

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,100

Open access books available

149,000

International authors and editors

185M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

Perspective Chapter: Airborne Pollution (PM_{2.5}) Forecasting Using Long Short-Term Memory Deep Recurrent Neural Network Optimized by Gaussian Process

Marco Antonio Olguin-Sanchez,

Marco Antonio Aceves-Fernández,

Jesus Carlos Pedraza-Ortega and Juan Manuel Ramos-Arreguín

Abstract

Forecasting air pollution is a challenging problem today that requires special attention in large cities since they are home to millions of people who are at risk of respiratory diseases every day. At the same time, there has been exponential growth in the research and application of deep learning, which is useful to treat temporary data such as pollution levels, leaving aside the physical and chemical characteristics of the particles and only focusing on predicting the next levels of contamination. This work seeks to contribute to society by presenting a useful way to optimize recurrent neural networks of the short and long-term memory type through a statistical process (Gaussian processes) for the correct optimization of the processes.

Keywords: deep learning, gaussian process, optimization, recurrent neural network, long-short term memory, airborne pollution

1. Introduction

Recurrent neural networks (RNN), especially Long Short Term Memory (LSTM), have proved their efficiency in working on time-dependent values by (as its names indicate) the use of memory (sequences) gives enough information to the network to work properly finding patterns and trends in the values, which are not so obvious at first glance. Also, the gaussian processes are a useful statistical technique that allows the hyperparameters of the network since it has shown that the processing time is reduced and, at the same time, the accuracy may be improved [1]. Afterward, there is airborne pollution, which is a complex system that affects billions of people worldwide, especially in a metropolis such as Hotan, China, Shanghai, China, Ghaziabad, India, or in the case of this study, Mexico City, Mexico. Also, there we have a lot of

types of particles interacting with each other in chemical, biological, and physical ways. The pollutants that are monitored by the SEDEMA's network in the City of Mexico are nitrogen dioxide (NO_2), Ozone (O_3), sulfur dioxide (SO_2) and particulate matter ($PM_{2.5}$, and PM_{10}). Hence, we propose the use of an RNN-LSTM and optimizing its hyperparameters using Gaussian processes to increase the accuracy in the forecast of airborne pollution instead of the use of the current method.

2. Literature review

2.1 Airborne pollution

Air pollution has been a problem that has been increasing in recent decades mainly in large cities, bringing with it respiratory diseases [2, 3]. This contamination is accompanied by the same pollutant particles that have a useful life depending on physical parameters (size and shape) and their chemical composition. Mexico City has been studied for decades [4] due to its high levels of pollution that affect more than 20 million people. The sites used in this work are the following: Northeast (Gustavo A. Madero—GAM, FES Aragón—FAR, Xalostoc—XAL), Northwest (Tlalnepantla—TLA,), Center (Hospital General de México—HGM, Merced—MER), Southeast (Nezahualcóyotl—NEZ, Santiago Acahualtepec—SAC) and Southwest (Ajusco Medio—AJM, Pedregal—PED, Santa Fe, SFE). The map of the monitoring sites is shown in **Figure 1**.

2.1.1 Multiple imputation by chained equations (MICE)

Dealing with the missing data problem often leads to two general approaches for imputing multivariate data: Joint modeling (JM) and fully conditional specification

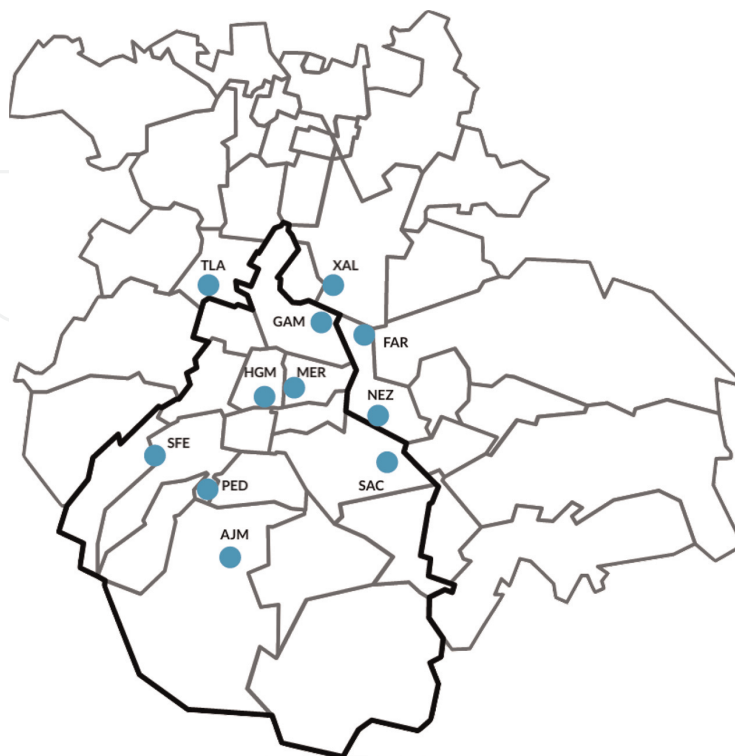


Figure 1.
Location of the monitoring sites in Mexico City.

(FCS), also known as multivariate imputation by chained equations (MICE) [5]. It is known that is a JM-type problem when we must specify a multivariate distribution for the missing data and obtain the imputation of its conditional distributions through Markov Monte Carlo chains (MCMC) techniques. On the other hand, it is FCS, which specifies the multivariate imputation model on a variable-by-variable basis using a set of conditional densities, one for each incomplete variable. The imputation starts by iterating over the conditional densities, usually, a low number of iterations is enough. In order to explain the model, let us use the following notation [5]: Let Y_j with ($j = 1, \dots, p$) be one of p incomplete variables and $Y = (Y_1, \dots, Y_p)$. The observed and missing parts of Y_j are denoted by Y_j^{obs} and Y_j^{mis} , respectively, then $Y^{obs} = (Y_1^{obs}, \dots, Y_p^{obs})$ and $Y^{mis} = (Y_1^{mis}, \dots, Y_p^{mis})$, these are the observed and missing data respectively in Y . The number of imputations is $m \geq 1$. The h -th imputed data set is denoted as $Y^{(h)}$ where $h = 1, \dots, m$. Now let $Y_{-j} = (Y_1, \dots, Y_{j-1}, Y_{j+1}, \dots, Y_p)$ denote the collection of the $p - 1$ variables in Y except Y_j . Finally, let Q denote the quantity of scientific interest. The mice algorithm has three main steps: imputation, analysis, and pooling. The analysis starts with an incomplete data set Y_{obs} . The second step is to compute Q on each imputed data set, here the model is applied to $Y^{(1)}, \dots, Y^{(m)}$ in the general identical. Finally, the third step is to pool the m estimates $Q^{(1)}, \dots, Q^{(m)}$ into one estimate \bar{Q} and estimate its variance.

2.1.2 Recurrent neural networks

Recurring neural networks (better known as RNN) can be used for any type of data. In practical applications, the use of symbolic values is more common. In a recurrent neural network, there is a one-to-one correspondence between the layers of the network and specific positions in the sequence. The position in the sequence is also known as its timestamp. Finally, RNNs are complete Turing, which means that this type of network can simulate any algorithm with sufficient data and computational resources [6]. A representation of this kind of network is shown in **Figure 2**.

2.1.3 Long-short term memory (LSTM)

To represent the hidden states of the k th hidden states (layer) the notation $h_t^{-(k)}$ is used and to simplify the notation it will be assumed that the input layer \bar{x}_t can be denoted by $h_t^{-(0)}$ (this layer is not hidden) [7]. To obtain good results, a hidden vector of dimension p must also be included, which will be denoted by $c_t^{(\bar{k})}$ and refers to the state of the cell. The state of the trap can be observed as the long-term memory within the network. The matrix that updates the values is denoted by $W^{(k)}$ and is used to permute the column vectors $[h_t^{-(k-1)}, h_{t-1}^{-(k)}]^T$. The matrix that is obtained always results in dimensions $4p \times 2p$. A $2p$ size vector is then premultiplied by the $W^{(k)}$ matrix resulting in a $4p$ vector. Now to find the updates we have the following; for setting up intermediates Eq. 1, for selectively forget and add to long-term memory eq. 2, for selectively leak long-term memory to hidden state (Eq. 3).

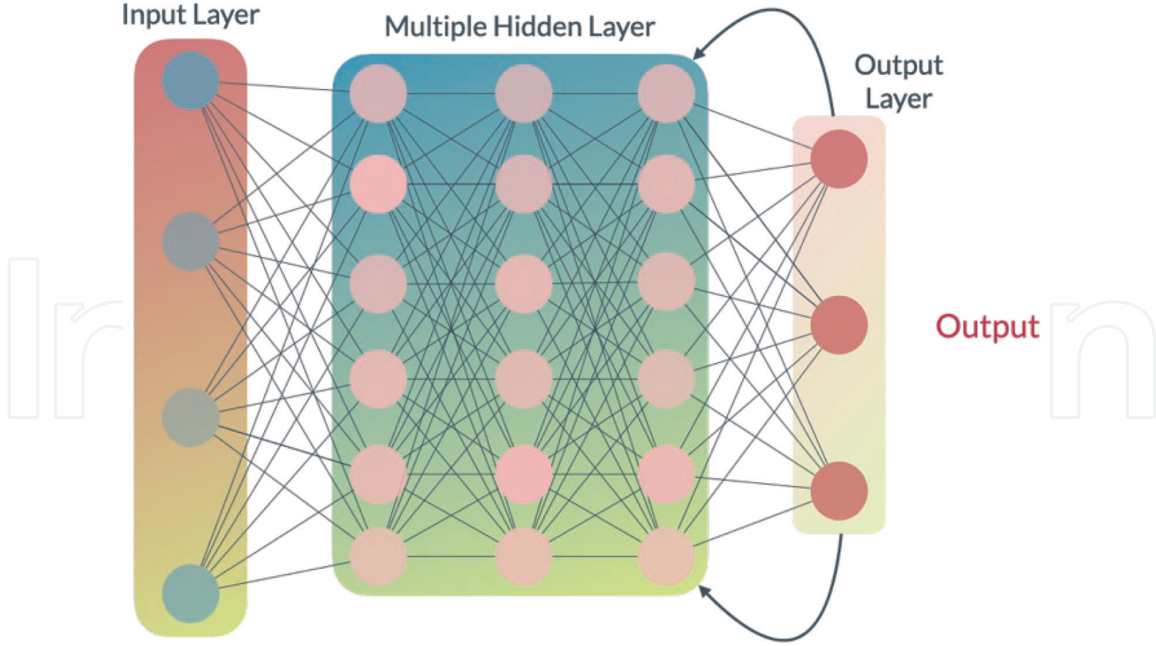


Figure 2.
Graphic representation of a recurrent neural network.

$$\begin{matrix} \text{InputGate} : \\ \text{ForgetGate} : \\ \text{OutputGate} : \\ \text{NewC - State} : \end{matrix} \begin{bmatrix} \bar{i} \\ \bar{f} \\ \bar{o} \\ \bar{c} \end{bmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \text{sigm} \end{pmatrix} W^{(k)} \begin{bmatrix} \bar{h}_t^{(k-1)} \\ \bar{h}_{t-1}^k \end{bmatrix} \quad (1)$$

$$c_t^{(\bar{k})} = \bar{f} \odot c_{t-1}^{(\bar{k})} + \bar{i} \odot \bar{c} \quad (2)$$

$$h_t^{(\bar{k})} = \bar{o} \odot \tanh(c_t^{(\bar{k})}) \quad (3)$$

Additionally, clarify that LSTM is an algorithm that belongs to recurrent neural networks or RNN [8]. The RNN's refer to neural networks that take their previous state as input, this means that the neural network will have two inputs, the new information entered into the network and its previous state, which is shown in **Figure 2**. With this model we can have short-term memory in the neural network [9]. These neural networks have applications in sequential predictions, that is, predictions that depend on a temporal variable.

2.2 Bayesian optimization using Gaussian processes

2.2.1 Multidimensional Gaussian distribution

To talk about Gaussian processes, we must first define a multivariable Gaussian distribution in several dimensions. Formally, this distribution is expressed as in Eq. 4:

$$p(x) = \frac{1}{(2\pi)^{D/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad (4)$$

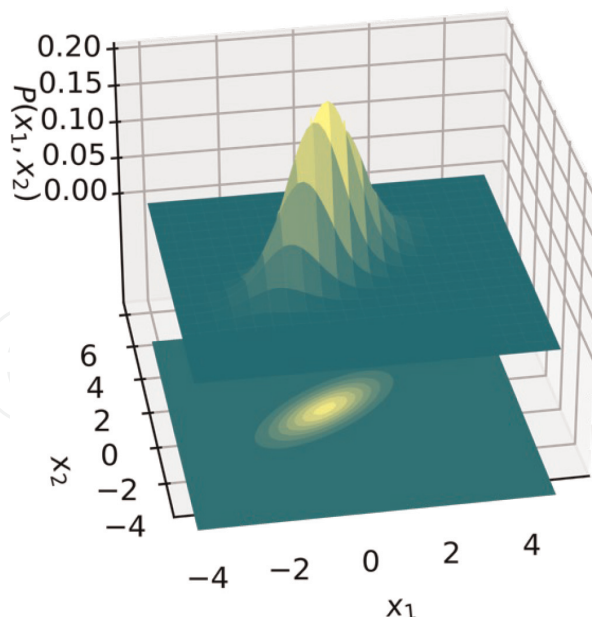


Figure 3.
Graphic example of a multidimensional gaussian distribution.

Where D is the number of dimensions, x is the variables, μ is the average vector, Σ is the covariance matrix. Gaussian processes try to model a function f given a set of points [10]. Traditional nonlinear regression machine learning methods usually give a function that they think best fits these observations. But, there may be more than one function that fits the observations equally well. When we have more observation points, we use our posterior-anterior as our anterior, we use these new observations to update our posterior. This is the Gaussian process. A Gaussian process is a probability distribution over possible functions that fit a set of points. Because we have the probability distribution over all possible functions, we can calculate the means as the function and calculate the variance to show how confident we are when we make predictions using the function (**Figure 3**).

2.2.2 Gaussian process

Because we have the probability distribution over all possible functions, we can calculate the means as the function and calculate the variance to show how confident when predictions are made using the function as demonstrated by Wang [10], we must take into account that:

- I.The (later) functions are updated with new observations.
- II.The mean calculated by the posterior distribution of the possible functions is the function used for the regression.

The function is modeled by a multivariable Gaussian of the form shown in Eq. 5:

$$P(f|X) = N(f|\mu, K) \quad (5)$$

Where $X = [x, \dots, x_n]$, $f =, \dots, f(x_n)$, $\mu =, \dots, m(x_n)$, $K_{ij} = k(x_i, x_j)$. Being X the points of the observed data, m represents the average function and K represents a definite positive kernel.

3. Materials and methods

3.1 Materials

The data used to train the model were obtained from the Atmospheric Monitoring System (SIMAT for its acronym in Spanish) database is conformed of four subsystems: RAMA, REDMA, REDMET, and REDDA, all are given by its website but we just focus on the Automatic Environmental Monitoring Network (RAMA for its acronym in Spanish).

3.2 Methodology

3.2.1 Data acquisition & preprocessing data

First, the database on air quality, RAMA (SEDEMA, 2021) [11] must be downloaded, which is public. This file (dataset) contains values captured by all stations capable of monitoring $PM_{2.5}$. In case the dataset contains more variables in addition to the one already mentioned, a preprocessing process must be carried out. First, the values of interest ($PM_{2.5}$ and air direction) should be classified, excluding any other. Once the data has been classified, it will be necessary to determine if there are missing data and in which cases it is convenient to impute because if the amount of data to be imputed exceeds 40% of the total data, it is advisable not to use that station since there is a loss very large data and a case of over-learning or data that does not reflect reality could be presented. To impute the missing data, the MICE algorithm will be used and once the algorithm has been applied, a new dataset will have to be generated with the imputed data (complete). Because the RNN-LSTM works by taking a tensor as input and already having an absent dataset of missing data, now it will be necessary to divide this data into three different datasets which would serve for training, testing, and data validation. To conclude with the preprocessing, the data will be normalized and later converted into tensors. To normalize the data, the min-max normalization will be used, which takes the minimum value of the data as “zero” and the maximum value as “one” and it is based on these that the normalization is performed. For tensors, they must take into account the batch value, which in turn takes values of 2^n with $n \in \mathbb{N}$. The value of the sequence to use as a parameter should also be considered when creating the tensors.

3.2.2 Instantiate LSTM and optimize the model

To begin with the training and optimization of the network, we are going to start the network with values of the hyperparameters selected at random, this is only to make the network start and work since later the values of each selected hyperparameter will be rewritten in optimization until the optimal point is found within the search space that is established. By having the data imputed, divided,

normalized, and transformed into tensors now, using the training dataset, as its name implies, we will begin the process of training the network, which will go hand in hand with the number of iterations (points) that have been assigned to the optimization process. In each iteration, an adjustment will be made in some of the hyperparameters, and saving the score of each model to end up with the best one.

3.2.3 Model evaluation

Having already a trained and optimized model, we can now determine the efficiency of the model by calculating the RMSE that the model has by comparing the predicted data against the real data. To generate predictions, we are going to use the evaluation set (**Figure 4**).

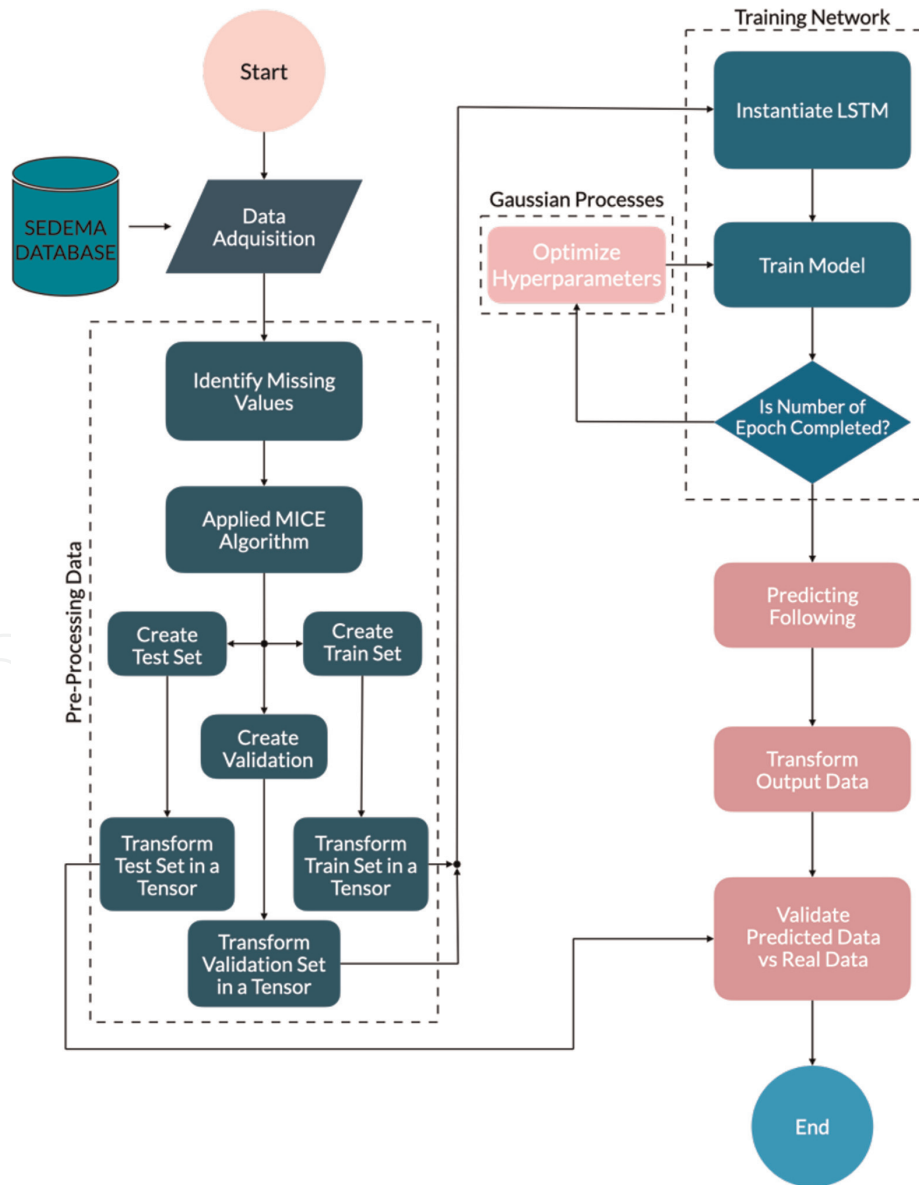


Figure 4.
 Proposed methodology.

4. Results and discussion

There was a considerable amount of missing values all across the data. After applying the MICE algorithm the data full fill. In order to confirm that the imputation was successfully checked that the distribution of the data did not change as is shown in **Figures 5** and **6**. These processes will be repeated for each station. This process was repeated for all the stations which were selected getting similar results in all cases. Once the preprocessing was finished, the training was started and by optimizing the model, a search space was stabilized for each hyperparameter of the RNN-LSTM considered. Now, with the model ready and tested, the results of the optimization process can be seen within the network. As is shown in **Figures 7** and **8**, for the LSTM optimized by GP, the loss function during training decreases rapidly in each epoch until it converges and the change between epochs is no longer so noticeable compared with simple LSTM (**Figures 9** and **10**). In both cases when the converges are reached, it means that the network has already stopped learning. Finally, the validation set is used to estimate the skills of the network obtained in its training for the prediction of $PM_{2.5}$ levels. This is shown for the LSTM optimized by GP in **Figures 11** and **12**, and

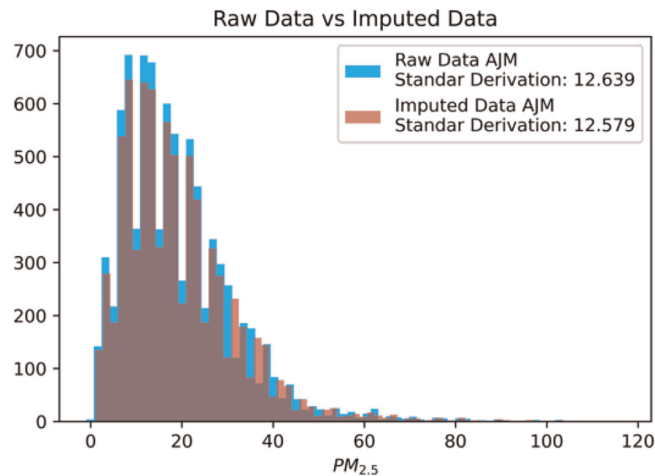


Figure 5.
Data distribution of AJM station before and after being imputed.

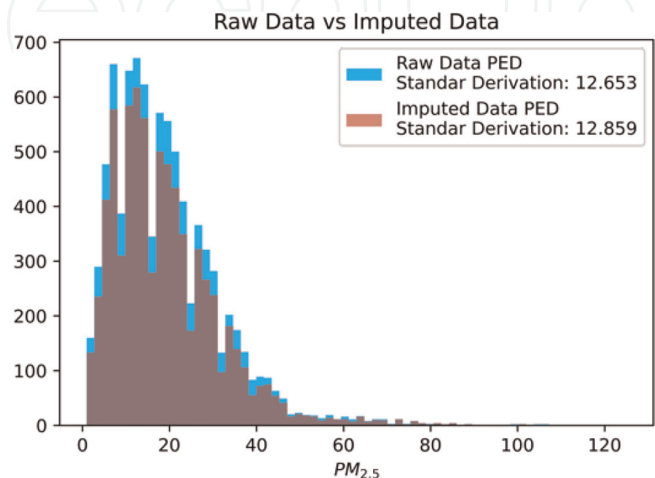


Figure 6.
Data distribution of PED station before and after being imputed.

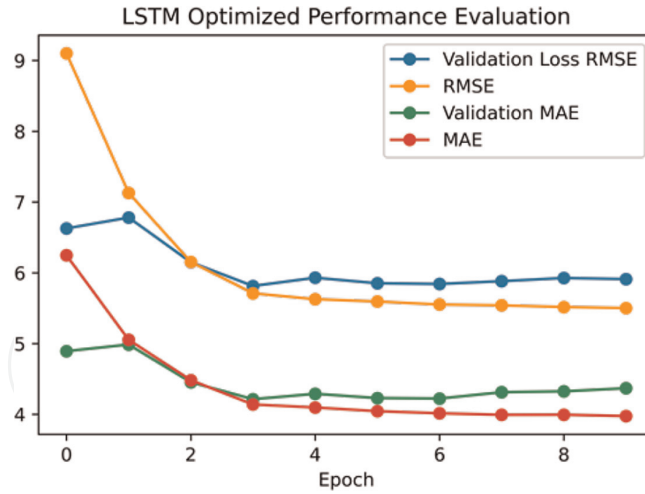


Figure 7.
Metrics obtained with LSTM model optimized by Gaussian process for the AJM station.

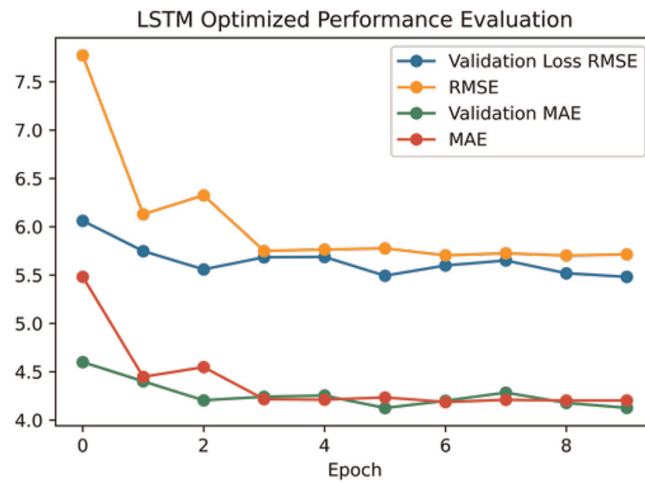


Figure 8.
Metrics obtained with LSTM model optimized by Gaussian process for the PED station.

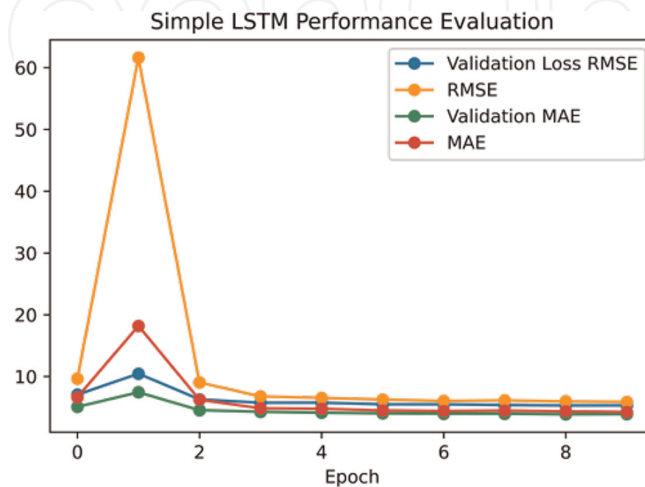


Figure 9.
Metrics obtained with simple LSTM model for the AJM station.

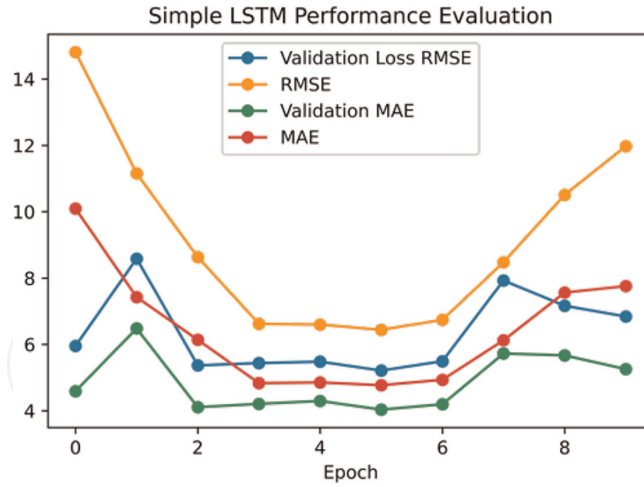


Figure 10
Metrics obtained with simple LSTM model for the PED station.

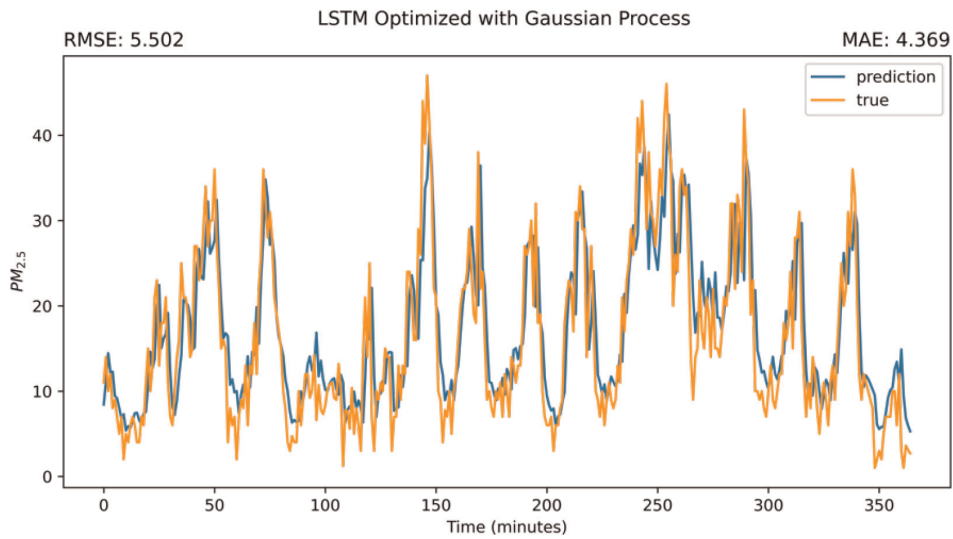


Figure 11.
LSTM optimized by gaussian process, validation forecast for AJM station.

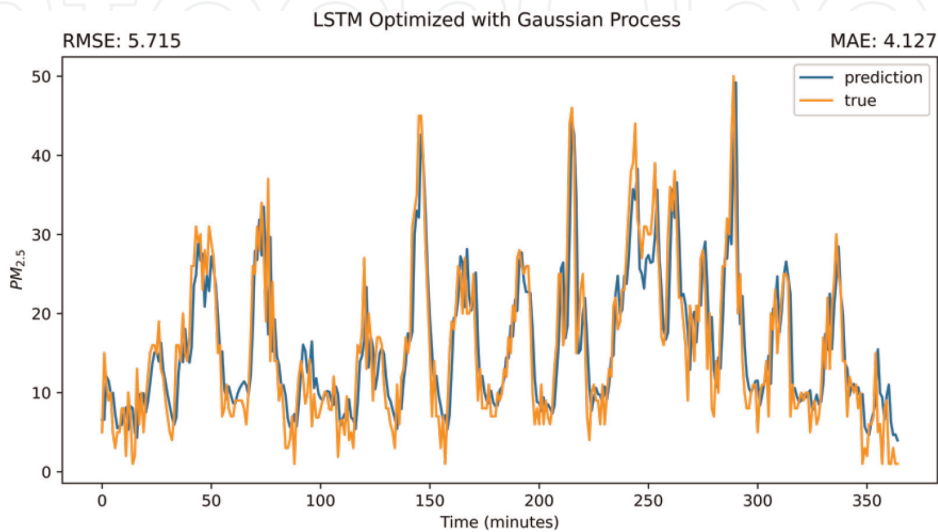


Figure 12.
LSTM optimized by gaussian process, validation forecast for PED station.

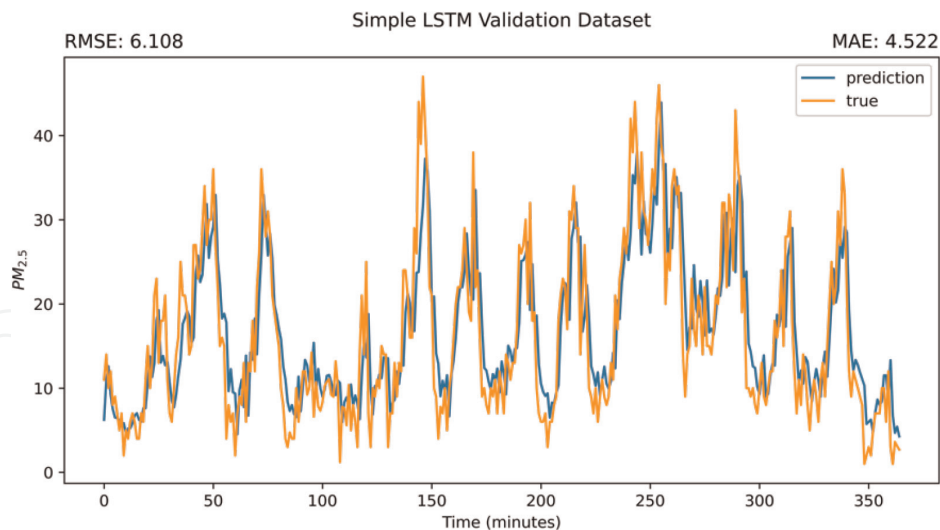


Figure 13.
Simple LSTM, validation forecast for PED station.

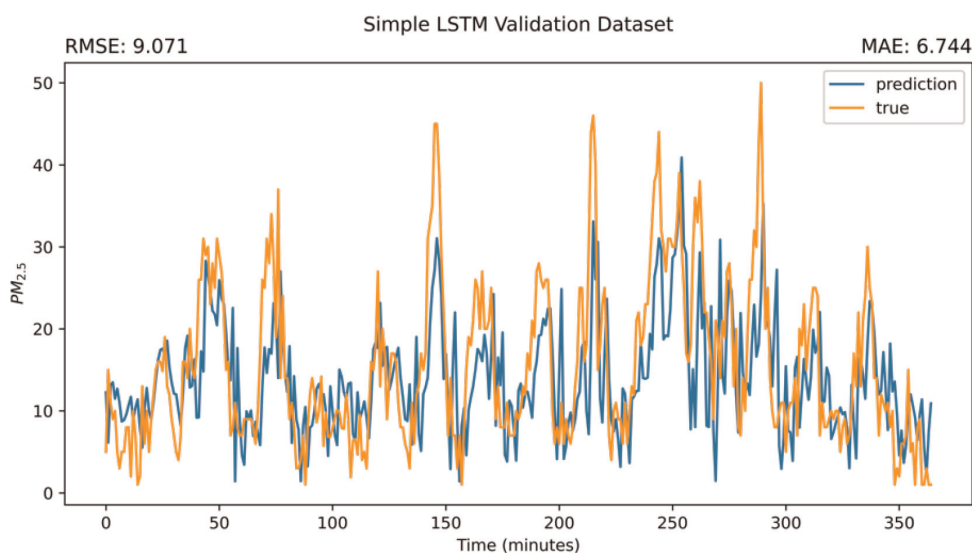


Figure 14.
Simple LSTM, validation forecast for PED station.

for the simple LSTM the results are shown in **Figures 13** and **14**. The results show are from AJM and PED stations.

5. Conclusion

The potential of using Gaussian Processes to improve the hyperparameters tuning is based on a strong mathematical model compared with other methods of hyperparameter tuning; this gives more confidence when looking for an optimized deep learning model. On the other hand, to use this method requires more time to compute the Bayesian search and it may be a thing to consider.

About the project, it does not pretend to end the airborne pollution problem, but it introduces a useful tool for the government and the citizen in order to prevent and plane his day because the model can make predictions along more than three hundred hours with high accuracy, so this gives 12 days to the corresponding users to avoid or prevent the contamination levels.

Conflict of interest

The authors declare no conflict of interest

Consent for publication


Not applicable

Author details

Marco Antonio Olguin-Sanchez, Marco Antonio Aceves-Fernández*,
Jesus Carlos Pedraza-Ortega and Juan Manuel Ramos-Arreguín
Autonomous University of Queretaro, Queretaro, Mexico

*Address all correspondence to: marco.aceves@uaq.mx

IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Sánchez CJ. Características fisicoquímicas de los gases y partículas contaminantes del aire. Su impacto en el asma. [Internet]. Available from: www.scielo.org.co/scielo.php?script=sci_abstract&pid=S0121-0793201200040007&lng=en&nrm=iso&tlng=es [Accessed: October 10, 2022]
- [2] Arbex MA, Saldiva PH, Pereira LA, Braga AL. Impact of outdoor biomass air pollution on hypertension hospital admissions. *Journal of Epidemiology and Community Health*. 2010;**64**(7):573-579. DOI: 10.1136/jech.2009.094342
- [3] Guo Y, Barnett AG, Zhang Y, Tong S, Weiwei Y, Pan X. The short-term effect of air pollution on cardiovascular mortality in Tianjin, China: Comparison of time series and case–crossover analyses. *Science of The Total Environment*. 2010;**409**(2):300-306. DOI: 10.1016/j.scitotenv.2010.10.013
- [4] Stephens SM, Sasha W, Olson F, Ramos J, Retama R, Armando, et al. Weekly patterns of México City's surface concentrations of CO, NO_x, PM₁₀, and O₃ during 1986-2007. *Atmospheric Chemistry and Physics*. 2008;**8**: 5313-5325. DOI: 10.5194/acpd-8-8357-2008
- [5] van Buuren S, Groothuis-Oudshoorn K. Mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*. 2011;**45**(3):1-67. DOI: 10.18637/jss.v045.i03
- [6] Siegelmann HT, Sontag ED. On the computational power of neural nets. *Association for Computing Machinery*. 1992;**1995**:440-449. DOI: 10.1145/130385.13043
- [7] Aggarwal CC. *Neural Networks and Deep Learning*. Midtown Manhattan, New York City: Springer Cham; 2018. pp. 293-294. DOI: 10.1007/978-3-319-94463-0
- [8] Kuri-Monge GJ, Aceves-Fernández MA, Ramírez-Montañez JA, Pedraza-Ortega JC. Capability of a recurrent deep neural network optimized by swarm intelligence techniques to predict exceedances of airborne pollution (PM_x) in largely populated areas. In: 2021 International Conference on Information Technology (ICIT), 2021. Amman, Jordan: IEEE; 2021. pp. 61-68
- [9] Ramírez Montañez JA, Aceves Fernandez MA, Arriaga ST, Ramos Arreguin JM, Salini Calderon GA. Evaluation of a recurrent neural network LSTM for the detection of exceedances of particles PM₁₀. In: 2019 16th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE). Mexico City, Mexico: IEEE; 2019. pp. 1-6
- [10] Wang J. An Intuitive Tutorial to Gaussian Processes Regression. [Internet]. 2020. Available from: https://www.researchgate.net/publication/344359964_An_Intuitive_Tutorial_to_Gaussian_Processes_Regression [Accessed: February 22, 2022]
- [11] SEDEMA. Red automática de monitoreo ambiental [Internet]. 2018. Available from: <http://www.aire.cdmx.gob.mx/> [Accessed: February 23, 2021]