculating the *update gate* z_t for the time step t:

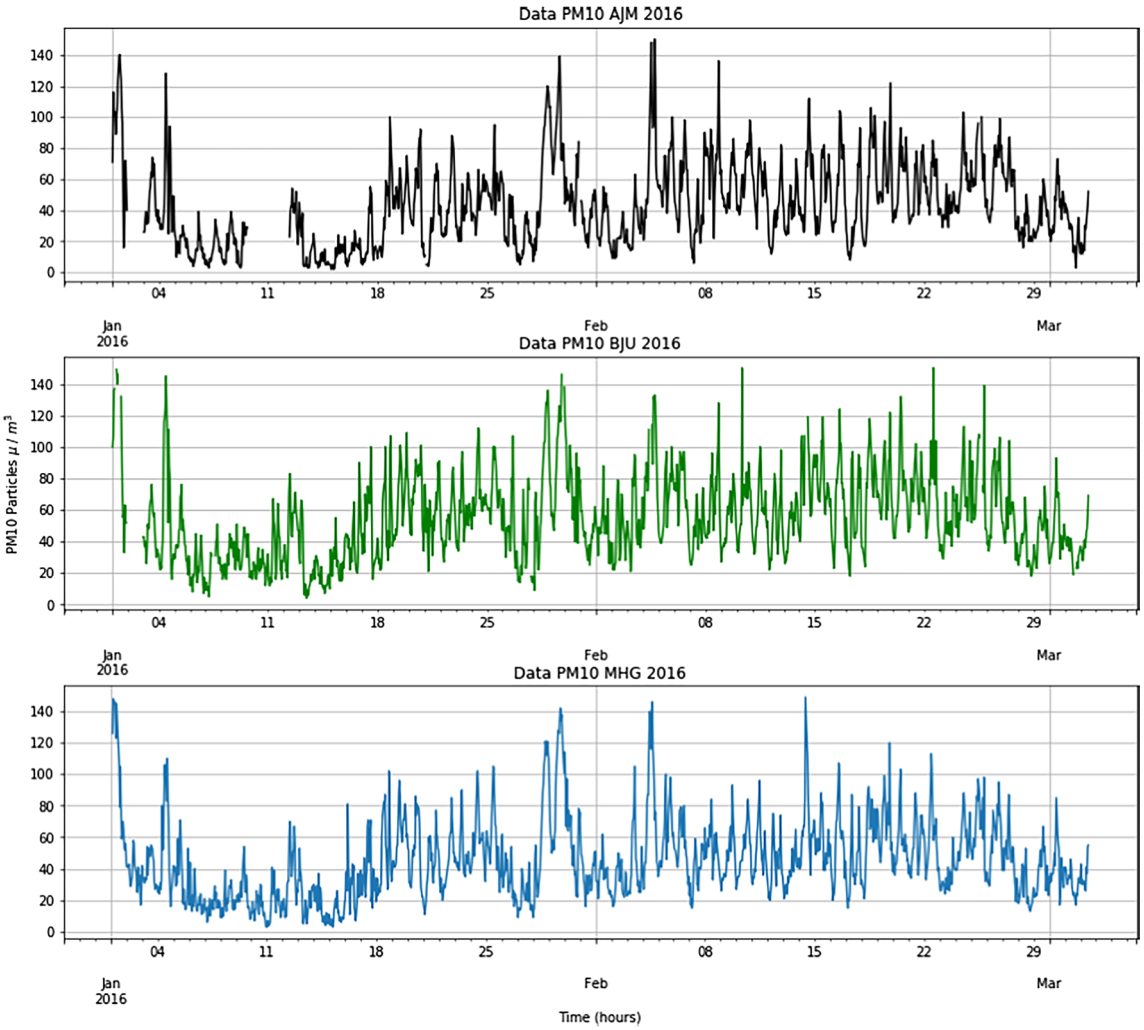

Fig. 7 Raw data

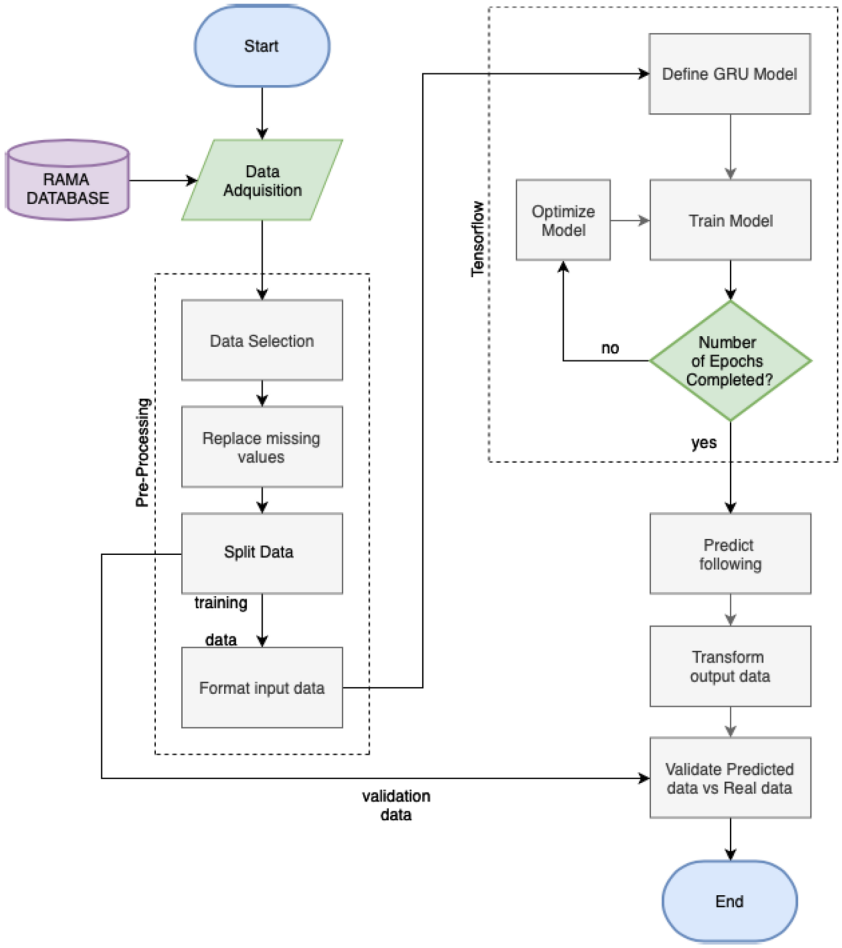

Fig. 8 Methodology flowchart

$$z_t = \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right) \tag{7}$$

When x_t is plugged into the network unit, it is multiplied by its own weight $W^{(z)}$. The same applies for h_{t-1} which holds the information for the previous t-1 units and is multiplied by its own weight $U^{(z)}$. Both results are added together, and a sigmoid activation function is applied normalize the result between 0 and 1.

Table 2 Batches used to train the model

Prediction Target (hours)	Training Steps
12	3000
24	3000
48	3000
120	3000

2. The *reset gate* uses the model to decide how much of the past information is forgotten (erased), to calculate this, Eq. 8 is used:

$$r_t = \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right) \tag{8}$$

3. Here, a new memory content is introduced, which will use the *reset gate* to store the most important information from the past.

$$h't \;\; = tanh\left(Wxt + [r_t{*}Uh_{t-1}]\right) \tag{9}$$

a) Multiplies the input x_t by a weight W and h_{t-1} by a weight U.
b) Calculate the Hadamard (element-wise) product between the reset gate r_t and Uh_t that will determine what to remove from the previous time steps.
c) Sums the results of the steps a) and b)
d) Applies the nonlinear activation function *tanh*.

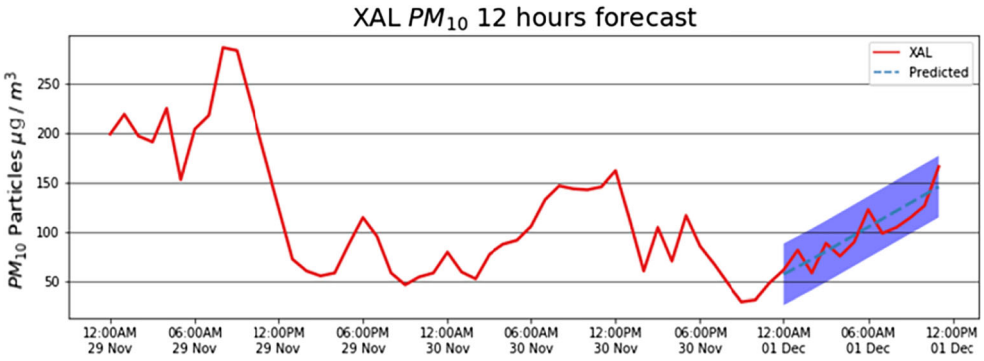

Fig. 9 LSTM prediction of 12 h of PM_{10} for XAL site

4. Finally, it is necessary to calculate the vector h_t that holds the information of the current unit and passes it down to the network, which determines what to collect from the past steps h_{t-1} using the following equation (Eq. 10):

$$h_t = [z_t * h_{t-1}] + [(1 - z_t) * ht] \quad (10)$$

Materials and methods

Materials

One of the main concerns around the world is to protect the health of the population, through the monitoring, prediction and diffusion of the air quality in the cities. The Atmospheric Monitoring System (SIMAT for its acronym in Spanish) is conformed by four subsystems: RAMA, REDMA, REDMET and REDDA.

The Automatic Environmental Monitoring Network (RAMA for its acronym in Spanish) use equipment to measure sulfur dioxide, carbon monoxide, nitrogen dioxide, ozone, PM10 and PM2.5. It has 29 monitoring stations in Mexico City (Fig. 6) (Nom 2005) and has a laboratory for maintenance and calibration of monitoring equipment (SEDEMA 2018).

The minimum value, maximum value, mean value, standard deviation and amount of missing values for each site are shown in Table 1. The mentioned table shows how the site "XAL" have greater values of concentration of PM_{10} particles, which implies a greater standard deviation compared with the remaining stations.

The time series used to train and test this model are PM_{10} particle pollution measures, (concentration, in $\mu g/m^3$) vs time (hours).

The Fig. 7 shows the first 1500 hours of each site. As it can be seen how the top graph (Data PM_{10} AJM 2016) there are some missing values, which may skew the training of the model. For this reason, the raw data must be filtered accordingly.

Methodology

The following methodology was followed for the PM10 airborne particle forecasting using as training data the sites aforementioned (Fig. 8):

1. **Data acquisition:** The first step is to get the required raw data from RAMA (SEDEMA 2018).
2. **Data Selection:** This part consists of selecting the data from the target sites and replace the outliers. For this model, the valid range will be from 0 $\mu g/m^3$ to 150 $\mu g/m^3$.
3. **Fill missing values:** Then, there will be the previous missing values plus the ones that were removed in the previous step. In this case, a simple regression is used to fill those missing values.
4. **Training and Validation data:** The data sets need to be separated into training and validation sets, since using the same data points for both training and validation may skew the model, giving a false indication of the real accuracy of the model. For this model, the data of a year will

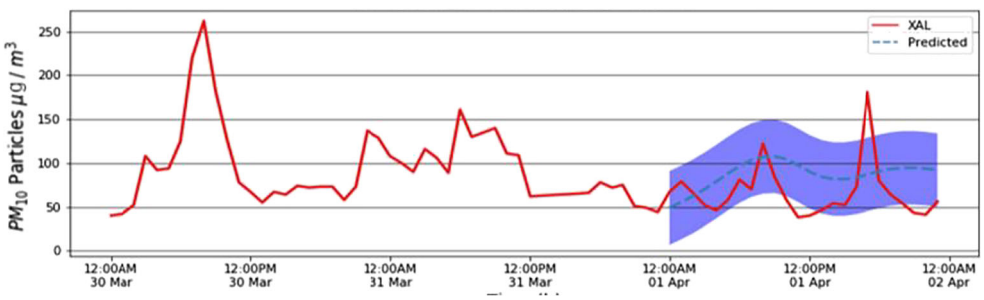

Fig. 10 LSTM prediction of 24 h of PM_{10} for XAL site

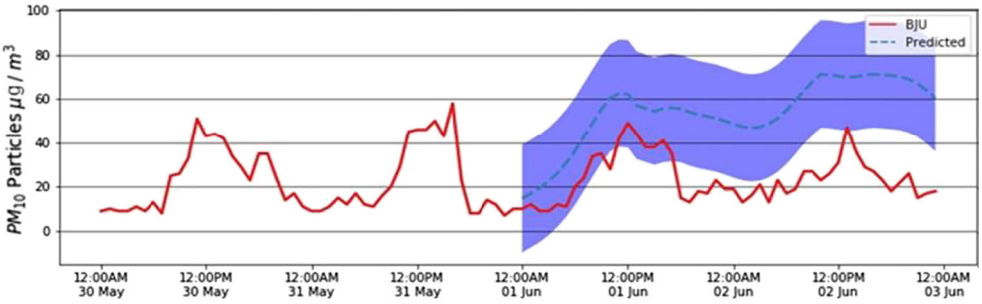

Fig. 11 LSTM prediction of 48 h of PM_{10} for BJU site

be divided into 12 sets (each month) and each of them will be used as a batch of training data.

5. **Format input data:** Before training the model, it is necessary to generate the batches with the proper format that will be used to train this model. The input data must be a 1 dimensional vector with and it must be normalized in a range of [0,1]. To train the model, four targets will be predicted, training the data for 3000 steps as shown on Table 2. Where the prediction target is the following data in hours. In this case, it was found that 3000 steps were the optimum value given that an asymptote at around that value was found and using more than 3000 steps will not give a better performance for this particular application.
6. **Model Construction:** In this section, the model parameters are set. For this work, a GRU and LSTM models were used, the used activation function was a hyperbolic tangent function and a sigmoid function for the gate activation, as show the Figs. 4 and 5. The output layer will be a fully connected layer of 100 units in all cases, to preserve the input length. The learning rate 0.01, and the batch size 1, due to the configuration of the model, each iteration, a new random batch will be selected, therefore 4000 batches will be used to train the model for each month.
7. **Model Training:** After setting all the parameters, the next step is to train the model.
8. **Model Optimization:** While the model is being trained, an observer that evaluates the loss and adjust the weights of the neurons if needed. For this model, the AdamOptimizer (Diederik et al. 2014) was used.
9. **Value Forecasting:** After the model is trained, a value is forecasted. In this work, 12, 24. 48 and 120 h forecasting is performed using the methodology described in this section.
10. **Transform output data:** The prediction is made in a [0,1] scale, so it is necessary to make an inverse transform to the original range [0,150], so the prediction is equivalent to the real PM_{10} particle concentration.
11. **Compare prediction against real data:** The last task to perform is to validate the model, since the further steps of the time series are known, it is possible to compare against the real data and calculate the accuracy of the prediction.

Results

Two different models were tested and analyzed, LSTM and GRU. Each of the two models were trained for the 12 months of 2016, using batches of 12, 24, 48 and 120 hours for each site mentioned in Table 1 as stated in the Methodology section.

As a result, 384 predictions were completed having a good overall performance. The results and analysis are presented in

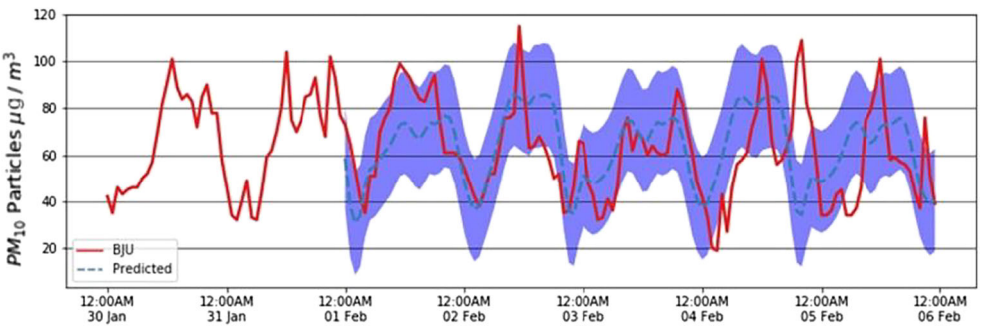

Fig. 12 LSTM prediction of 120 h of PM_{10} for BJU site

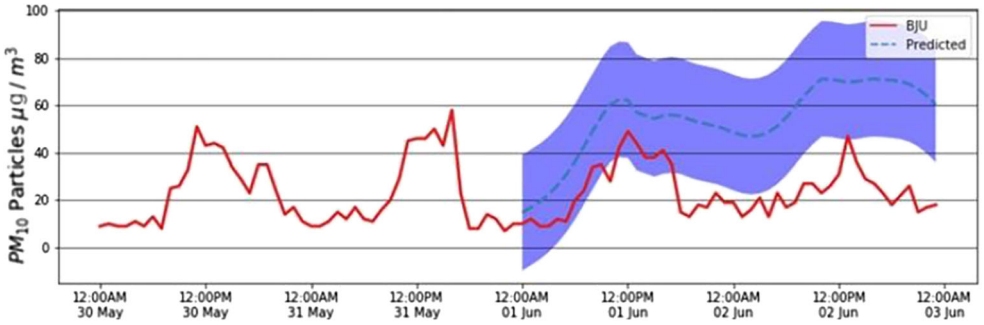

Fig. 13 GRU prediction of 48 h of PM_{10} for BJU site

the next sub-sections, firstly the LSTM results followed by the GRU results and finally a comparison between both model results.

Long short term memory

Figures 9, 10, 11 and 12 show different results for a Long-Short Term Memory network (LSTM) for different hour prediction (12, 24, 48 and 120 hours, respectively).

Gated recurrent unit

Using the methodology shown on Fig. 8, a gated recurrent unit network has been trained and the results are shown on Figs. 13, 14, 15 and 16.

Error estimation

The RMSE in LSTM model was low considering the mean standard deviation of 66.38 previously mentioned in Table 1. The best result was for a batch length of 24 hours in the AJM station, with a RMSE of 1.36, the largest RMSE was 75.79 for a batch length of 120 hours in the XAL station.

For GRU model the RMSE was greater in all cases, having as lower RMSE 3.09 for 48h predictions in the AJM station and 107.25 as maximum value for 12h prediction in XAL station.

Tables 3 and 4 show the mean, minimum and maximum value for each station and each length of prediction time.

In Tables 3 and 4, is shown the mean, minimum and maximum root mean square error (RMSE) of each of the test sites for every length of the prediction time.

Figures 17 and 18 shows the mean RMSE and the variance in each of the stations. They show that in both LSTM and GRU models, AJM and MGH stations had better results than BJU and MGH stations, which implies that the training data for AJM and MGH was better. However, in general Gated Recurrent Unit shows a smaller mean, which seems to indicate that the prediction is slightly better. Also, it is noteworthy to mention that the prediction capabilities of the models proposed in this article, depends largely upon the number of invalid data and there is a relationship between the standard deviation of the input data and the root mean squared error of the prediction.

Figures 19 and 20 shows the mean RMSE and the variance for the different length prediction targets (12, 24, 48 and 120 hours). Where 48 and 120 hour predictions show more consistence in contrast to shorter predictions. 12 hour predictions showed the lowest mean RMSE but also had larger variance.

In Figs. 19 and 20 are shown the RMSE for both LSTM and GRU predictions for the prediction length. LSTM (Fig. 19) shows that with 12 h prediction target the prediction shows a smaller RMSE than prediction with a longer target length. This may indicate that in accordance with the LSTM topology, the shorter prediction length is stored in the memory and for longer prediction lengths, some of the data is reset, thus the network tends to re-train the data, which results in a higher RMSE for 24 and 48 h. This needs to be investigated further.

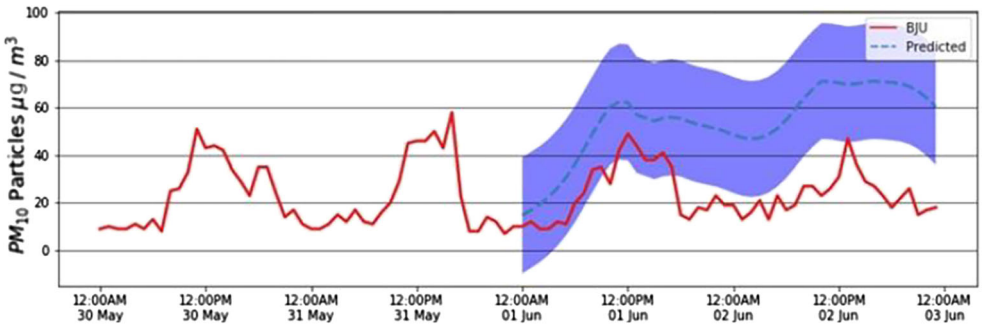

Fig. 14 GRU prediction of 24 h of PM_{10} for BJU site

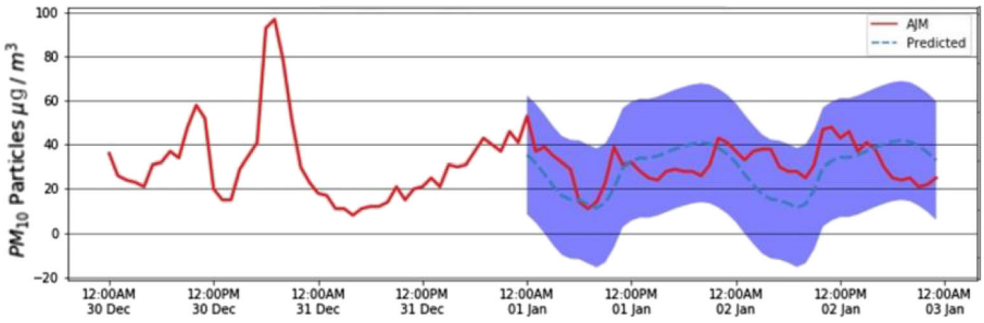

Fig. 15 GRU prediction of 48 h of PM_{10} for AJM site

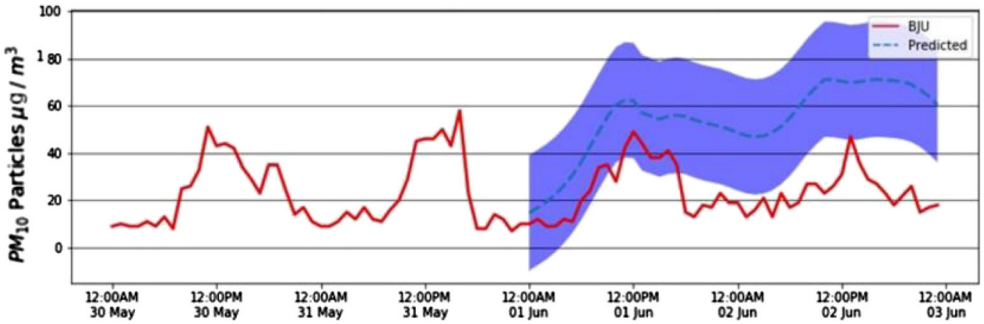

Fig. 16 GRU prediction of 120 h of PM_{10} for BJU site

For GRU, the RMSE for the prediction length seems steady, showing a slightly better RMSE for 120 h prediction length, showing that GRU has a greater capability than LSTM in terms of the length to which the prediction can be carried out.

Figures 21 and 22 show that the monthly prediction had notable variations due to the missing values in the training data. The missing values were replaced by a linear regression, which generated flat signals within the training data sets, increasing the RMSE when forecasting.

For both models GRU and LSTM, July and September had a significant better accuracy than the rest of the months, while May had a lot of variation and greater RMSE (see Figs. 21 and 22). This could result due to the missing values, but January, February and November had better results in LSTM model than GRU model using the same training data.

The results showed good predictions considering the mean standard deviation of 26.76 μ/m3 (see Table 1). The LSTM model had an overall better accuracy than the GRU model, but it was not a significant difference. Hence both models can be used to predict PM10 particle time series.

Conclusions

Two different approaches were implemented using deep neural networks: LSTM and GRU. The methodology proposed in this work show that is feasible to model the highly non-linear behavior with both models with moderate accuracy. Also, several experiments were carried out to demonstrate the robustness of the proposed model for 12, 24, 48 and 120 h in advance, which show

Table 3 LSTM root mean square error in PM_{10} $\mu g/m^3$

Hours	12			24			48			120		
Site	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
AJM	3.64	40.42	**13.30**	1.36	33.69	19.08	8.21	41.40	18.31	14.58	35.86	20.20
BJU	8.38	55.15	18.86	8.03	45.60	20.81	14.31	39.54	24.46	8.11	38.47	21.98
MGH	7.38	28.41	15.06	3.24	28.89	**16.98**	2.03	42.02	**16.81**	2.13	23.19	**16.85**
XAL	6.84	84.51	31.04	3.04	91.11	43.22	46.19	46.19	46.19	26.98	75.79	51.16

Bold values indicate the best result for that particular experiment

Table 4 GRU root mean square error in PM_{10} $\mu g/m^3$

Hours	12			24			48			120		
Site	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
AJM	4.86	31.51	**16.32**	9.91	66.78	20.95	3.09	39.56	**16.67**	14.48	52.47	24.10
BJU	8.64	64.06	20.07	5.11	45.71	21.01	11.23	77.64	26.80	7.36	36.02	23.28
MGH	4.11	37.18	17.85	7.53	24.37	**15.83**	5.43	30.09	18.56	9.92	26.76	**19.47**
XAL	3.77	107.25	46.82	6.44	103.69	47.36	5.59	81.20	44.09	71.38	71.38	71.38

Bold values indicate the best result for that particular experiment

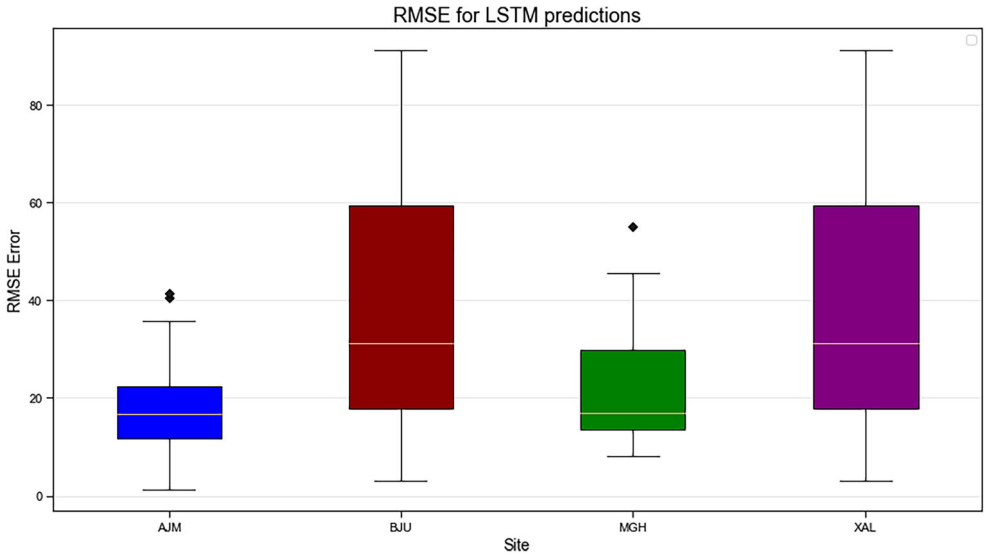

Fig. 17 Boxplot of RMSE for LSTM model predictions by site

that this method may be used for longer periods of time. Furthermore, the predictions were carried out by month, showing that the number of missing data and the standard deviation of the raw data may play an important role in the prediction of PM10. As future work, other methods to pre-process the data may be used. Also, ensemble methods may be used to improve upon the results.

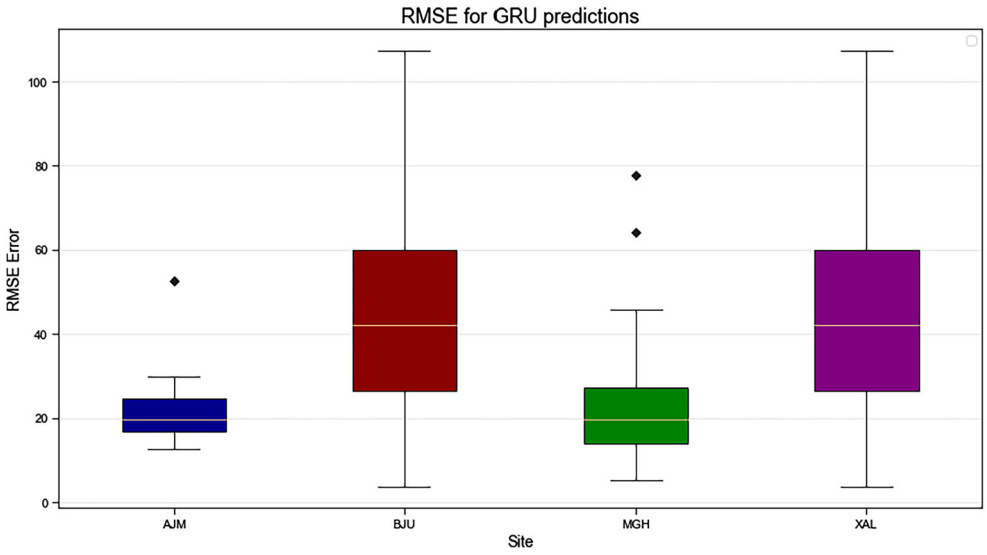

Fig. 18 Boxplot of RMSE for GRU model predictions by site

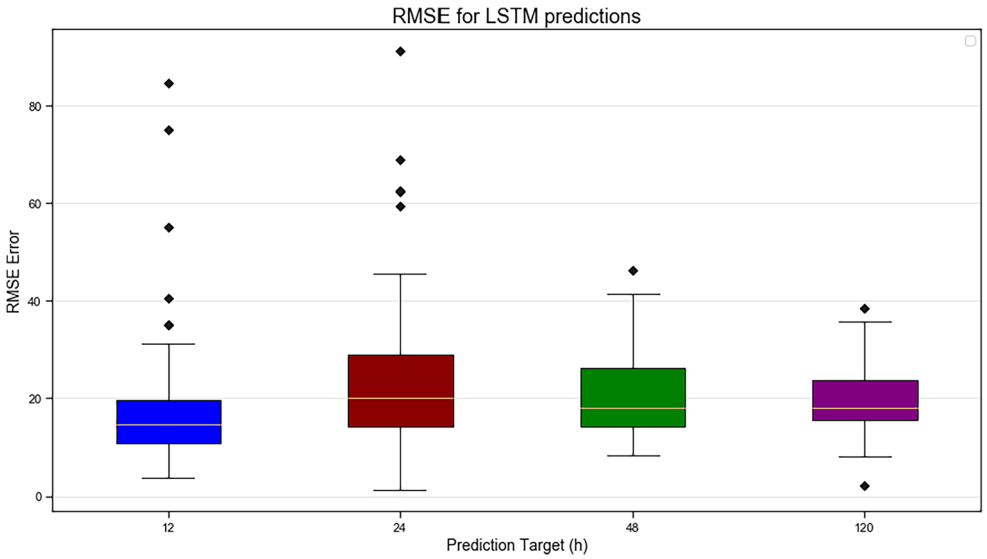

Fig. 19 Boxplot of RMSE for LSTM predictions by prediction target length

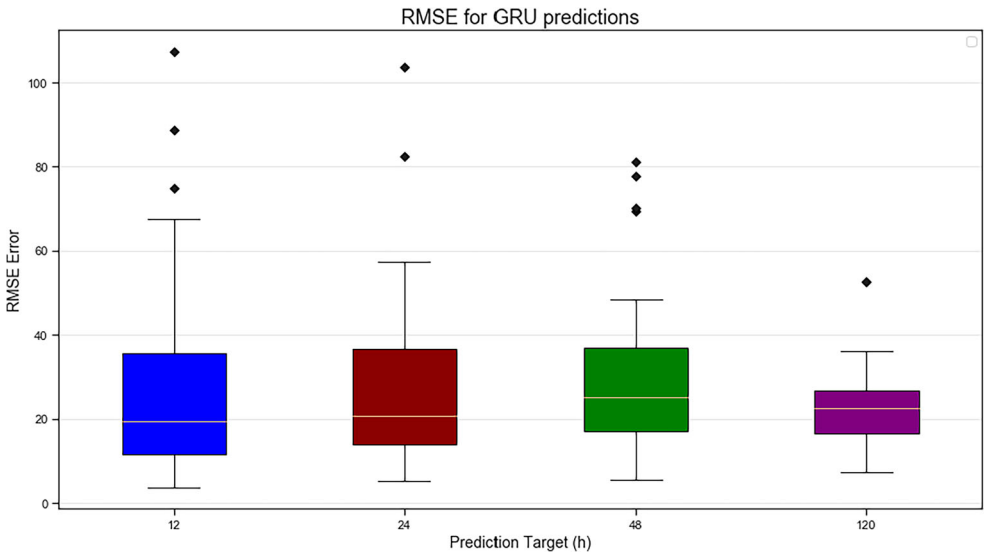

Fig. 20 Boxplot of RMSE for GRU predictions by prediction target length

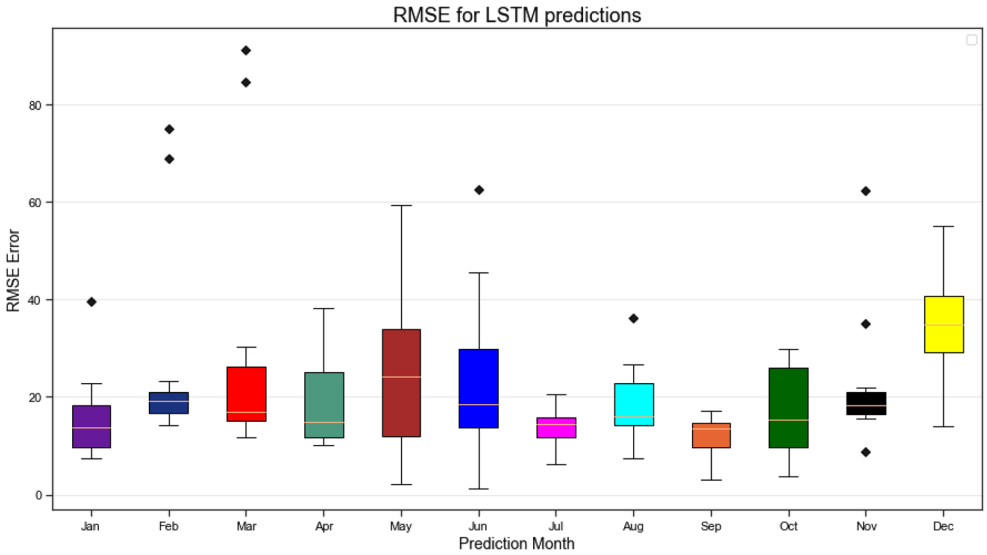

Fig. 21 Boxplot of RMSE for LSTM predictions by month

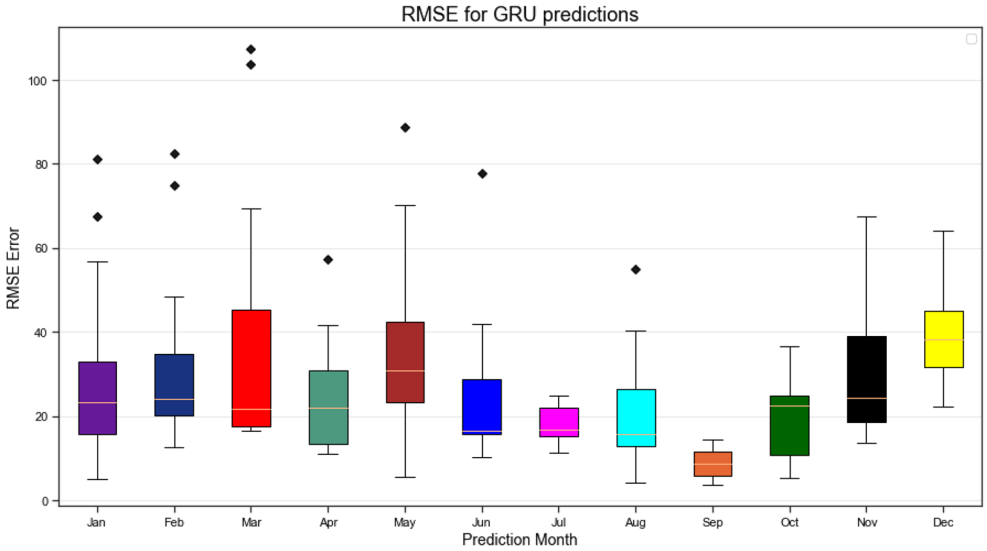

Fig. 22 Boxplot of GRU for LSTM predictions by month

References

Aceves-Fernandez MA et al (2018) Feature extraction of EEG signal upon BCI systems based on steady-state visual evoked potentials using the ant colony optimization algorithm. Discret Dyn Nat Soc: 21. https://doi.org/10.1155/2018/2143873

Aceves-Fernandez MA et al (2015) Design and implementation of ant colony algorithms to enhance airborne pollution models. Int J Environ Sci Toxicol Res 3(2):22–28

Akbarzadeh MA et al (2018) The association between exposure to air pollutants including PM10, PM2. 5, ozone, carbon monoxide, sulfur dioxide, and nitrogen dioxide concentration and the relative risk of developing STEMI: a case-crossover design. Environ Res 161:299–303

Bartzis JG et al (2019). Environmental data treatment to support exposure studies: the statistical behavior for NO2, O3, PM10 and PM2.5 air concentrations in Europe. https://doi.org/10.1016/j.envres.2019.108864

Bos PM et al (2019) Pulmonary toxicity in rats following inhalation exposure to poorly soluble particles: The issue of impaired clearance and the relevance for human health hazard and risk assessment. Regul Toxicol Pharmacol:104498

Cabrera-Hernandez MC, Aceves-Fernandez MA et al (2019) Parameters influencing the optimization process in airborne particles PM10 Using a Neuro-Fuzzy Algorithm Optimized with Bacteria Foraging (BFOA). Int J Intellig Sci 9:67–91. https://doi.org/10.4236/ijis.2019.93005

Caudillo L et al (2020) Nanoparticle size distributions in Mexico city. Atmos Pollut Res 11(1):78–84

Cho K, Van Merriënboer B et al (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*

Chung J, Gulcehre C et al (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv: 1412.3555*

Diederik P et al (2014) A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego

Domínguez-Guevara R et al (2019) Propuesta de red neuronal convolutiva para la predicción de partículas contaminantes PM10 Convolutional Neural Network Proposal for Particulate Matter PM10 Prediction. Res Comput Sci 148(7):51–63

Franceschi F et al (2018) Discovering relationships and forecasting PM 10 and PM 2.5 concentrations in Bogotá, Colombia, using Artificial Neural Networks, Principal Component Analysis, and k-means clustering. Atmos Pollut Res. https://doi.org/10.1016/j.apr.2018.02.006

Goglio P et al (2019) Advances and challenges of life cycle assessment (LCA) of greenhouse gas removal technologies to fight climate changes. J Clean Prod:118896

Grivas G et al (2006) Artificial neural network models for prediction of PM10 hourly concentrations, in the Greater Area of Athens, Greece. Atmos Environ 40:1216–1229. https://doi.org/10.1016/j.atmosenv.2005.10.036

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

JHeck JC, Salem FM (2017) Simplified minimal gated unit variations for recurrent neural networks. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)* (pp. 1593-1596). IEEE

Längkvist M et al (2014) A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recogn Lett 42:11–24

Montañez JAR, Fernandez MAA, Arriaga ST, Arreguin JMR, Calderon GAS (2019, September). Evaluation of a recurrent neuralnetwork LSTM for the detection of exceedances of particles PM10. In 2019 16th International Conference on Electrical Engineering, Computing Scienceand Automatic Control (CCE) (pp. 1–6). IEEE

Nom (2005) (norma oficial mexicana nom-025-ssa1–1993). salud ambiental. criterios para evaluar la calidad del aire ambiente, con respecto a material particulado. última modificación, septiembre del 2005. Secretaría de Salud, México (Health Secretariat)

Oh HR et al (2015) Long-range transport of air pollutants originating in China: a possible major cause of multi-day high-PM10 episodes during cold season in Seoul, Korea. Atmos Environ 109:23–30

Ordonez de León B, Aceves-Fernandez MA et al (2019) An improved particle swarm optimization (PSO): method to enhance modeling of airborne particulate matter (PM10). Evol Syst. https://doi.org/10.1007/s12530-019-09263-y

Schraufnagel DE et al (2019) Air pollution and noncommunicablediseases: a review by the Forum of International Respiratory Societies' Environmental Committee, Part 2: air pollution and organ systems. Chest 155(2):417–426

SEDEMA (2018). Red automática de monitoreo ambiental. Available at http://www.aire.cdmx.gob.mx/, Accessed July 2018

Skrzypski J, Kami K, Jach-Szakiel E, Kami W (2009) Application of artificial neural networks for classification and prediction of air quality classes. WIT Trans Ecol Environ 127:219–228

Yetilmezsoy K. (2012). A prognostic approach based on fuzzy-logic methodology to forecast PM10 levels in Khaldiya residential area, Kuwait. Aerosol Air Qual Res. 12. https://doi.org/10.4209/aaqr.2012.07.0163

Zhou G-B, Wu J, Zhang C-L, Zhou Z-H. Minimal gated unit for recurrent neural networks. Int J AutomComput 13(3):226–234