DEPARTMENT OF ARTIFICIAL INTELLIGENCE

MASTER THESIS

Interpretable forecasts of NO_2 concentrations through deep SHAP

A thesis submitted in fulfilment of the requirements for the degree of Master in Advanced Artificial Intelligence: Fundamentals, Methods and Applications

Author:
María Vega García

Supervisor:
Jóse Luis Aznarte Mellado

Madrid, Spain.

2019

Master in Advanced Artificial Intelligence: Fundamentals, Methods and Applications

MASTER THESIS

Interpretable forecasts of NO_2 concentrations through deep SHAP

Author:

María Vega García

Supervisor:

Jóse Luis Aznarte Mellado

Fdo: María Vega García

Fdo: Jóse Luis Aznarte Mellado

Madrid, Spain. 2019

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

Abstract

Faculty of Computer Science Engineering
Dept. Artificial Intelligence

Master in Advanced Artificial Intelligence

Interpretable forecasts of NO_2 concentrations through deep SHAP
by María Vega García

Increasing in pollution levels in cities, especially in developed countries, and the consequences that it has on health and environmental, have prompted institutions to take preventive measures to reduce pollution levels. European Union has established thresholds for certain gases, such as nitrogen dioxide (NO_2). If these thresholds are exceeded, institutions of each city belonging to the EU, must activate measures previously defined, to go down NO_2 concentrations.

Meteorological parameters are related to episodes of high/low NO_2 concentrations. In this paper, pollution time series and meteorological features measured in the Madrid region were used to predict NO_2 concentrations. Detailed relationships between NO_2 concentrations and meteorological data were established through computational intelligence models. Therefore, these data were used to develop a predictive model. The proposal shows good precision in the prediction, proving computational intelligence has great potential in the pollution time series forecasting.

When constructing a prediction model, in addition obtaining good accuracy, it is important to know why that prediction is made. Interpreting the output of the computational intelligence model is difficult due to the architecture of the model itself. A SHAP (SHapley Additive Explanations) approach is applied to interpret the complex model outputs. This method assigns each feature a value of importance for a particular prediction. Three SHAP-based explanation methods are compared to determine which method is more suitable for the pollution time series data and for the computational intelligence model chosen.

In this research therefore, a model based on computational intelligence is developed to predict the levels of NO_2 concentrations. Through methods based on explanations we will obtain a deeper vision of how the computational intelligence model behaves on the pollution time series, allowing an increase in the confidence of the users on the results obtained from the prediction model. As a product of this study, a short version of this document has been sent for consideration in the Ecological Informatics[1] international journal on computational ecology and ecological data science.

[1] https://www.journals.elsevier.com/ecological-informatics

Contents

List of Figures

Chapter 1

Introduction

This introductory chapter gives a general description of the challenges generated by the interpretation of deep learning models. Specifically, the outputs generated by a deep learning model will be interpreted in the study of pollution time series. It offers a presentation of the air quality and meteorological factors involved, as well as the motivation and objectives of this research. Finally, there is a brief description of the structure of this document.

1.1 Presentation

Increasing in pollution levels in developed countries, specifically in cities where the largest population is concentrated, and due to environmental and health problems caused by high pollution levels, has driven research on this issue, especially in relation to predictive models, so that we can take preventive measures to reduce pollution levels. As discussed in Ballester Diez *et al.* (1999) and Cuchi *et al.* (2010), at the end of the 70s and during the following decade, most experts thought the levels that were registered in the most cities of the countries developed, air pollution did not represent a major health hazard. Today, 30 years later, the main agencies responsible for the protection of health and environment, such as OMS, European Environment Agency or EPA, recognize that the inhalation of pollutants represents an increased risk of death early.

This change is due to the advance in the knowledge and understanding of the effects of air pollution on health provided by a large number of scientific works worldwide. These studies have shown the importance of air quality in the health of the population and have identified the main mechanisms of action by which exposure to air pollution causes damage to health. The effects of air pollution on health range from alterations in lung function, leading to lung cancer, heart problems and other symptoms and discomforts that have increased mortality or hospital visits. Mortality in cities with high pollution levels exceeds between 15% and 20% that recorded in cleaner cities.

OMS for Europe established 40 $\mu g/m^3$ on average annually and 200 $\mu g/m^3$ on average in one hour for nitrogen dioxide to protect the population from the harmful health effects of gaseous NO_2. As an air pollutant, NO_2 is related to short-term concentrations above 200 $\mu g/m^3$, it is a toxic gas that causes significant inflammation of the airways and is the main source of nitrate aerosols and in the presence of ultraviolet light of ozone. The main sources of anthropogenic emissions of NO_2 are combustion processes: heating, electricity generation and motor vehicles and ships. Epidemiological studies have revealed that bronchitis symptoms in asthmatic children increase in relation to prolonged exposure to NO_2 and the decrease in lung function development is also associated with NO_2 concentrations currently in European and North American cities. To allow preventive measures and reduce NO_2 levels, field experts focus on predicting NO_2 levels that pose a high health risk.

These predictions could help research centers and institutions to advance the implications of high pollution concentrations and their duration. Periods of high pollution are defined as the period in which high concentrations of NO_2 are measured. This period has been defined when pollution levels exceed thresholds established. In addition, meteorology plays an important role in the severity and duration of pollution periods, as it is one of the causes of increases and decreases in the concentration levels of NO_2 Ballester Diez *et al.* (1999). For example, when temperatures are high it implies that pollution levels increase, in another case, when temperatures are low it influences the decrease of NO_2 concentrations. The difficulty of modelling time series lies in the nature of the series itself. Depending on the attributes of the time series, some techniques are more appropriate than others, which leads to a first decision on which model should be used. Deep learning models have shown good results for the time series forecast.

Machine or deep learning models have good precision in the time series forecast, however the interpretation of these models is complicated. These models for certain people are like a 'crystal ball' where entering the input data and magically returns some results. Understanding the outputs of complex models can help us to understand how the learning model, that produces those results, behaves. There are several interpretation methods that could be applied to deep learning models. It is here where we will have to decide which method is the most appropriate to explain the complex model outputs on the pollution time series.

1.2 Motivation

Historically, the problem of forecasting pollution time series has been addressed through time series analysis. These techniques depend largely on the selection and parametrization of the models. In recent years, several authors have proposed the use of computational intelligence techniques, mainly neural networks, although with the recent emergence of deep learning lately there are studies where these types of techniques are applied so that more accurate predictions are produced. These approaches depend on the selection of variables to input the model.

This selection is based on the relationships that exist between the meteorological variables and the concentrations of nitrogen dioxide. On the one hand, the proposals on time series analysis are sensitive to the collinearity of the variables, but are computationally smaller and their interpretation is simpler. On the other hand, deep learning models better capture the relationships between variables, however are computationally expensive and their results are difficult to explain.

Although deep learning models produce more accurate results, are still mostly black boxes. Humans have an important role that is sometimes overlooked. If users do not trust a model or a prediction, they will not use it. Users will rely on an individual prediction sufficiently to take any action based on it and will trust on a model if its implementation is done in a reasonable manner. Understanding the reasons behind the predictions are important enough to assess confidence. Also, this knowledge provides us with information about the model, which can be used to transform an untrustworthy model or prediction.

Due to the EU General Data Protection Regulation (GDPR) went into effect on May 25, 2018, decisions are generated only from automatic processing, usually using machine learning, must be explained to the interested party, what has led AI industry professionals to a debate on the right to explanation. These professionals do not agree on whether the explanation in the context of GDPR refers to how the technologies or the factors work that led to the automated decision. There are many different opinions in the AI community such that the GDPR will only give people information about the existence of an automated decision making or the functionality of the system, but will not explain the reason for that decision. GDPR in practice for the AI community suggests that an interested party has the right to know about the automated system so that can make an informed decision to opt out, or how the GDPR will make deep learning illegal. For all this, it is important to spend time studying the interpretation of prediction models.

Several investigations have been conducted to show methods for interpreting complex models. Some methods explain the predictions of any classifier or regressive faithfully and interpretatively by approximating them locally with an interpretable model or assigning each feature a value of importance for a particular prediction. Applying the methods based on explanations for a complex model to interpret the results of the prediction of pollution time series can be of great interest to understand how the model behaves and to be able to take preventive measures and reduce pollution levels. These are the central motivations for this research.

1.3 Objetives

The main objectives of this research are:

- Select a deep learning model which can cope with all the different requirements considered, taking into account accuracy, scalability and interpretation.

- Provide a framework that allows us to identify the features values that influence the result of the complex model forecasting, based on local explanation methods.

- Interpret the complex model outputs by analysing two different pollution scenarios, so that we can determine if there are significant differences in the features values that make the model prediction forecast one value or another.

1.4 Outline of the study

This section gives a more detailed description of the work presented in this document. The document is structured as follows: *Chapter 1* is an introductory chapter with preliminary information and where the motivation and objectives of this study are discussed. The main contributions of this work were detailed in comparison with the research related above, as well as the current detailed summary of the study.

Chapter 2 provides a detailed description of the materials used in the following chapters. It leads the reader through the problems caused by having high pollution concentrations, as well as temporary remedies that have been used to reduce pollution levels. By using meteorological phenomena in the same way as using features that capture the seasonality of the time series, it is intended to facilitate the prediction system by generating relevant features that may be influential in it.

Chapter 3 addresses the problem of finding a model to predict NO_2 concentrations in the city of Madrid. A regression based approach was adopted. Several prediction models adapted to the time series are commented, from traditional techniques to more current techniques. The proposal is to use a deep learning model that captures the dependencies of the time series in order to obtain a good accuracy.

Based on the outputs obtained from the model constructed in the previous chapter, *Chapter 4* consists of a study on the methods that help us to interpret the prediction generated by the complex model. First an introduction about the interpretation methods, interpretation scope and explanation properties are made. Second, additive feature attribution methods are presented and three methods belonging to this group are explained. Third, methods based on SHAP (SHapley Additive Explanations) are proposed and three methods that will be compared and analysed in depth. Finally, the most appropriate method to explain NO_2 predictions generated by the complex model is discussed.

Once we have decided the explanation method to interpret the prediction, in *Chapter 5* the SHAP-based explanation method is applied to understand the outputs of the deep learning model on two scenarios: considering all the NO_2 values within the data set and only studying the cases when the NO_2 is high. With the help of some plots, we can understand the features explanations so that we see locally and globally how the model behaves for both scenarios. Finally in the last part, *Chapter 6*, this work will be discussed in its entirety and will be finished with the conclusions obtained.

Chapter 2

Materials and methods

This chapter describes the materials and methods we have used to do the study. The city of Madrid has an anti-pollution protocol that is activated depending on the concentrations of nitrogen dioxide. It is important to determine the appropriate data to be able to predict the levels of nitrogen dioxide in a way that allows us to know in advance the activation of these protocols. To do this we should use, apart from pollution data, other data that are related to the levels of dioxide nitrogen, such as meteorological data. The methods used to construct the data set that will be used to finally build the pollution time series are described.

2.1 Data description

Pollution problems have increased especially in big cities, so governments have decided to take preventive measures so that pollution levels can be reduced. Complying with European regulations Commission (2008), the city of Madrid has a system for monitoring air pollution. The data gathered by this system are public and are available on an hourly basis de Madrid (2015). In 2016, local government imposed new anti-pollution measures in the protocol de Madrid (2018) that includes restrictions on traffic when the dioxide nitrogen concentrations reached certain thresholds established by the EU. As discussed in Aznarte (2017), anti-pollution protocol establishes three levels of action that rise according to the average dioxide nitrogen concentrations per hour: a previous warning for breach of a threshold of 180 $\mu g/m^3$, a second when the concentrations are above 200 $\mu g/m^3$ and an alert when values higher than 400 $\mu g/m^3$ are recorded. The city is divided into five zones, each with several stations. The protocols would be activated if these limits are violated in at least two stations of the same zone and if it occurs during two consecutive hours. We will focus only on the study of one station.

2.1.1 Pollution data

Study location and pollution stations correspond to hourly pollution concentrations registered at the surveillance systems. These data have been provided by the air quality network of the Ayuntamiento de Madrid. The observations consist of one location, specifically at Escuelas Aguirre station (located at 3º 40' 56,35" W, 40º 25' 17,63" N) from 2017 to 2018. As can be seen in the figure 2.1, pollution values show a clear intra-day pattern. The highest value are located in two peaks around the morning and the afternoon (with the highest average value at 21h), while the night hours (from 00h to 05h) have lower average concentrations. In the same figure we can see that the distribution of NO_2 presents a positive skew, with values over the thresholds being rare. In fact, the 99.9% of the data are equal or below the pre-warning threshold, while a total amount of 78 points exceed it.

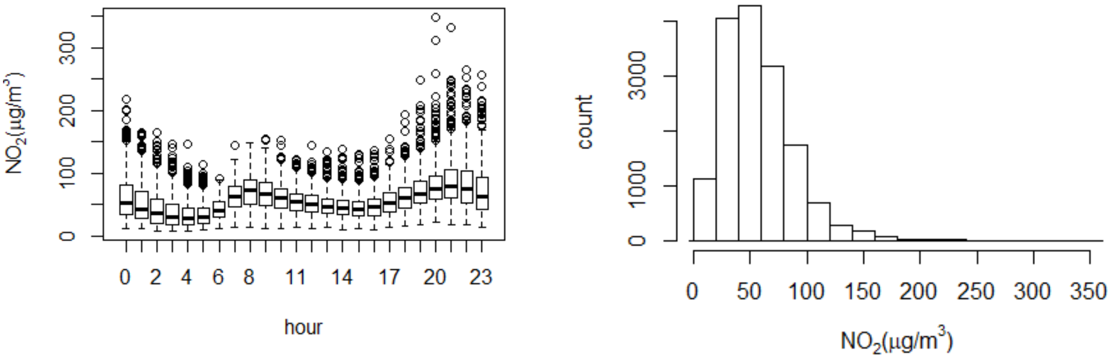

Figure 2.1: Intra-day distribution (left) and histogram (right) of nitrogen dioxide concentrations in the Escuelas Aguirre station from 2017 to 2018.

2.1.2 Meteorological data

Meteorological stations correspond to hourly meteorological data was provided by Ayuntamiento de Madrid. The observations consist of five locations from 2017 to 2018. Weather stations located in Junta Mpal. Moratalaz station (located at 3º 38' 43" W, 40º 24' 28" N), Junta Mpal. Villaverde station (located at 3º 42' 41" W, 40º 20' 54" N), E.D.A.R. La China station (located at 3º 40' 47" W, 40º 21' 57" N), Centro Mpal. De Acústica station (located at 3º 44' 25" W, 40º 26' 32" N) and Junta Mpal. Hortaleza station (located at 3º 39' 29" W, 40º 27' 46" N). Weather observations consist of average temperature in Celsius degrees, direction of the wind, precipitation in l/m^2, pressure in mb, solar radiation in kW/m^2, wind speed in m/s and degree of humidity in percentage.

Meteorological data does not belong to the same station as the pollution data. Therefore, to use this information, we have made an arithmetic average of the temperature, direction of the wind, precipitation, pressure, solar radiation, wind speed and percentage of relative humidity of the five stations. Two missing values were found in the meterological data that were estimated by the last observation carried forward method (LOCF). In order to take into account changes in the state of the atmosphere, an approximation to the mixture layer or atmospheric boundary layer Tutiempo (2019) is calculated by the difference of the pressure at time t less the pressure at time $t - 1$.

Figure 2.2 shows dispersion diagram between the meteorological and pollution data, so that we can have a preliminary view of how the data behaves towards each other. Figures 2.2 (A) and (F) show a relationship between the features where low values of wind speed and solar radiation produce high NO_2 values, the other graphs, at first glance, do not show much information and do not find significant evidence among the features. To conclude any relationship between these features with the pollution data, more analysis is needed to capture non-linearity.

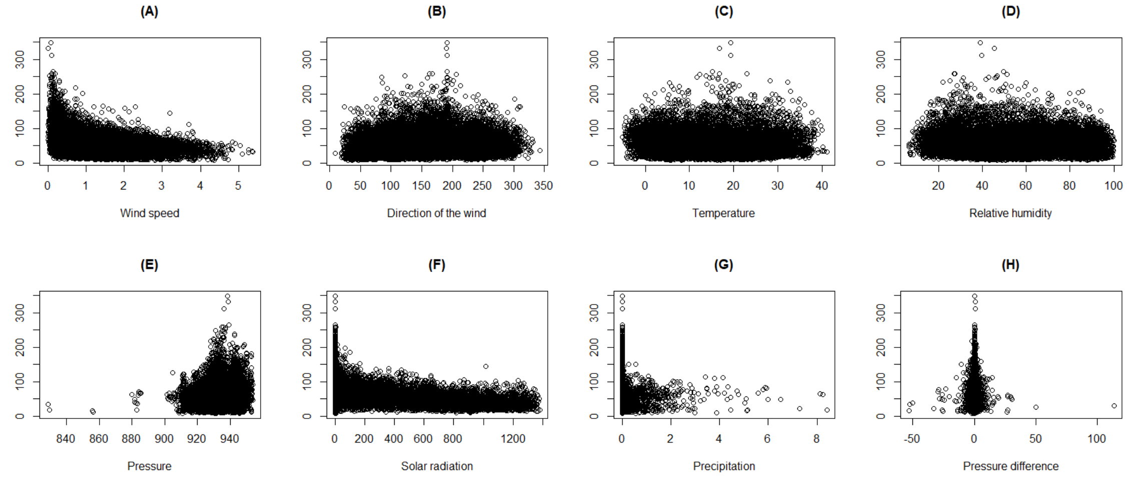

Figure 2.2: Scatter plots of meteorological features with pollution concentrations.

2.2 Features

In the problem of forecasting pollution concentrations, the independent features should contain relevant information. Meteorological data and dioxide nitrogen levels themselves play a crucial role in the pollution time series forecast. To avoid the problem of dimensionality, so that computing time is adequate, it is important not to include features that may not influence on the NO_2 concentrations. In our approach, meteorological data will be considered since encode an approximate picture of the local atmospheric situation, and are therefore related to the variability of NO_2 concentrations Arain *et al.* (2009).

Once the data was put together in the same data set, the series were aligned and merged using the time dimension and UTC times were used for all of them. Then, each of them was delayed to create new variables to take into account the possible autocorrelation features of the series in the following way: for NO_2 concentrations, a set of dimensions of the laps was selected in order to consider counts the recent past hours. That is:

$$d_{NO_2} = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,\\ 23, 24, 25, 26, 27, 28) \tag{2.1}$$

Different forecast horizons are not calculated for the meteorological data, only current time values are considered:

$$X_{met} = (TMP, VV, DV, HR, P, RS, LL, DP) \tag{2.2}$$

Besides, Fourier features that code the series' periodicities 2.3, a calendar feature and dioxide nitrogen predictions of a European model called MACC 2.4 are considered:

$$X_f = (S1.24, C1.24, S2.24, C2.24, S1.168, C1.168, S2.168, C2.168) \tag{2.3}$$

$$X_{em} = (MACC.1, MACC.2, MACC.3) \tag{2.4}$$

Then, for each hour t in the two years covered by the data, a vector was constructed as follow:

$$z_t = (y_t^{NO_2}, x_t^{TMP}, x_t^{VV}, x_t^{DV}, x_t^{HR}, x_t^{P}, x_t^{RS}, x_t^{LL}, x_t^{DP}, x_t^{cal}, x_t^{S1.24}, x_t^{C1.24}, x_t^{S2.24},\\ x_t^{C2.24}, x_t^{S1.168}, x_t^{C1.168}, x_t^{S2.168}, x_t^{C2.168}, x_t^{MACC.1}, x_t^{MACC.2}, x_t^{MACC.3}; y_{t+h}) \tag{2.5}$$

At the end, a matrix with 15.635 rows and 49 columns, 48 predictive variable and one response variable y_{t+h} are obtain 2.6. Then z_t corresponds to the data set on which a prediction model will be constructed.

$$\begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} = \left[\begin{array}{ccc|c} x_{1,1} & \cdots & x_{1,m} & y_{1+h} \\ \vdots & \ddots & \vdots & \vdots \\ x_{n,1} & \cdots & x_{n,m} & y_{n+h} \end{array}\right] \tag{2.6}$$

In order to define a regression problem, y_t is the hourly NO_2 observations at time t, n represents the instances the data set and m represents the predictive variables. h correspond with the forecast horizon.

Chapter 3

Chosing a model to forecast a pollution time series

This chapter addresses the problem of choosing the most appropriate model to predict pollution time series. A brief introduction is made about the models that are generally used to make the time series forecast. It starts from the most traditional models, based on statistical models, to the most current models, focused on machine and deep learning models. Long short term memory models have characteristics, in terms of time dependencies, which a priori seem to may be suitable for forecasting time series. The internal functioning of these models is discussed in depth so that we can understand how it behaves. Once the architecture of the LSTM models has been explained, the experimental design carried out to predict the NO_2 levels, through a pollution time series, for the city of Madrid is shown. Finally, the choice of the model evaluation metric is made.

3.1 Introduction

Time series data analysis has been an interest topic in fields such as Economics, Engineering and Medicine. Time series often contain temporal dependencies that cause two time points to belong to different classes or predict different behaviour. This characteristic generally increases the difficulty of analysing them. The methods used to carry out the study of time series are several. Traditional techniques generally depended on hand-made characteristics and required expert knowledge.

Traditional techniques are applied to time series can be seen in Hyndman & Athanasopoulos (2018). Time series data can exhibit a variety of patterns, and it is often useful to divide the time series into several components: trend, seasonality and cycles. There are several methods to extract these components, such as moving averages, classical, SEATS and STL decomposition. Often, this is done to improve the understanding of time series, but can also be used to improve forecast accuracy. Exponential smoothing has also motivated some of the most successful forecasting methods. These methods are based on a description of the trend and seasonality in the data. ARIMA models provide another approach to the time series prediction that are intended for describing the autocorrelations in the data. Exponential smoothing and ARIMA models are the two most commonly statistical approaches used to time series forecasting and give a complementary focuses to the problem. Other methods are the regression models that allow the inclusion of a large amount of relevant information for the predictor variables, however it does not permit the dynamics of subtle time series can be managed with the ARIMA models.

Subsequently, traditional artificial neural network (ANN) techniques have been applied to manipulate time series data focus on modeling and forecasting. The most recent researches include the use of recurent neural networks (RNN). Besides, hybrid approaches are frequent for the time series forecasting using ANN and ARIMA models as can see in Khashei & Bijari (2011) and Faruk (2010), as well as the use of Support vector machines models coupled with empirically decomposed set and partial autocorrelation function can see in Hu *et al.* (2013).

With the emergence of deep learning, new models have been used for time series analysis and forecasting. Several approaches to deep learning can be found in the literature to perform prognostic tasks, for example, Deep Belief Networks together with RBM Kuremoto *et al.* (2014), stacked Autoencoders Lv *et al.* (2014), Liu *et al.* (2015) and Liu *et al.* (2014) and instead of Autoencoders, Deep Belief Networks used to build a hybrid model in which the ANN models the joint distribution among the predictive time variables Grover *et al.* (2015). Within deep learning models, long short term memory (LSTM) models are capable of solving many time series tasks that can't be solved by forward networks that use time windows of fixed size.

LSTM is a recurrent neural network architecture that has been proven to outperform traditional RNN in numerous temporal processing tasks Gers *et al.* (2002). LSTM is able to learn about data with long and short range temporal dependencies, which makes it an interesting option, for example, in the data sequence problems Sutskever *et al.* (2014). LSTM is a problem that arises from the deployment of an RNN, since the gradient of some of the weights starts to be too small or too large if the network is deployed for too many time steps. This problem is solved with a LSTM network architecture Hochreiter & Schmidhuber (1997). A typical implementation of a LSTM network is the hidden layer is replaced by a complex block of computing units composed of gates that trap the error in the block. As follow the details of LSTM architecture.

3.2 LSTM model

Long Short Term Memory networks, usually called LSTM, are a special type of RNN, capable of remembering values over arbitrary intervals. LSTMs were introduced by Hochreiter and Schmidhuber (1997), and were refined and popularized by other people. These models work tremendously well on a large variety of problems, such as classification, processing and prediction time series given time lags of unknown duration. Relative insensitivity to gap length gives an advantage to LSTM over methods.

All recurrent neural networks have the form of a chain of repetitive modules of the neural network. In standard recurrent neural networks, this repetition module will have a very simple structure, with a single layer. LSTMs also have this chain structure, but the repetition module instead of having a single layer of neural network, has four layers that interact with each other as can see in the figure 3.1.

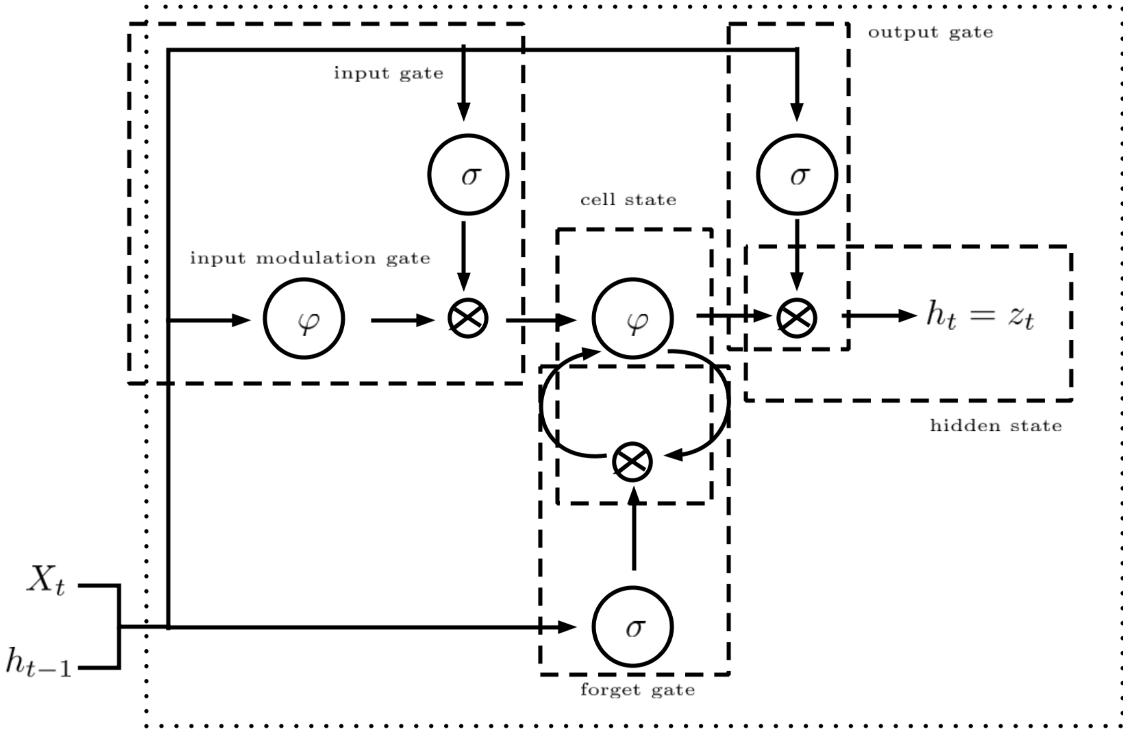

Figure 3.1: LSTM diagram

The key to LSTMs is the state of the cell, i.e, the memory part of the LSTM unit. The cell keeps a record of the dependencies between the input sequence elements. Loop arrows indicate the recursive nature of the cell that allow information from the previous intervals to be stored in the LSTM cell. LSTM has the ability to remove or add information to the state of the cell, always regulated by structures called gates. It consists of three gates: input gate, ouput gate and forget gate. Some variations of the LSTM unit do not have one or more of these gates or perhaps have other gates.

The first step in a LSTM is deciding what information will be removed from the state of the cell. Forget gate controls the extent to which a value remains in the cell. After getting the output of previous state h_{t-1}, forget gate helps taking decisions about what must be removed from h_{t-1} state and thus keeping only relevant stuff. The output of the forget gate tells the state of the cell what information should be forgotten or maintained. From equation 3.1, the activation function is applied to the weighted input and the previous hidden state.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3.1}$$

The next step determinate what new information is going to store in the state of the cell. Input gate controls the extent to which the new value flows into the cell. The important parts are the activation functions of each gate. Because the equation of the cell state 3.3 is a summation between the state of the previous cell, having only one activation function would add memory and could not delete and/or forget the memory. This is why the input modulation gate 3.2 has another activation function that allows the state of the cell forget the memory.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{3.2}$$

$$\bar{c_t} = \varphi(W_c[h_{t-1}, x_t] + b_c) \tag{3.3}$$

Now is the time to update the status of the previous cell c_{t-1}, in the new state of the cell c_t. Previous gates have already decided whether or not to allow information to flow through the cell. From equation 3.4, the state of the previous cell c_{t-1} is forgotten multiplying by the forget gate f_t and adding new information through the output of the input gate $\bar{c_t}$. These are the new candidate values, depending on how much it is decided to update each status value.

$$c_t = f_t \circ c_{t-1} + i_t \circ \bar{c_t} \tag{3.4}$$

Finally, it need to decide what we are going to output. Output gate controls the extent to which the value of the cell is used to calculate the activation output of the LSTM unit so that it advances to the next hidden state. From equation 3.5, the output of the state of the cell is multiplied with the output gate activation function so that generating on the hidden state only what want it.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3.5}$$

Working memory is usually called the hidden state 3.6, which is responsible for deciding what information should be passed to the following sequence. Also, hidden state is used for predictions.

$$h_t = o_t \circ \varphi(c_t) \tag{3.6}$$

In addition, there are connections inside and outside LSTM gates, a few of which are recurrent. The weights of these connections, which are learned during the training process, determine how the gates work.

3.3 Experimental design

Once the LSTM architecture was described, prediction model to forecast the dioxide nitrogen concentrations is shown. A priori a LSTM model is a good option to forecast pollution time series. Experimental design of the process is seen in the figure 3.2.

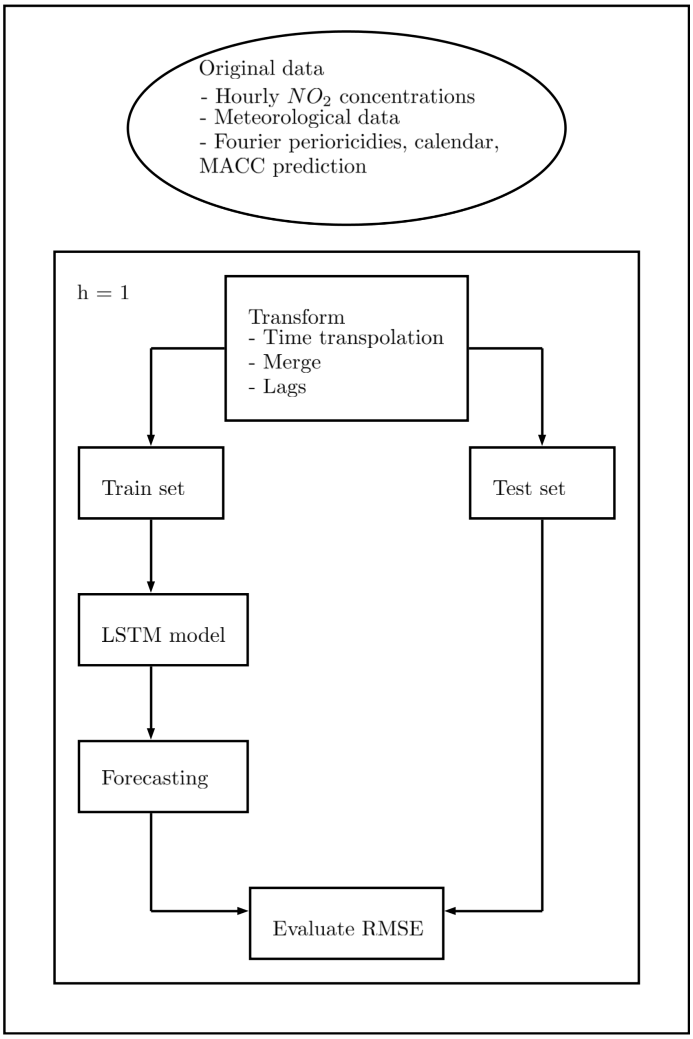

Figure 3.2: Diagram of experimental design

As commented in subsection 2.2, we obtained a data set with 15.635 rows and 49 columns. Throughout this document, we take h = 1, although h can take any positive integer value. In order to avoid overfitting, and assuming that NO_2 concentrations remained stable during the whole period, we divided our data set in two blocks: a block from 08/01/2017 02:00:00 to 03/05/2018 07:00:00 will be used to train the model, i.e, 10.476 instances and the rest of the data, 5.159 instances, from 03/05/2018 08:00:00 to 16/12/2018 16:00:00 will be used to test their properties.

LSTM model with 48 input nodes, a hidden layer of 20 LSTM blocks or neurons and 1 output node are build. The default Rectified Linear Unit (ReLU) activation function is used for the LSTM memory blocks. Once the architecture of the model is defined, we need to train the model. Training involves making a prediction based on the current state of the model, calculating how incorrect is the prediction and updating the weights or parameters of the network to minimize this error and do the model predict better. This process is repeated until our model converges and can no longer learn. Loss function is the mean square error (MSE), which calculates a loss value that the training process tries to minimize when fitting the weights of the network. The optimizer is Adam, which decides how the network weights will be updated based on the output of the loss function. LSTM model is trained for 200 times and a batch size of 32 is used.

3.4 Model evaluation

Once the model has been estimated, it is necessary to evaluate it to quantitatively estimate the generalization model capacity, i.e, the performance on the complete distribution of the possible data, and not only on the data set used to learn. To evaluate the performance of the LSTM model, we adopt two performance indexes: the mean square error (MSE) and the root mean square error (RMSE). These indexes are calculated as follows:

$$\text{MSE} = \frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y_j})^2 \qquad \text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y_j})^2} \tag{3.7}$$

where y_j denotes the observed concentration, $\hat{y_j}$ the corresponding predicted value and n the number of evaluation samples. MSE calculate the error between two data sets, i.e, calculate the difference between the observed values and the values predicted by the model. Finally, we use RMSE to determine the variability of the LSTM regression model.

Chapter 4

Comparing three explanation methods based on SHAP to interpret pollution prediction

This chapter discusses the explanation methods for interpreting the outputs of a prediction model. Once the prediction model is estimated, it is important to interpret its output. Knowing why a model makes that prediction provides confidence in the user, information about how a model can be improved and allows us to understand the process that the model performs. Many times simpler models are used since their interpretation is simpler, although this makes to obtain less accurate predictions. However, in many current applications, due to the increased amount of data, simple models are not the most appropriate. Interpreting complex models is complicated. From here, a variety of different methods have been proposed to address this problem. A brief introduction to interpretation scope, explanation properties and the different explanation methods will be made and then three explanation methods based on SHAP are compared to determine which one is the most suitable for the data and model used.

4.1 Introduction

When we estimate a prediction model and see that it works well, we wonder why we don't trust the model and ignore why a certain decision is made. The problem is that with a single evaluation metric we don't get a complete description about real-world tasks. When it comes to predictive modelling, in many cases it is sufficient to know that the predictive performance in a set of test data was good, because the problem may have little risk, that is, an error in the model will not cause serious consequences or method has already been studied and evaluated in depth. Some applications have been sufficiently studied so that there is sufficient practical experience with the model and the problems with the model have been resolved over time. But in other cases, knowing why that prediction was made can help us to learn more about the problem, the data, detect bias in the models and the reason why a model can fail, so that we can audit and debug it. In addition, for some problems we cannot settle for obtaining the prediction, since many times people have the curiosity to understand and learn why that prediction occurs, as well as to obtain the knowledge captured by the model. In these cases the interpretations and explanations of the model are crucial.

The concept of interpretability does not have a mathematical definition: Miller (2018) defines interpretability as the degree to which a human can understand the cause of a decision and Kim *et al.* (2016) as the degree to which a human can systematically predict the output of the model. Higher the interpretability of a model easier it is for someone to understand why certain decisions or predictions have been made. A model is more interpretable than another if its decisions are easier for one human to understand than the other's decisions. The best explanation for describing the model is generally the model itself, since a simple model is easily represented and understood. However, when we work with complex models, such as machine or deep learning models, the same model does not help since it is difficult to interpret due to its complexity. Therefore, in these cases we must look for a model that provides an explanation Lundberg & Lee (2017).

Models interpret complex models are the agnostic model methods that can be applied to any supervised learning model. Agnostic model methods work by changing the input of the machine or deep learning model and measuring changes in the prediction output. For example as commented on Molnar *et al.* (2018), some of these methods would be partial dependence plots and permutation feature importance. Other methods could be model independent methods that return data instances as explanations. These methods can be differentiated according to explain the general behaviour of the model, such as the Partial Dependence Plots, Accumulated Local Effects, Feature Interaction, Feature Importance, Global Surrogate Models and Prototypes and Criticisms, or explain the individual predictions, as Local Surrogate Models, Shapley Value Explanations and Counterfactual Explanations. Individual Conditional Expectation and Influential Instances explain the behaviour of the model both globally and individually.

4.2 Interpretation scope

Algorithm transparency is about how the algorithm learns a model from the data and what kind of relationships it can learn. Algorithm transparency only allows us to see how the algorithm works, but not for the specific model is learned at the end or for how the individualized predictions are made. Knowledge of the algorithm is required, but not of the data or model learned. In this study we focus on the interpretation of the model and not on the transparency of the algorithm. Simple models have generally been studied in depth and their interpretation is simple, so these models are characterized by high transparency. Deep learning models are less understood and internal functioning is still being studied, so it is considered less transparent.

A model is interpretable if you can understand the complete model at once Lipton (2016). To explain the result of the global model, you need the trained model, knowledge of the algorithm and the data. This interpretation level is about understanding how you make decisions of the model, based on its features and each of the components learned. Global interpretation of the model helps to understand what features are important and what interactions exist between them. This type of interpretation is difficult to achieve in practice, since when a model exceeds three dimensions it is hardly conceived for humans. In general, when trying to interpret a model, only model parts are considered. Global interpretation of the model is generally beyond our reach, but there is a good possibility of understanding at least some models at a modular level. Not all models are interpretable at the parameter level. Depending on the type of model, the interpretable parts will be one or another. In the case of model weights it will only make sense in the context of the other features of the model.

Examining the model and seeing what the model predicts for a particular input and explaining why. Local explanations may be more precise than global explanations, since the prediction locally can depend only linearly or monotonously on some features, rather than having a complex dependence on them. Methods which make individual predictions more interpretable, are agnostic model methods or methods based on explanations discussed below.

4.3 Explanation properties

Explaining model predictions through explanation method, an algorithm generates explanations. An explanation generally relates the feature values of an instance with the model prediction in a compressible way. Let's look first at the explanation methods properties and explanations commented on Robnik-Šikonja & Bohanec (2018). These properties determine how good an explanation method or explanation is. First explanation methods properties and then explanations properties are commented.

- Expressive power describes the language of the explanations that the method can generate.

- Translucency describes how much the explanation method is based on analysing the prediction model. It can be to decompose the internal representation of the model, treating the model as a black box or combining both cases. A model with high translucency can rely on more information to generate explanations, while a model with low translucency is more portable.

- Portability shows the set of prediction models that can be explained by the explanation method.

- Algorithmic complexity deals with the computational complexity of the method produces the explanations.

Quality of explanations is another very important aspect, which groups several properties of explanation methods:

- Accuracy is the ability that an explanation generalizes to other instances not yet known.

- Fidelity shows how well the explanation approximates the prediction of the prediction model. This property is very important since an explanation with low fidelity is useless to explain the prediction model. Local fidelity expresses how well the explanations reflect the model behaviour on a subset of data or for an individual data instance. Fidelity and precision are closely related, since if a model has high accuracy and an explanation has high fidelity, it will imply that the explanation also has high precision.

- Consistency is the degree to which similar explanations are generated from models have been trained in the same task and produce similar predictions. Similar models may produce similar predictions, but it does not influence explanations of similar instances to generate similar explanations due to the variation in the explanation method. However, if explanations are very similar then explanations are very consistent. High consistency would be desirable if the models are really based on similar relationships.

- Stability is the degree to which the explanations generated are similar for similar instances. While consistency compares explanations between models, stability compares explanations between similar instances for the same model. The predictions of similar instances may be the same, but it does not imply explanations are the same due to the variation of certain explanation methods. High stability means slight variations in the features of an instance do not substantially change the explanation. Stability lack may be the result of a high variation of the explanation method.

- Comprehensibility is the ability to understand explanations. This property is difficult to define and measure, but very important to succeed. Understanding explanations depends largely on people. Some ideas for measuring this property include measuring the explanation size or testing how well people can predict the behaviour of the prediction model from the explanations. In addition, it considered should be given to understand the features used in the explanation.

- Certainty corresponds with confidence. Explanations should show the certainty of the prediction model, since a model can be sure of its prediction but the explanation may or may not reflect it.

- Degree of importance shows the level at which the explanation reflects the feature importance.

- Novelty says if an explanation reflects the fact that the explained instance belongs to a new region far from the distribution of training data. This concept is related to certainty. The greater the novelty, surely the model has little certainty due to the lack of data.

- Representativeness indicates whether the explanations of the model can encompass the behaviour of the complete model or represent only an individual prediction.

In a typical environment of data science problems, users are concerned with both the prediction accuracy and the interpretation of the prediction model. Complex models have better accuracy, but are more difficult to interpret. It can be relieved by sacrificing some prediction accuracy for a more transparent model or by using an explanation method that improves the interpretation of the model. We will focus on the local explanation methods and comment in depth on additive feature attribution method and explanation methods based on SHAP as we will see below.

4.4 Additive Feature Attribution Method

As commented previously, the best explanation to interpret it is to use the same model. A simple model is easily interpretable, but when we work with complex models, such as deep learning models, the same model does not help us, since its interpretability is complicated due to its complexity. We must look for a model that provides an explanation for this type of model. The current explanation methods, such as LIME Ribeiro *et al.* (2016), DeepLIFT Shrikumar *et al.* (2017), Layer-Wise Relevance Propagation Bach *et al.* (2015), Shapley regression values Lipovetsky & Conklin (2001), Shapley sampling values Strumbelj & Kononenko (2014) and Quantitative input influence Datta *et al.* (2016) use the same explanation model.

Let f be the original prediction model we want to explain and g the explanation model. We focus on local methods designed to explain a prediction $f(x)$ based on a single x input. Explanation models often use simplified entries $x^{'}$ that are assigned to the original inputs through a mapping function $x = h_x(x^{'})$. Local methods always try to ensure $g(z^{'}) \approx f(h_x(z^{'}))$ whenever $z^{'} \approx x^{'}$, as discussed in Lundberg & Lee (2017). Explanation method used by the previous methods is the additive feature attribution method which is based on a linear function of binary variables:

$$g(z^{'}) = \phi_0 + \sum_{i=1}^{M} \phi_i z_i^{'} \tag{4.1}$$

where $z^{'} \in \{0,1\}^M$, M is the number of simplified input features, and $\phi_i \in R$. An effect ϕ_i to each feature is attributed, and summing the effects of all feature attributions approximate the output $f(x)$ of the original model. Many current methods match with additive feature attribution method, three of which are discussed below.

4.4.1 LIME

Local Interpretable Model-agnostic Explanations (LIME) presented by Marco Tulio Ribeiro, Sameer Singh and Carlos Guestrin in 2016 Ribeiro *et al.* (2016) is an algorithm that interprets the predictions of any individual model faithfully, approximating locally an interpretable model around a given prediction. Local linear explanation model, that LIME uses, fits to equation 4.1 exactly and, therefore, is an additive feature attribution method.

An explanation is defined as a $g \in G$ model, where G is a class of potentially interpretable models. The domain of g is $\{0,1\}^{d'}$, i.e, g acts on the presence/absence of the interpretable components. Since not all $g \in G$ models can be simple enough to be interpretable, we take $\Omega(g)$ as a measure of the complexity of the explanation $g \in G$. In addition, we use $\pi_x(z)$ as a measure of proximity between an instance z to x, in order to define it locally around x. Let $L(f, g, \pi_x)$ be a measure that indicates how much unfaithful is g to approximate f in the locality defined by π_x. To find the effect ϕ, LIME minimizes the following objective function:

$$\xi = \underset{g \in G}{\arg\min}\, L(f, g, \pi_x) + \Omega(g) \tag{4.2}$$

L is minimized to ensure the interpretation and local fidelity of the explanation model g to the original model f using $\Omega(g)$ low enough so that it can be interpreted.

4.4.2 DeepLIFT

Deep Learning Important FeaTures (DeepLIFT) is a method proposed by Avanti Shrikumar Shrikumar *et al.* (2017). DeepLIFT is a novel algorithm for assigning an importance score to the inputs for an output given and was proposed as a method of explaining recursive prediction for deep learning models. Importance is calculated using the differences of a reference state. This difference allows the method to propagate an important signal, even in situations where the gradient is zero and as the reference difference is continuous, it avoids discontinuities in the gradient caused by bias terms. DeepLIFT decomposes the output prediction of a deep learning model into a specific input by propagating backwards the contributions of all the neurons in the network to each input features. Method compares the activation of each neuron with its reference activation and assigns contribution scores according to the reference difference.

Let t be the activation of an interest neuron and let $x_1, \ldots, x_n$ be neurons in some intermediate layer necessary and sufficient to calculate t. Suppose that t^0 represents the reference activation of t. We define the amount Δt as the reference difference, that is $\Delta t = t - t^0$. We denote the contribution scores as $C_{\Delta x_i \Delta t}$ for Δx_i, that is, it is considered as the amount of reference difference of t attributed to the reference difference of x_i. DeepLIFT uses a 'summation-to-delta' property that states:

$$\sum_{i=1}^{n} C_{\Delta x_i \Delta t} = \Delta t \tag{4.3}$$

That is, the sum over all the contributions of neurons in $C_{\Delta x_i \Delta t}$ to Δx_i equals the difference-from-reference of t. If we take $\phi_i = C_{\Delta x_i \Delta t}$ y $\phi_0 = t^0$, then DeepLIFT's explanation model matches equation 4.1 and is thus additive feature attribution method.

4.4.3 Layer-Wise Relevance Propagation

Layer-Wise Relevance Propagation (LRP) method was proposed by Bach Bach *et al.* (2015), which defines relevance as the contribution of each input variable to the prediction. Shrikumar *et al.* (2017) and Kindermans *et al.* (2016) demonstrated that, in the absence of modifications to deal numerical stability, the original LRP rules were equivalent within a scale factor to an elementary product between the gradient and the input. Shrikumar *et al.* (2017) comments that this method is equivalent to the DeepLIFT method with the reference activations of all neurons set to zero. Then $g \in G$ is an explanation model for the original model f, so it coincides with equation 4.1 and is therefore an additive feature attribution method.

4.5 SHAP (SHapley Additive exPlanation) Values

Many current methods, such as the three methods discussed above, to interpret the individual predictions of the machine or deep learning model are part of the additive feature attribution method Lundberg & Lee (2017). This class of methods explains the model output as a sum of real values attributed to each input feature. A surprising attribute of the class of additive feature attribution method is the presence of a single unique solution in this class with three desirable properties: local precision, missingness and consistency. Local precision indicates that the sum of the feature attributions is equal to the output of the function which are trying to explain. Missingness states that features that are already missing attributed no importance. Consistency means that, even if we change a model so that a feature has a greater impact on the model, the attribution assigned to that feature will never decrease. These properties are familiar to the classical methods of estimating Shapley values Lipovetsky & Conklin (2001), Strumbelj & Kononenko (2014) and Datta *et al.* (2016), but are unknown to the other additive feature attribution methods Ribeiro *et al.* (2016), Shrikumar *et al.* (2017) and Bach *et al.* (2015).

We calculate SHAP values, as a unified measure of feature importance, by defining $f_x(S) = f(h_x(z^{'})) = E\left[f(x) \mid x_S\right]$ where S is the set of non-zero indexes in $z^{'}$ and $E\left[f(x) \mid x_S\right]$ is the expected value of the function conditioned on a subset S of the input features. SHAP values combine these conditional expectations with game theory and with classic Shapley values to attribute ϕ_i values to each feature. Only one possible explanation model g that follows equation 4.1 and according Lundberg & Lee (2017) satisfies the three properties is as follows:

$$\phi_i = \sum_{S \subseteq N \backslash \{i\}} \frac{\mid S \mid ! (M - \mid S \mid -1)!}{M!} \left[f_x(S \cup \{i\}) - f_x(S))\right] \tag{4.4}$$

where N is the set of all input features. This result implies that methods not based on Shapley values violate local precision and/or consistency. However, this SHAP value definition is designed to align with Shapley regression values Lipovetsky & Conklin (2001), Shapley sampling values Strumbelj & Kononenko (2014) and Quantitative input influence Datta *et al.* (2016), while allowing connections with LIME Ribeiro *et al.* (2016), DeepLIFT Shrikumar *et al.* (2017) and Layer-Wise Relevance Propagation Bach *et al.* (2015).

As discussed in Lundberg & Lee (2017), calculating the exact value of SHAP values is a challenge. However, by combining the ideas of the additive feature attribution method, we can approximate these values. SHAP is a unified approach to explain the result of any machine or deep learning model which connects the theory of games with local explanations, joining several methods and representing the additive feature attribution method. A model-agnostic approximation method (Kernel SHAP) and two specific approximation methods of the model type (Gradient SHAP, Deep SHAP) are described. Kernel SHAP improve the sample efficiency of estimates of SHAP values without taking into account the model type. By restricting ourselves to the specific model type, such as Deep and Gradient SHAP, faster approximation methods obtain.

4.5.1 Kernel SHAP

Linear LIME uses a linear explanation model to locally approximate f. Linear LIME is an additive feature attribution method and we know Shapley values are the only possible solution to equation 4.2 and satisfy the three properties discussed above. Since LIME selects the L function, the kernel weighting $\pi_{x'}$ and the regularization term Ω in a heuristic form, equation 4.2 doesn't recover the Shapley values. One consequence is that local precision and/or consistency properties are violated.

Let's see how to calculate the parameters of equation 4.2 and how to find $\pi_{x'}$, L, and Ω that retrieves the Shapley values. According to Shapley kernel theorem commented on Lundberg & Lee (2017), under additive feature attribution method definition, the specific forms of $\pi_{x'}$, L, and Ω that make solutions of equation 4.2 consistent with the three properties are the following:

$$\Omega(g) = 0 \tag{4.5}$$

$$\pi_{x'}(z^{'}) = \frac{(M-1)}{(M \text{ choose } \mid z' \mid) \mid z' \mid (M- \mid z' \mid)} \tag{4.6}$$

$$L(f, g, \pi_{x'}) = \sum_{z' \in Z} \left[f(h_x(z^{'})) - g(z^{'}) \right]^2 \pi_{x'}(z^{'}) \tag{4.7}$$

where $\mid z^{'} \mid$ is the number non zero elements in $z^{'}$. We must keep in mind that $\pi_{x'}(z^{'}) = \infty$ when $\mid z^{'} \mid \in \{0, M\}$, which enforces $\phi_0 = f_x(\emptyset)$ and $f(x) = \sum_{i \in \{1,\ldots,M\}} \phi_i$. However, in practice these infinite weights can be avoided during optimization using the restrictions mentioned.

In the Shapley kernel theorem we can see that $g(z^{'})$ follows a linear form and that L is a square loss function. Therefore, equation 4.2 can still be solved using linear regression. From here, we deduce that the input mapping that LIME uses is equivalent to the approximation of the SHAP mapping $f(h_x(z^{'})) \approx f([Z_S, E[Z_S]])$, so we obtain a regression model based on an agnostic model for estimating SHAP values.

Because of these statements between linear regression and SHAP values, we see that equation 4.4 corresponds to a difference in means. From here, it is normal to use a kernel so that through a linear least squares regression to recapitulate Shapley values. As we have just seen, the parameters of equation 4.2 are not chosen in a heuristic way, so recover the Shapley values and Kernel SHAP is an additive feature attribution method that satisfy local precision, missingness and consistency.

4.5.2 Gradient SHAP

According to Sundararajan *et al.* (2017), for linear models, the products of the model coefficients and feature values are regularly inspected to debug the predictions. The gradients correspond to the coefficients of the deep network model and, therefore, the product of the gradient with the feature values may approach an attribution feature method. However, gradients break with sensitivity, a property that all attribution feature methods must satisfy. As we have commented in the section 4.5, attribution methods must verify three properties: local precision, missingness and consistency. Gradients violate the missingness, but satisfy the consistency property. In addition, we find that other attribution feature methods in the literature break at least one of these axioms. These methods include DeepLift Shrikumar *et al.* (2017) or LRP Binder *et al.* (2016) which do not fulfil local precision and/or consistency.

Integrated gradients method that combines the consistency of the gradients with the missingness of methods, such as DeepLift Shrikumar *et al.* (2017) or LRP Binder *et al.* (2016) is defined. Suppose a function $F : \mathbb{R}^n \rightarrow [0,1]$ that represents a deep network. Let $x \in \mathbb{R}^n$ be the input at hand and $x^{'} \in \mathbb{R}^n$ the baseline input. A straight line from the baseline to the input is considered and gradients are calculated at all points. Integrated gradients are obtained by accumulating these gradients. Integrated gradients along the i^{th} dimension for an input x and the baseline input $x^{'}$ are defined as follows:

$$\text{IntegratedGrads}_i(x) := (x_i - x_i^{'}) \times \int_{\alpha=0}^{\alpha=1} \frac{\partial F\left(x^{'} + \alpha \times (x - x^{'})\right)}{\partial x_i} \cdot d\alpha \tag{4.8}$$

where $\frac{\partial F(x)}{\partial x_i}$ is the gradient of $F(x)$ along the i^{th} dimension.

Integrated gradients satisfy an axiom called completeness, so it implies the missingness as discussed in Sundararajan *et al.* (2017). Missingness refers to a case in which the baseline and the input differ by one variable, for which the completeness states that the difference between the two output values is equal to the attribution to this variable. In addition, integrated gradients satisfy the consistency because are only based on the gradients of the function represented by the network. However, integrated gradients is not the only unique method to do so. We identify a class of methods called path methods or attribution methods based on integrated gradients that generalize the integrated gradients, so that path methods are the only methods that satisfy the properties discussed above.

As integrated gradients add the gradients along the inputs that fall in the straight line between the baseline input and the input, there are many other paths that monotonously interpolate between the two points and therefore each of those paths produces a different attribution method. Let $\gamma : (\gamma_1, \ldots, \gamma_n) : [0, 1] \rightarrow \mathbb{R}^n$ be a smooth function that specifies a path in $\mathbb{R}^n$ from baseline input $x^{'}$ to input x. Given this path function γ, path integrated gradients are obtained by integrating the gradients through the path $\gamma(\alpha)$ for $\alpha \in [0, 1]$. Formally, path integrated gradients along the i^{th} dimension for an input x are defined as follows:

$$\text{PathIntegratedGrads}_i^{\gamma}(x) := \int_{\alpha=0}^{\alpha=1} \frac{\partial F(\gamma(\alpha))}{\partial \gamma_i(\alpha)} \cdot \frac{\partial \gamma_i(\alpha)}{\partial \alpha} \cdot d\alpha \tag{4.9}$$

where $\frac{\partial F(x)}{\partial x_i}$ is the gradient of $F(x)$ along the i^{th} dimension. Path methods are the only attribution methods that always satisfy the three properties.

Once these methods have been defined, we can explain a model based on the path integrated gradients. Expected gradients are an implementation based on path integrated gradients and Shapley values. Expected gradients are an attribution feature method designed for differentiable models based on an extension of Shapley values. Values of the path integrated gradients are a bit different from the Shapley values, it require a unique reference value to integrate them. Expected gradients, also known as Gradient SHAP, arise as an adaptation to make these methods approximate Shapley values. These methods reformulate the integral as an expectation and combine that expectation with reference values for the entire data set. Gradient SHAP is an additive feature attribution method that satisfy local precision, missingness and consistency.

4.5.3 Deep SHAP

The connections between the Shapley values and DeepLIFT Shrikumar *et al.* (2017) allow us to take advantage of the knowledge about deep networks to improve the computational performance of the methods. DeepLIFT approximates Shapley values assuming that the input features are independent of each other and the deep model is linear Lundberg & Lee (2017). DeepLIFT is an additive feature attribution method that satisfies two properties discussed above, local precision and missingness, and we know Shapley values represent the only attribution values that satisfy the three properties including consistency property. That is why DeepLIFT can become a compositional approximation of Shapley values which results in what we will call Deep SHAP.

Deep SHAP combines Shapley values calculated for smaller components of the network into Shapley values for the whole network, recursively passing the DeepLIFT multipliers, calculated in terms of Shapley values, backward through the network. A linear approximation would be the following:

$$\phi_i(f_3, y) \approx \sum_{j\in\{1,2\}} \frac{\phi_i(f_j, y)}{y_i - E[y_i]} \cdot \frac{\phi_i(f_3, x)}{x_j - E[x_j]}(y_i - E[y_i]) \tag{4.10}$$

Deep SHAP derives an effective linearization of the Shapley values, calculated for each component. Deep SHAP does not perform this process in a heuristic way as DeepLIFT does, so Deep SHAP is an additive feature attribution method that satisfy local precision, missingness and consistency.

4.6 Results and discussion

Three methods based on SHAP have been described, which are local explanation methods and it is time to decide which of them is the best suitable to describe the data and the model. First of all, data set and model used to calculate the explanations are defined. Data set and features were discussed in chapter 2. Model necessary to calculate SHAP values was a LSTM model discussed in chapter 3. Data set, as discussed above, was separated into two blocks: train and test set.

Similar results were obtained with the Gradient SHAP and Deep SHAP explanators. However, better performance is obtained with Deep SHAP than with Gradient SHAP for deep learning models, as commented in SHAP (2019). Kernel SHAP explainer could not be computationally configured for a LSTM model, because Kernel SHAP needs one or two dimensions passed through the prediction model. However, a LSTM network expects the input data X to be provided with a specific matrix structure in the form of [samples, time steps, features]. If data should be of the form [samples, features], so that Kernel SHAP could be used, it would imply that we would be framing our problem as a time step for each sample, scenario in which we are not, because in model use displaced features of the response variable.

It has been decided to use Deep SHAP as an explanation method. Process to calculate Deep SHAP explainer consists in calculating the explainer through LSTM model and train set. Taking this explainer and test set, SHAP values are obtained. Diagram is shown in the figure 4.1.

Once SHAP values were calculated to explain LSTM model outputs, we decided to keep only the most promising features to explain the predictions. Thus, instead of explaining the model output with 49 features, we will explain it with 14. Vector, which was build in the subsection 2.2, will be reduced to the following:

$$z_t = (y_{t-1}, y_{t-2}, y_{t-23}, y_{t-24}, y_{t-25}, x_t^{TMP}, x_t^{VV}, x_t^{DV}, x_t^{HR}, x_t^{P}, x_t^{RS}, x_t^{LL}, x_t^{DP}, x_t^{cal}; y_{t+h}) \tag{4.11}$$

As a result, we obtain a matrix of 15.635 rows, divided into 10.476 for training and 5.159 for test, and 15 columns 4.12 (14 predictive variables and one response variable).

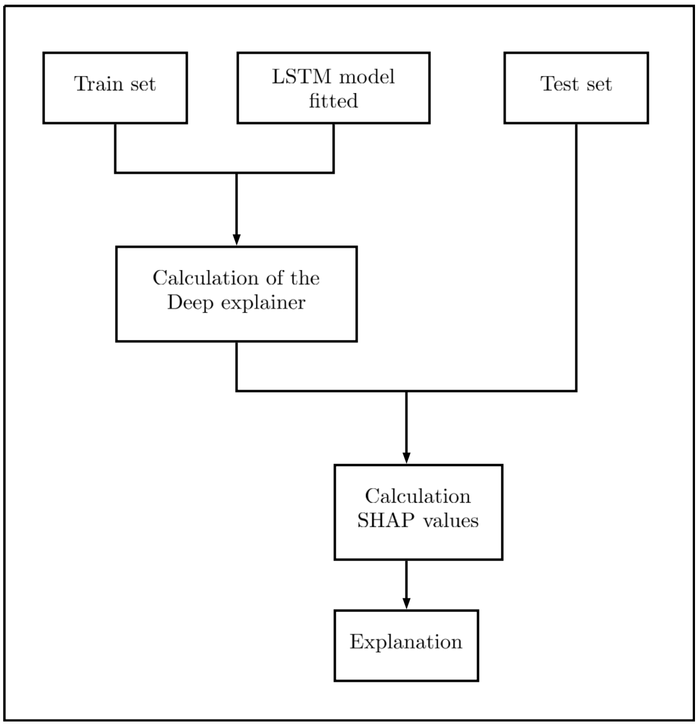

Figure 4.1: Diagram of SHAP experimental design

$$\begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} = \left[\begin{array}{ccc|c} x_{1,1} & \cdots & x_{1,m} & y_{1+h} \\ \vdots & \ddots & \vdots & \vdots \\ x_{n,1} & \cdots & x_{n,m} & y_{n+h} \end{array}\right] \tag{4.12}$$

where y_t is the hourly NO_2 observations at time t, n represents the instances the data set and m represents the predictive variables. h correspond with the forecast horizon.

As we have just commented, this is the process we have followed to calculate the explanations that help us to understand the LSTM model outputs. We take all the instances of the data set, that is, we want to explain what happens in any circumstance, when pollution levels are high, low or moderate in the city of Madrid. Apart from this scenario, we will calculate SHAP values when high NO_2 concentrations are present. Defining the threshold at 180. Difference with the previous case is test set with 5.159 instances is filtered and we only keep the instances whose value is greater than 180. A subset with 15 rows and 14 columns is obtained. So SHAP values are calculated with this test subset data and LSTM model.

SHAP values are calculated with the general data set of the test set and with test subset. Training set is not filtered because we want to maintain the same expected value for both scenarios. This decision is taken so that we can compare both cases to be able to analyse whether the outputs of the LSTM model show differences, or per contra, behave in a similar way for a pollution scenario or another.

Chapter 5

Deep SHAP to explain LSTM forecast

In this chapter SHAP values are applied to explain the predictions generated by the LSTM model. Specifically, one of the explanation methods discussed in the previous chapter 4 applies: Deep SHAP. To understand the explanations obtained by the Deep SHAP method, we rely on some graphs so that we can understand the predictions both locally and globally. In this way we get a complete view of why the LSTM model makes that prediction. Model predictions are explained taking into account all the values available in the data set and subsequently, the values generated when the pollution levels in the city of Madrid are high, are explained. By comparing these two situations, we can see if the model outputs are influenced by different features or not, depending on the data set used.

5.1 Introduction

SHAP does a great job of decoding the influence of the input variables in the predictions. SHAP values calculate the importance of a feature by comparing what a model predicts with and without the feature. However, since the order in which a model sees features can affect its predictions, this is done in every possible order, so that the features are fairly compared. Let see how SHAP can help us to obtain a local knowledge, i.e, how the probability of obtaining a higher/lower NO_2 level of each observation is formulated.

In this experiment two scenarios are studied. The first one takes the entire available data set and the second, a subset of values where the concentrations of NO_2 takes high values. Data and features used are described in chapter 2. LSTM model applied to forecast pollution time series is discussed in the chapter 3. Once the model is estimated, a train error equal 17.35 RMSE and a test error equal 18.90 RMSE are obtained. From here, we can apply the SHAP-based indicators to explain the predictions generated by the LSTM model. As discussed in the chapter 4, we will focus on the SHAP indicators for the Deep SHAP explainer.

Explanations obtained by the Deep SHAP method are represented graphically.

Features impact of a model is usually represented by a bar plot to show the global importance of the features or a partial dependence plot to represent the effect of changing a single feature. However, since SHAP values are attributions of individualized features, unique to each prediction, it allows other types of representations. Force plots show explanations locally, SHAP summary plots replace typical bar plots to show global importance and SHAP dependence plots give an alternative to partial dependence plots, since it better capture the effects of feature interactions.

5.2 Complete data set of the pollution values

On the one hand, the results are displayed when take all pollution values. Explanation of the first prediction using a LSTM model can see in the figure 5.1. Explanation shows features each contributing to push the model output from the base value to the model output for the first prediction. Features cause a higher value in the prediction are in red and those causes it to take lower value are in blue. Explanations for the complete test data set can see in the figure 5.2.

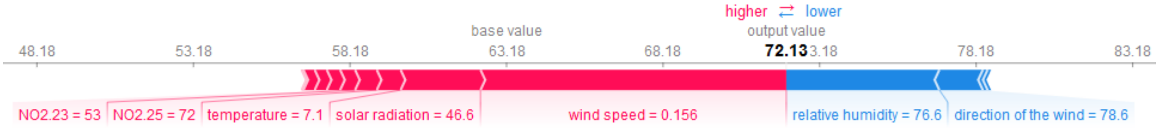

Figure 5.1: First prediction generated by the LSTM model was explained using Deep SHAP. Red feature attributions push the score higher, while blue feature attributions push the score lower.

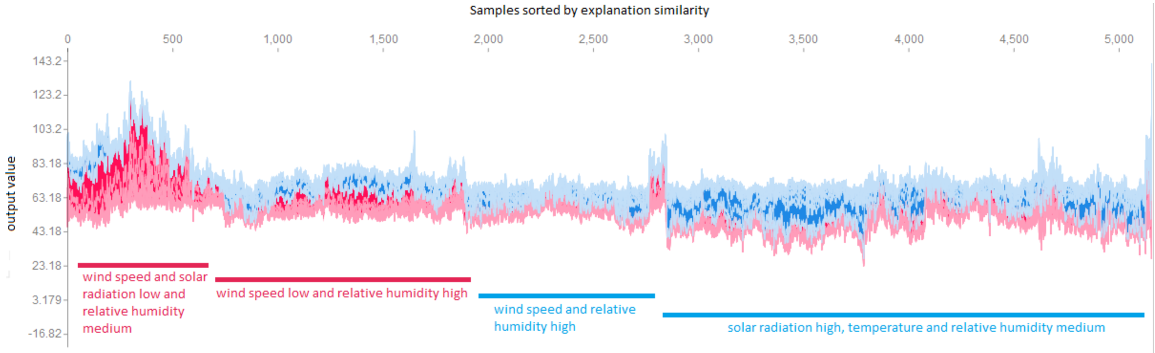

Figure 5.2: LSTM model with the attributions of SHAP features in the pollution time series identifies among 5000 instances that share similar reasons for the concentration of NO_2. The prediction generated by the LSTM model was explained using Deep SHAP. Red feature attributions push the score higher, while blue feature attributions push the score lower (as in the figure 5.1 but turned 90°). A few of the noticeable subgroups are annotated with the features that define them.

Deepen the explanations so that we have an overview of what features are most important to the model. Importance standard bar plots of the feature show the relative importance in the training data set, however it does not represent the range and distribution of the impacts that the feature has on the model output, and how the feature value is related to its impact. We can plot the SHAP values of each feature for each sample to visualize the effect of the features among the population through the SHAP summary plots.

In Figure 5.3 a) Features are classified by the sum of the magnitudes of the SHAP values in all the samples, i.e, by their global impact $\sum_{j=1}^{N} \mid \phi_i^{(j)} \mid$. SHAP values are used to show the distribution of the impacts that each feature has on the model output. SHAP values $\phi_i^{(j)}$ are drawn horizontally, stacked vertically when it runs out of space. Each point represents a row of the data set. The gradient color indicates the original value of that feature (high red, low blue). If the impact of the function on the model output varies smoothly as its value changes, then this color will also have a smooth gradation. Also, we can just take the average absolute value of the SHAP values for each feature to get a standard bar graph, figure 5.3 b). The vertical axis indicates the variable name, in order of importance from top to bottom. On the horizontal axis is the SHAP values that indicates how much is the change in log-odds.

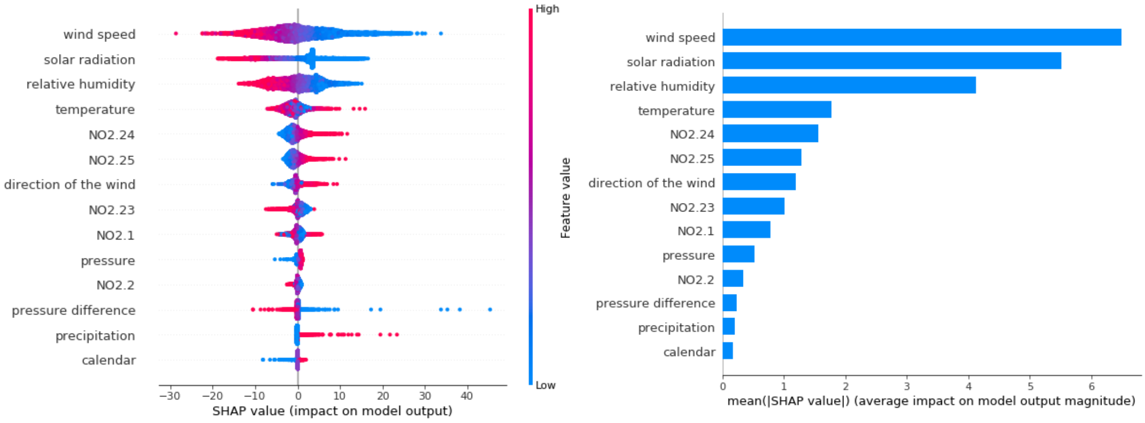

Figure 5.3: a) SHAP summary plot of a LSTM model with 14 features of the pollution time series. The higher SHAP value of a feature, the higher NO_2 levels. Each individual in the data set is executed through the model and a point is created for each feature attribution value, so that each instance is displayed as a point on the line of each entity. Points are coloured by the feature value for each instance and are accumulated vertically to show the density. b) Corresponds to the same graph as a), but taking the average absolute value of the SHAP values for each feature, obtaining a bar graph.

The partial dependence plots represent the expected output of a model when the value of the features are fixed. Feature values vary and the result of the model is represented. Showing how the model outputs change as the features change helps us to explain how the model depends on that feature. An alternative to these plots using the SHAP values are the SHAP dependence plots.

The impact of wind speed, solar radiation and relative humidity extends over a relatively wide range. To understand how these features effect the model output, i.e, how the importance attributed to the features change as its values changes. We can plot the SHAP value of these features vs. the value of these feature for all the examples in a dataset, figure 5.4. While standard partial dependence plot only produce lines, SHAP values are represented as a function where each point represents an instance of the data set. In this way, SHAP dependence plot capture vertical dispersion due to the interaction effects in the model. The horizontal axis shows the real value of the features, the vertical axis shows the effect of each feature on the prediction and interaction effects can be visualized by colouring each point with the value of another feature. Besides, this information is shown in three axes so that we better capture the relationships between variables, as can be seen in the figure 5.5.

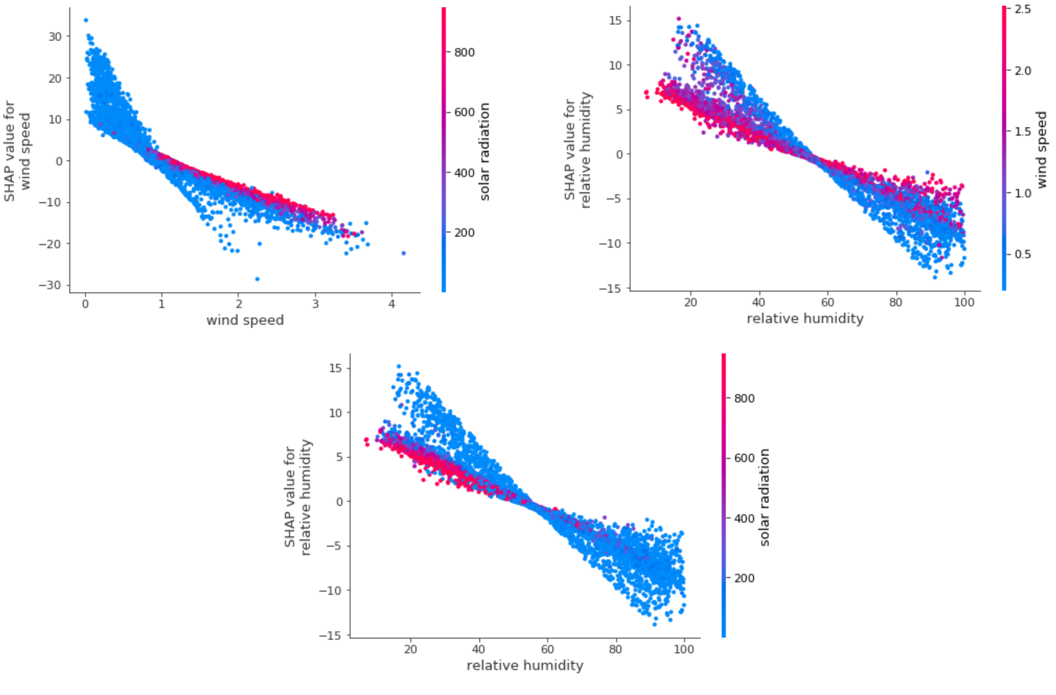

Figure 5.4: SHAP dependence plots of a LSTM model of the pollution time series. Each point is an instance. The x-axis represents one feature and the y-axis represents the SHAP value attributed to that feature. Each point is coloured by the another feature.

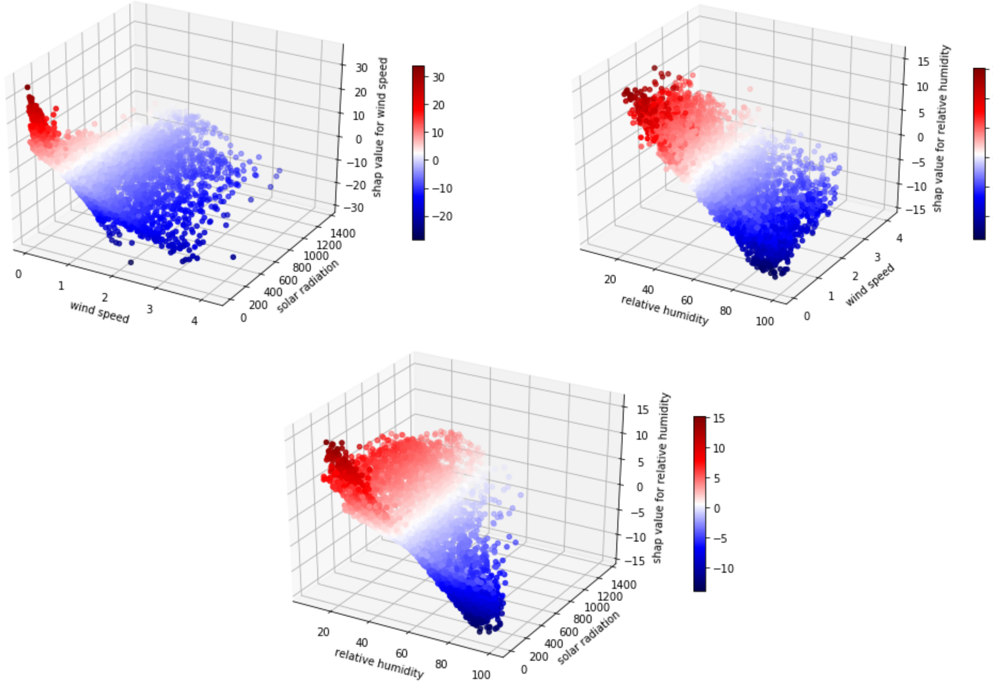

Figure 5.5: SHAP dependence plots of a LSTM model of the pollution time series. Each point is an instance. The x-axis and the y-axis represent the two features and the z-axis represents the SHAP value attributed to one of the two features. Each point is coloured based on the SHAP value attributed to a feature.

5.3 Subset of values with high levels of pollution

On the other hand, the results are shown when nitrogen dioxide takes values higher than 180. The explanation of the first prediction and the explanations for the complete test data set, in the same way that it has been performed for the other scenario, is show in the figures 5.6 and 5.7.

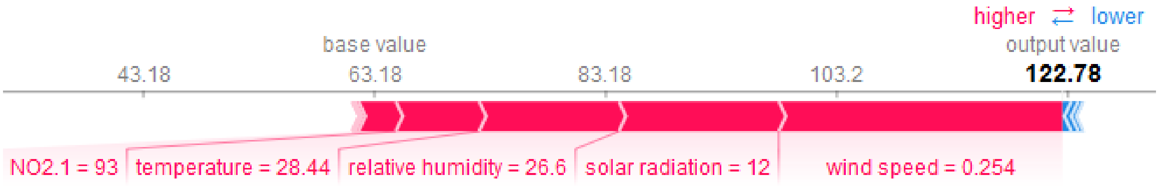

Figure 5.6: The first prediction of NO_2 when pollution data takes high levels generated by the LSTM model was explained using Deep SHAP.

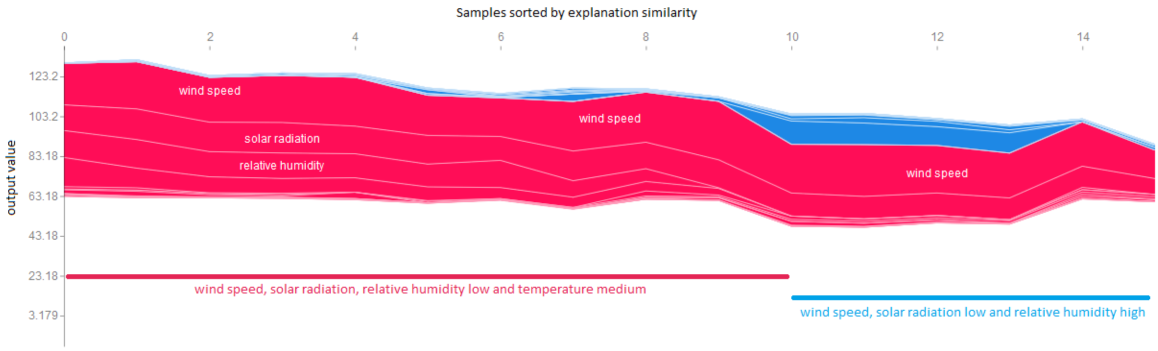

Figure 5.7: LSTM model with the attributions of SHAP values in the pollution time series identifies among 16 instances that share similar reasons for the concentration of NO_2. NO_2 prediction when pollution data takes high levels generated by the LSTM model was explained using Deep SHAP.

The results of the global explanations through SHAP summary plots can be seen in the figure 5.8. The effect of wind speed, solar radiation and relative humidity features are analysed using SHAP dependence plots are shown in the figures 5.9 and 5.10.

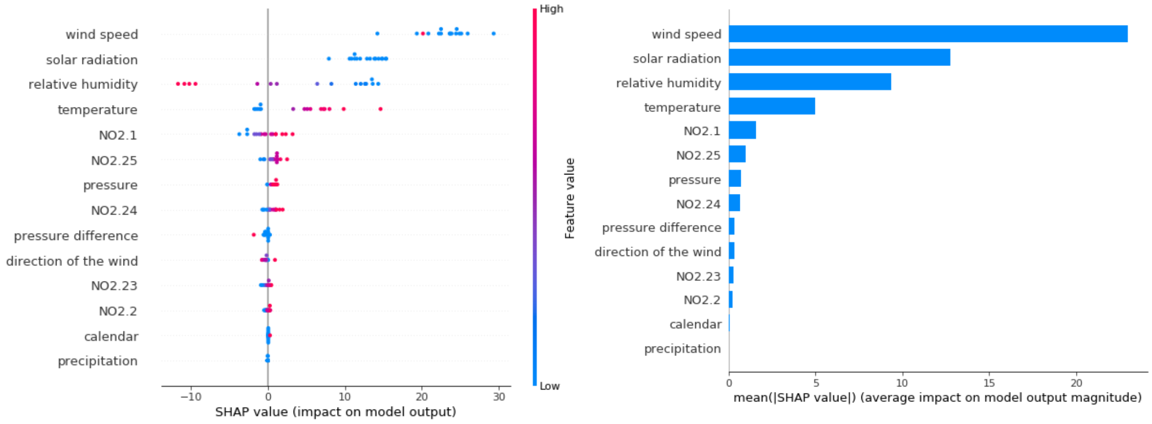

Figure 5.8: a) SHAP summary plot of a LSTM model with 14 features of the pollution time series when NO_2 takes high levels. b) Corresponds to the same graph as a), but taking the average absolute value of the SHAP values for each feature, obtaining a bar graph.

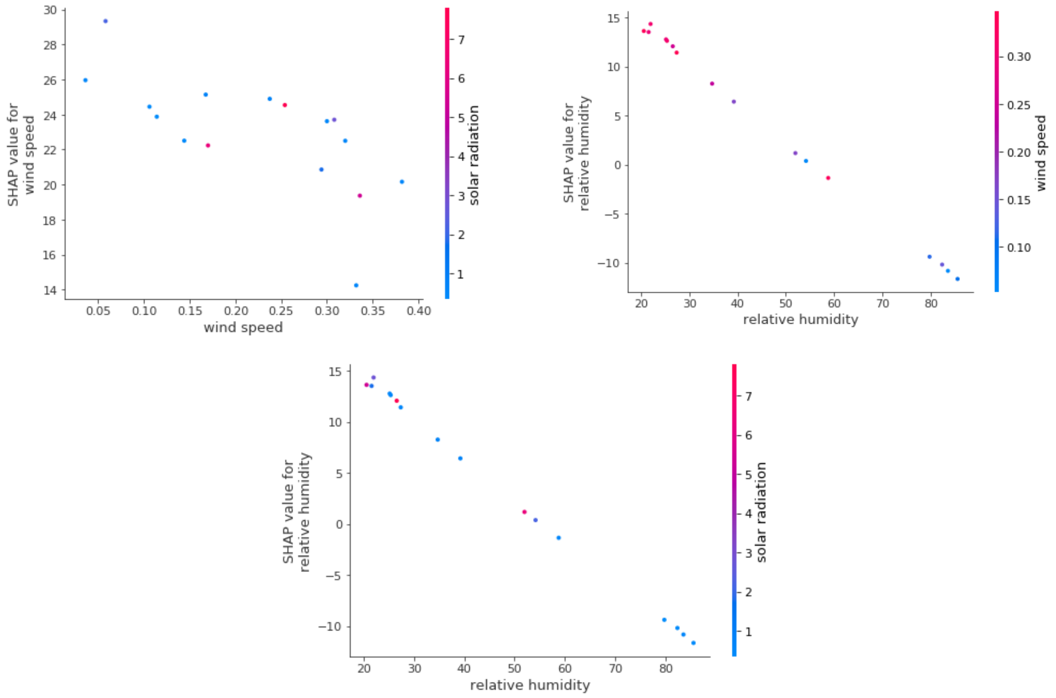

Figure 5.9: SHAP dependence plots of a LSTM model of the pollution time series when NO_2 takes high levels.

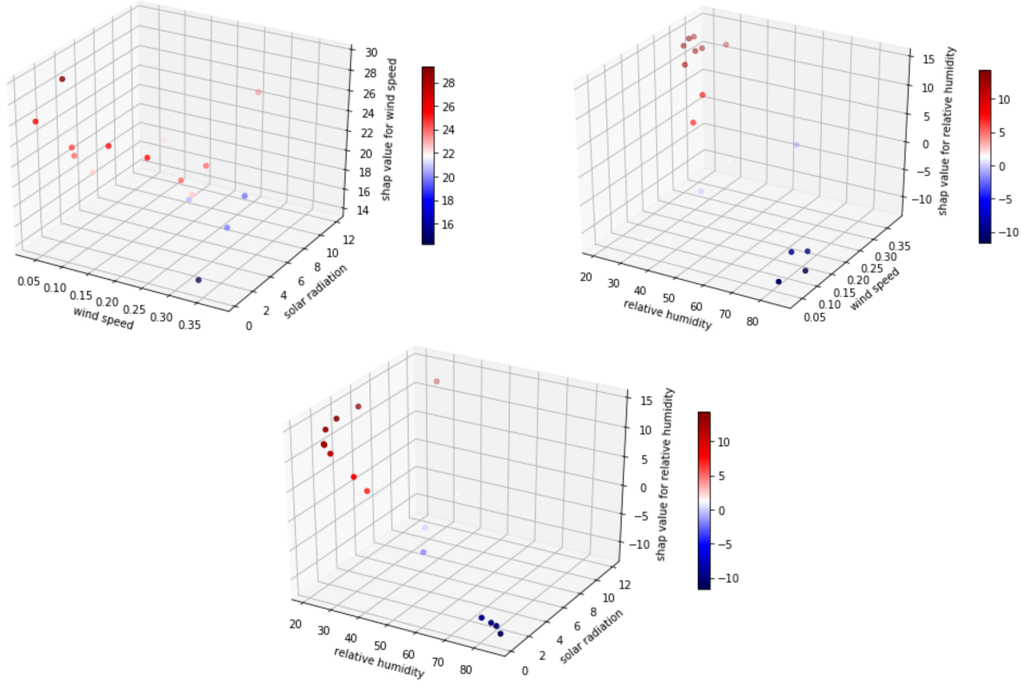

Figure 5.10: SHAP dependence plots on three dimensions of a LSTM model of the pollution time series when NO_2 takes high levels.

Chapter 6

General discussion

This chapter interprets the explanations based on SHAP of the LSTM model outputs obtained through the Deep SHAP explainer. Explanations were shown in a some plots in the previous chapter 5 to help see the effects that the features have on the prediction. Understanding the results of model output for both scenarios, we could know how the model behaves for the data of the pollution time series and if there are differences when high levels of pollution occur in the city of Madrid. We finish with the conclusions obtained from this study, the ethical and social implications of the research and the future work.

6.1 General discussion

Explanation results of the SHAP indicators, using Deep SHAP, obtained in chapter 5 are commented. First, the results locally and subsequently globally are analysed. In this way we get a complete view of why the LSTM model predicts that NO_2 value, both for the complete test data set and for the data set when NO_2 takes high levels.

SHAP value attributed to the features locally can be seen in the figures 5.1, 5.2, 5.6 and 5.7. Force plots show that the base value, i.e, the average model output over the training dataset we passed, is equal to 63.18. On the one hand in the figure 5.1, LSTM model predicts a first value of NO_2 equal to 72.13. Observed value for that instance it is equal to 111. On the right side, the blue features are those that push towards lower values, while on the left side, the red features try to increase the value. Wind speed (0.156) and solar radiation (46.6) cause the value of the prediction of NO_2 to increase, while relative humidity (76.6) decreases it, therefore it is expected that a medium value will be obtained for this observation. We can see some of the most notable subgroups of features in the figure 5.2. When wind speed, solar radiation and relative humidity features take low values, LSTM model push high NO_2 levels. However, when wind speed and relative humidity features take high values, LSTM model push low levels of NO_2.

On the other hand in the figure 5.6, LSTM model predicts a value equal to 122.78. Observed value for that instance it's equal to 184. Wind speed (0.254) and solar radiation (12) cause the value of the prediction of NO_2 to increase. Therefore it is expected that a high value will be obtained for this observation. We can see some of the most notable subgroups of features in the figure 5.7. When wind speed, solar radiation and relative humidity features take low values and temperature takes medium values, LSTM model push high NO_2 levels. However, when relative humidity feature take high values, LSTM model push low levels of NO_2. In both cases, when SHAP value attributed of wind speed and solar radiation features take positive values, NO_2 prediction value takes a higher value.

SHAP value attributed to the features globally can be seen in the figures 5.3 and 5.8. In both examples, all features are continuous, wind speed feature has the maximum impact of the prediction and a high values of solar radiation and relative humitidy features can be very significant, otherwise the model ignored the calendar and precipitation features. LSTM model shows a high degree of non-linearity. The impact of wind speed, solar radiation and relative humidity features extends over a relatively wide range. On the one hand, in the figures 5.3 and 5.8 can be seen that high values of the wind speed, solar radiation and relative humidity features are associated with negative SHAP values attributed in the independent variable, i.e, NO_2 decreases when the values of the wind speed, solar radiation and relative humidity feature are high. On the other hand, high values of the temperature, NO2.24, NO2.25 and direction of the wind features are associated with positive SHAP values attributed in the independent variable, i.e, NO_2 increases when the values of the temperature, NO2.24, NO2.25 and direction of the wind are high. In addition, the calendar and precipitation features do not have any impact on the independent variable, because most of the SHAP values attributed are equal to zero.

We can see how wind speed, solar radiation and relative humidity features affect the LSTM model output in the figures 5.4, 5.5, 5.9 and 5.10. In both cases the same conclusions are obtained. The first example shows on the x-axis the wind speed, on the y-axis the solar radiation and on the z-axis the SHAP value attributed to the wind speed. Higher SHAP value attributed to the wind speed represent a higher concentration of NO_2 levels due to the absence of wind speed and the presence of medium values of solar radiation. Low levels of NO_2 are associated with periods when the wind is high. The second example shows on the x-axis the relative humidity, on the y-axis the wind speed and on the z-axis the SHAP values attributed to the relative humidity. Higher SHAP values attributed to the relative humidity represent a higher concentration of NO_2 levels due to the presence of low values of relative humidity and wind speed. Low levels of NO_2 are associated with periods where the percentage of relative humidity is high. Finally, the last example shows on the x-axis the relative humidity, in the y-axis the solar radiation and in the z-axis the SHAP value attributed to the relative humidity. Higher SHAP value attributed to the relative humidity represent a higher concentration of NO_2 levels due to the presence of low values of relative humidity and medium values of solar radiation. Low levels of NO_2 are associated with periods when the percentage of relative humidity is high and where solar radiation takes low values.

6.2 Conclusions

Concerns about the growing increase in pollution levels and the problems it cause, have increased in recent years in developed countries. Knowledge about the implications of having high pollution levels in cities has increased. Previously it was thought that the implications were not so serious. While it is difficult to act on this problem, because pollution increases or decreases due to several factors, it is necessary to apply preventive measures to reduce pollution levels.

Knowing in advance the concentrations of NO_2 allows to apply efficiently preventive measures, such as restricting traffic in areas where higher levels of NO_2 were concentrated. Several studies address the problem of forecasting NO_2 concentrations, using traditional techniques and more current techniques such as deep learning models, however there is still a gap between the information provided by the model and the needs of end users. To cover this gap, not only an accuracy model is needed but also a model that provides interpretable results on the levels of NO_2 and their causes.

We have seen in chapter 2 that the European Union has established an anti-pollution protocol. The protocol establishes three levels of action in relation to the average concentrations of NO_2 per hour. The first is set when the threshold exceeds 180 $\mu g/m^3$, the second when it exceeds 200 $\mu g/m^3$ and an alert when the values are higher than 400 $\mu g/m^3$.

Traditionally, models developed in pollution studies use statistical tools, such as time series analysis. More recent studies use computational intelligence approaches such as artificial neural networks to predict risk levels of NO_2 concentrations. Each approach has its advantages and disadvantages, some models depend a lot on the parametrization (traditional models) to those whose interpretation of the results cannot be easily interpreted (artificial neural networks). As a result of the first part of the investigation, a LSTM model was selected for its performance and robustness versus the collinearity of variables, avoiding discarding expensive features or parametrizations as with traditional models. In addition, LSTMs capture the temporal dependence of the variables and are suitable when the time window is used, allowing a more robust approach against overfitting compared to other models. Several studies make assumptions about the influence of meteorological parameters and pollution levels. Given the robustness of the LSTM model versus the collinearity of variables, all the available influential features are used as input parameters to the system. Most studies predict NO_2 concentrations and we did so using a LSTM model for regression. From here, interpreting the LSTM model outputs can be expensive for some users because these types of models are difficult to understand.

The best explanation to interpret it is the model itself, however when we work with complex models the same model does not work for us. In the literature you can find several explanation methods. Some of these methods focus on interpreting the model outputs globally, however explaining the outputs globally can be complicated when three dimensions are exceeded. Local explanations are more precise and interpretable than global ones, which is why agnostic methods or methods based on explanations are studied. This document studies in depth three explanation methods based on

SHAP to interpret the results obtained from the LSTM model. It is decided to use this type of explainer because are part of the additive feature attribution methods and have a single unique solution in this class that meets the local precision, missingness and consistency properties. Previous methods violated one or two of these three properties. The tool used to implement the LSTM model and explanation method based on SHAP, was python. This programming language provides libraries that allow us to implement the code in a simple and easily understandable way by any user. The only drawback was in the implementation of the Kernel SHAP explainer with the LSTM model, because this functionality was not yet available in the library used. Deep SHAP explanation method is used to interpret the LSTM model prediction because it could be computationally adapted to our model and because it presented better performance than the Gradient SHAP explanation method. Therefore, we obtain two of the proposed objectives. We have selected a deep learning model that obtains good precision, is scalable and can be interpreted by an explanation method, as well as giving a framework to identify the most influential values in the output of the prediction model through a local explanation method.

Research presented in this document addresses the prediction of NO_2 concentrations in the air and its interpretation. Two sets of data were compared to analyse if there were significant differences between the complete data set and the subset of data when high pollution levels are present. However, both sets present the same results. Wind speed feature has the greatest impact on the prediction. Wind speed is inversely proportional to the concentration of NO_2 levels. When wind speed is low, SHAP values attributed of the wind speed feature assume positive values, which prompts LSTM model to predict a high NO_2 values. In the same way, solar radiation and relative humidity features behave, which are also very significant in the prediction value. Temperature, NO2.24, NO2.25 and direction of the wind features are directly proportional to the concentration of NO_2 levels. When these variables take low values, SHAP values attributed of each feature take negative values, which drives the LSTM model to predict a low NO_2 values. Finally, model did not take into account the calendar and precipitation features. Therefore, this proposal helps to provide information on the features influence the LSTM model output and the differences or similarities between both scenarios getting the last objective stated.

Although we have not obtained significant deferences between one data set or another, this analysis helps us to understand why the LSTM model makes those predictions. We can see that the LSTM model outputs are not influenced by different features when high levels of pollution are present, but for this case as well as when there are more moderate levels, the wind speed is important to determine that the levels of pollution increase. From here, knowing the factors that influence the model outputs, we can take preventive measures to reduce the concentrations of NO_2 and understand why pollution protocols are activated in the city of Madrid. As a result of this study, a short version of this document has been prepared and submitted for consideration in the Ecological Informatics journal.

6.3 Ethical and social implications

A good technological application not only has to be useful or commercially viable, but it must also use the data in an ethical and responsible way. Many times users do not trust the data products they use, because they feel abused because of the false and misleading content show them. Recovering trust in users is usually complicated, so the only solution is to be reliable. To regain user confidence, in Patil *et al.* (2018) a golden rule for data is suggested as a starting point. However, the golden rule is not enough by itself. The five C's are used: consent, clarity, consistency, control and consequences as a framework to implement this rule for data. Let's look at the ethical and social implications that our research presents through the five C's:

- Pollution and meteorological data used in our research are public as discussed in chapter 2. By being public, Madrid city council has given its consent to collect and use the data. Therefore, the data used in our investigation have a consent.

- In this document we develop a framework to interpret the deep learning model outputs of the pollution time series so that we can understand how the model behaves, so the data provided by the Madrid city council is clearly used to research purposes, providing information to reduce pollution levels in Madrid.

- Explaining the prediction model outputs using SHAP-based explanation methods provide consistency and confidence to users because methods make them understand how the model works.

- Interpreting the predictions of a complex model allows us to obtain control and transparency of how the model behaves on pollution data so that users can understand its operation and the factors that make this automated decision occur.

- This study is designed to add value to the user by giving a clearer view of how the prediction model behaves allowing more appropriate preventive measures to be taken to reduce pollution levels.

Through this research, we obtain a guideline that can help us to reduce pollution levels by decreasing environmental and health impacts, guaranteeing our application does not cause any damage since it follows the five C's.

6.4 Future work

Novel methods used in this field and the results produced from the research are related to the possibility of being able to interpret more accurate prediction models that are contributing to the knowledge of the air quality field. When studying explanation methods about complex models, and given the results obtained, the proposal is of interest to institutions and researchers to plan in advance the impact of high NO_2 concentrations, so that preventive measures can be taken. Aiming this research to the institutions, it would be interesting to apply these explanation methods on the prediction models used to activate the anti-pollution protocols, so that the model can be improved based on the results of the model outputs. In addition, by adding more information to the prediction model, such as including information on traffic or gas emissions generated by older boilers, two points to explore can be established; (i) how these factors influence the prediction model output (ii) understand what are the factors that really have the greatest impact on the levels of NO_2, providing greater knowledge about the pollution behaviour.

A framework for applying SHAP-based explanation methods on a complex time series prediction model was provided. Explanation methods extend and support knowledge about the influence of meteorological factors and pollution episodes, showing the most influential variables in the model outputs. Although more research is required on the influence of other features on the model outputs that may be relevant. Including new factors and generating features of them will surely increase the accuracy of this proposal, especially when the NO_2 concentrations exceed the established thresholds since, as we have seen in chapter 5, only with the meteorological variables there are no differences in the model outputs for the different pollution scenarios presented.

This line of research is a promising and interesting topic that was promoted by the author of this work. This research presents a new application of explanation methods to understand why NO_2 concentrations increase or decrease so that we can anticipate to take appropriate measures. In addition, it supports the knowledge of the influence of meteorological factors with the levels of NO_2. The results are promising, but pollution levels remain high in large cities. That is why it is important to continue studying the NO_2 concentrations, with the aim of providing better results in order to reduce pollution levels. It is hoped that it will contribute to increase knowledge in this field, providing useful information to improve people's living conditions and try to remedy climate change.

Bibliography

Arain, M. A., Blair, R., Finkelstein, N., Brook, J. & Jerrett, M. (2009). Meteorological influences on the spatial and temporal variability of no2 in toronto and hamilton. *The Canadian Geographer/Le Géographe canadien* **53**(2), 165–190.

Aznarte, J. L. (2017). Probabilistic forecasting for extreme no2 pollution episodes. *Environmental Pollution* **229**, 321 – 328. URL `http://www.sciencedirect.com/science/article/pii/S0269749116321480`.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Muller, K.-R. & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* **10**(7), e0130140.

Ballester Diez, F., Tenias, J. M. & Perez-Hoyos, S. (1999). Efectos de la contaminacion atmosferica sobre la salud: una introduccion. *Revista Espanola de Salud Publica* **73**, 109–121.

Binder, A., Montavon, G., Lapuschkin, S., Müller, K.-R. & Samek, W. (2016). Layer-wise relevance propagation for neural networks with local renormalization layers. In: *International Conference on Artificial Neural Networks*. Springer.

Commission, E. (2008). Air quality standards according to directive 2008/50/ec. ferro, c.a.t., stephenson, d.b., 2011. deterministic forecasts of extreme events and warnings. in: Jolliffe, i.t., stephenson, d.b. (eds.), forecast verification. john wiley and sons, ltd, pp. 185e201. url http://dx.doi.org/10.1002/9781119960003.ch10/summary.

Cuchi, A., Wadel, G. & Rivas, P. (2010). Cambio global espana 2020/50. *Sector Edificacion* .

Datta, A., Sen, S. & Zick, Y. (2016). Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In: *2016 IEEE symposium on security and privacy (SP)*. IEEE.

de Madrid, A. (2015). Sistema de vigilancia de la calidad del aire. url http://www.mambiente.munimadrid.es/sica/scripts/index.php.

de Madrid, A. (2018). Protocolo de medidas a adoptar durante episodios de alta contaminacion por dioxido de nitrogeno. url http://www.madrid.es/UnidadesDescentralizadas/Sostenibilidad/CalidadAire/Ficheros/ProtocoloSuperaNO2consol.pdf.

Faruk, D. Ö. (2010). A hybrid neural network and arima model for water quality time series prediction. *Engineering Applications of Artificial Intelligence* **23**(4), 586–594.

Gers, F. A., Eck, D. & Schmidhuber, J. (2002). Applying lstm to time series predictable through time-window approaches. In: *Neural Nets WIRN Vietri-01.* Springer, pp. 193–200.

Grover, A., Kapoor, A. & Horvitz, E. (2015). A deep hybrid model for weather forecasting. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM.

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

Hu, J., Wang, J. & Zeng, G. (2013). A hybrid forecasting approach applied to wind speed time series. *Renewable Energy* **60**, 185–194.

Hyndman, R. J. & Athanasopoulos, G. (2018). *Forecasting: principles and practice.* OTexts.

Khashei, M. & Bijari, M. (2011). A novel hybridization of artificial neural networks and arima models for time series forecasting. *Applied Soft Computing* **11**(2), 2664–2675.

Kim, B., Khanna, R. & Koyejo, O. O. (2016). Examples are not enough, learn to criticize! criticism for interpretability. In: *Advances in Neural Information Processing Systems.*

Kindermans, P.-J., Schütt, K., Müller, K.-R. & Dähne, S. (2016). Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv preprint arXiv:1611.07270* .

Kuremoto, T., Kimura, S., Kobayashi, K. & Obayashi, M. (2014). Time series forecasting using a deep belief network with restricted boltzmann machines. *Neurocomputing* **137**, 47–56.

Lipovetsky, S. & Conklin, M. (2001). Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry* **17**(4), 319–330.

Lipton, Z. C. (2016). The mythos of model interpretability. *arXiv preprint arXiv:1606.03490* .

Liu, J. N., Hu, Y., He, Y., Chan, P. W. & Lai, L. (2015). Deep neural network modeling for big data weather forecasting. In: *Information Granularity, Big Data, and Computational Intelligence.* Springer, pp. 389–408.

LIU, J. N., HU, Y., YOU, J. J. & CHAN, P. W. (2014). Deep neural network based feature representation for weather forecasting. In: *Proceedings on the International Conference on Artificial Intelligence (ICAI)*. The Steering Committee of The World Congress in Computer Science, Computer . . . .

LUNDBERG, S. M. & LEE, S.-I. (2017). A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems*.

LV, Y., DUAN, Y., KANG, W., LI, Z. & WANG, F.-Y. (2014). Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems* **16**(2), 865–873.

MILLER, T. (2018). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* .

MOLNAR, C. *et al.* (2018). Interpretable machine learning: A guide for making black box models explainable. *E-book at¡ https://christophm. github. io/interpretable-ml-book/¿, version dated* **10**.

PATIL, D., MASON, H. & LOUKIDES, M. (2018). Ethics and data science.

RIBEIRO, M. T., SINGH, S. & GUESTRIN, C. (2016). Why should i trust you?: Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM.

ROBNIK-ŠIKONJA, M. & BOHANEC, M. (2018). Perturbation-based explanations of prediction models. In: *Human and Machine Learning*. Springer, pp. 159–175.

SHAP (2019). Lundberg. url https://github.com/slundberg/shap.

SHRIKUMAR, A., GREENSIDE, P. & KUNDAJE, A. (2017). Learning important features through propagating activation differences. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org.

STRUMBELJ, E. & KONONENKO, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems* **41**(3), 647–665.

SUNDARARAJAN, M., TALY, A. & YAN, Q. (2017). Axiomatic attribution for deep networks. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org.

SUTSKEVER, I., VINYALS, O. & LE, Q. V. (2014). Sequence to sequence learning with neural networks. In: *Advances in neural information processing systems*.

TUTIEMPO (2019). Capa límite. url https://www.tutiempo.net/meteorologia/capa-limite.html.