International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com || Volume 11 - Issue 04 || April 2025 || PP. 09-25

Analysis of the Behavior of the Ozone Time Series in México City Using Machine Learning Trend 2010 - 2024

M. Sc. Zenteno Jiménez José Roberto
Geophysical Engineering, National Polytechnic Institute, México City,
ESIA-Unidad Ticóman Mayor Gustavo A. Madero
Email: jzenteno@ipn.mx

Abstract: This new study includes an analysis and forecast of the time series of daily ozone maxima for 2024 and daily concentrations, plus a trend from 2010 to 2024 for the case of daily ozone maxima in Mexico City, Machine Learning and Deep Learning algorithms are used to study the behavior, as well as the classic ARIMA method for evaluation of the forecast model. RMSE, MSE, MAE and MAPE were used.

Keywords: Machine Learning, Deep Learning, Arima, Time Series, Ozone, Bivariate Probability Distributions, Stochastic Gaussian Mixture, Neural Networks.

Introduction

Ozone in México City: An Invisible Enemy

Mexico City, one of the world's largest metropolises, faces a persistent environmental challenge: high concentrations of tropospheric ozone. Unlike stratospheric ozone, which protects us from ultraviolet radiation, tropospheric ozone, at ground level, is a pollutant harmful to health. Tropospheric ozone originates from chemical reactions between nitrogen oxides (NOx) and volatile organic compounds (VOCs) in the presence of the intense solar radiation that characterizes the Mexican capital. Vehicles, industry, and other human activities are the main emitters of these precursors. Mexico City's geographic location, surrounded by mountains and volcanoes, makes it difficult for pollutants to disperse. Furthermore, during the hot, dry season (February to June), the lack of wind and low humidity favor ozone accumulation.

Figure 1. Map of México City, source Google Maps

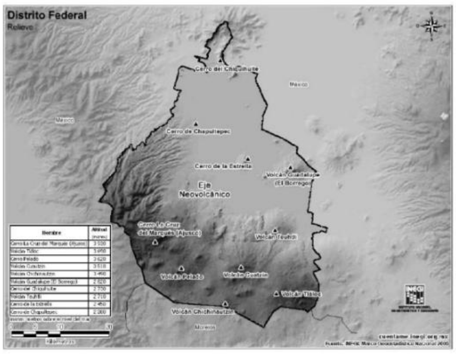


Figure 2. Map of México City, source INEGI

Health and Environmental Effects

Exposure to high levels of ozone can cause respiratory problems, eye and throat irritation, and aggravate diseases such as asthma. Furthermore, ozone damages vegetation and reduces agricultural productivity.

Actions to Combat Ozone

To mitigate this problem, various measures have been implemented:
"No Driving Today" Program: Restricts vehicle traffic based on license plate number.

Environmental Contingencies: These are activated when ozone levels exceed established limits, with additional restrictions on traffic and industrial activities.

Stricter Regulations: These seek to reduce emissions from vehicles and industries.

Promoting sustainable mobility: The use of public transportation, cycling, and walking is encouraged.

The fight against ozone requires everyone's participation. Reducing car use, opting for sustainable transportation alternatives, and supporting environmental policies are key actions to improve the quality of the air we breathe.

Artificial Intelligence to the Rescue: Machine Learning and Ozone in México City (2010-2024)

Ozone pollution in México City is a complex problem that requires innovative solutions. In this article, we explore how machine learning algorithms enable time series analysis, such as that of ozone, allowing for a deeper understanding of its patterns and more accurate prediction of its behavior in discrete time. Machine learning algorithms are capable of analyzing large volumes of data and detecting complex patterns that would be difficult to identify with traditional statistical methods.

Unraveling Ozone: Statistical Methodologies for Time Series Analysis in Mexico City. Mexico City, with its complex atmospheric dynamics, presents a continuing challenge in the study of air quality. Tropospheric ozone, a secondary pollutant, requires detailed analysis of its temporal patterns to understand its behavior and take effective measures. In this article, we explore key statistical methodologies for analyzing the ozone time series in the Mexican capital, with a focus on the trend from 2010 to 2024.

Time series analysis allows for the identification of patterns, trends, and seasonality in ozone data over time. This is important for:

Understanding the evolution of ozone: Identifying whether ozone levels have increased, decreased, or remained stable.

Evaluating the effectiveness of public policies: Determining whether implemented measures have had an impact on ozone reduction.

Forecasting: Predicting future ozone levels and anticipating potential episodes of high pollution.

Methodologies

Classical Methodologies

To analyze the time series of daily ozone maximums and daily concentrations, the following will be used:

ARIMA Model:

Autoregressive integrated moving average (ARIMA) models are widely used to model and forecast time series.

Data quality and availability are essential for obtaining reliable results and are the basis for the official México City website: http://www.aire.cdmx.gob.mx/default.php

It is important to consider the influence of external factors, such as extreme weather events or changes in public policy.

The interpretation of the results must be done in the context of México City's complex atmospheric dynamics.

Machine Learning Algorithms

Recurrent Neural Networks (RNNs) from Deep Learning:

These networks are especially effective for modeling sequential data, such as time series.
They can capture long-term dependencies in the data, allowing for more accurate predictions of ozone levels.

Random Forest Models:

These models are robust and can handle data with noise and missing values.
They are useful for identifying the most important variables influencing ozone concentration.

Gaussian Mixture Model (GMM):

Unsupervised Learning:

GMMs are primarily used for data clustering, which is a type of unsupervised learning. This means that the algorithm works with unlabeled data, trying to find patterns and inherent structures within the data itself.

GMMs are probabilistic models, meaning they describe the probability distribution of the data. They assume that data points are generated from a mixture of several Gaussian (normal) distributions.

Development

The first thing we do is clean the data of unread or missing values in their respective analysis. We then work with the 365 x 45 matrix of daily ozone maximums from all stations in Mexico City and the Metropolitan Area and the daily concentrations, which are 8,776 data points, using an 8,776 x 45 matrix, obtaining a mean from this data set.

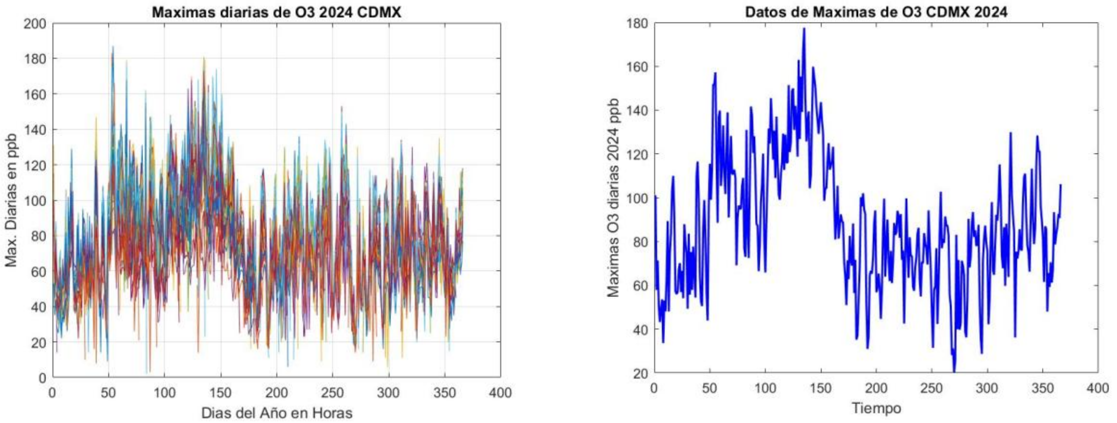


Figure 3. Average of daily highs

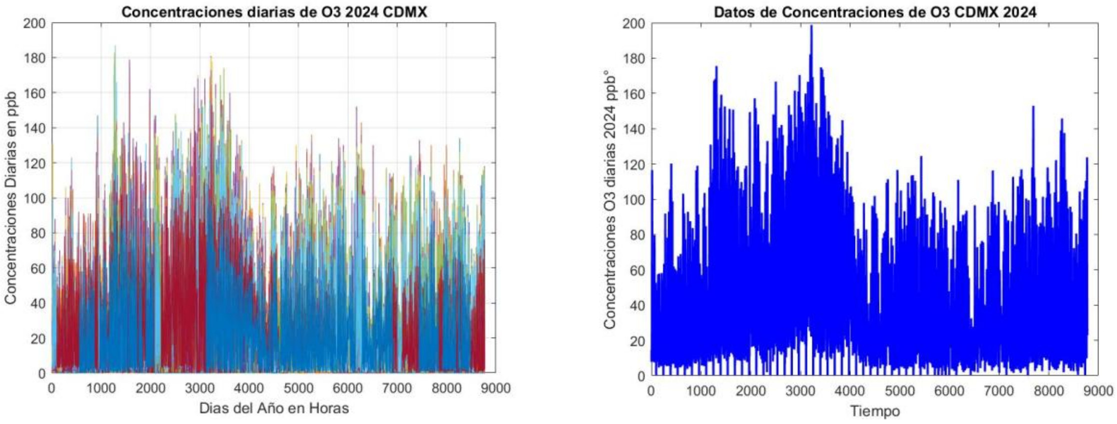


Figura 4. Average of daily concentrations

Statistical data

Daily O3 concentrations in ppb		Daily O3 maximums in ppb	
Mean:	82.58	Mean:	87.96
Sdt:	70.62	Sdt:	29.77
Min:	0.100	Min:	20.33
Max:	198.8	Max:	177.6
Var:	4987.18	Var:	886.51
Number of data: 8776		Number of data: 365	

Gaussian Mixture Methodology (GMM)

We begin by applying the Gaussian Mixture Methodology (GMM), which uses a primary bivariate analysis, using the 2022-2023 and 2023-2024 trend clusters, as in previous work with daily maxima due to their normal behavior.

Mixture of Gaussians:

A model that assumes that observed data are generated by a weighted combination of several Gaussian distributions. Each Gaussian distribution is called a "component" of the mixture. Each Gaussian component in the mixture has its own mean, variance, and weight. The weights represent the probability that a data point belongs to that component.

International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com || Volume 11 - Issue 04 || April 2025 || PP. 09-25

Mathematical Methodology:

Probability Density Function:

The probability density function of a GMM is defined as a weighted sum of the probability density functions of the Gaussian components:

$p(x) = \sum(k=1 \text{ to } K)\ \pi k\ N(x \mid \mu k, \Sigma k)$	Where: x is the data vector. K is the number of components. πk is the weight of component k. $N(x \mid \mu k, \Sigma k)$ is the Gaussian probability density function of component k, with mean μk and covariance matrix Σk.

EM Algorithm:

The EM algorithm is an iterative algorithm consisting of two steps:

Expectation Step (E): Calculates the probability that each data point belongs to each component.

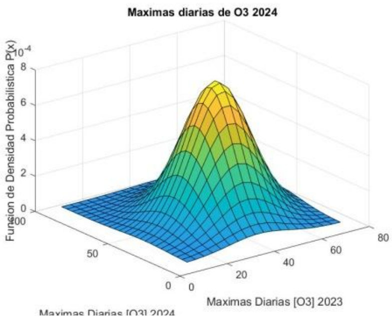
Maximization Step (M): Updates the GMM parameters (weights, means, and covariance matrices) to maximize the likelihood of the data.

General 2D Gaussian Mixture

Gaussian mixture distribution with 2 components in 2 dimensions

Component 1:
Mixing proportion: 0.500000 2022- 2023
Mean: 47.1670 45.2458

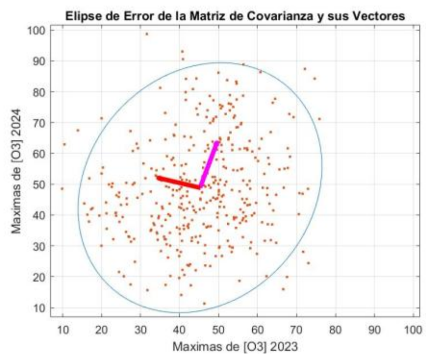
Component 2:
Mixing proportion: 0.500000 2023- 2024
Mean: 45.2458 48.8442

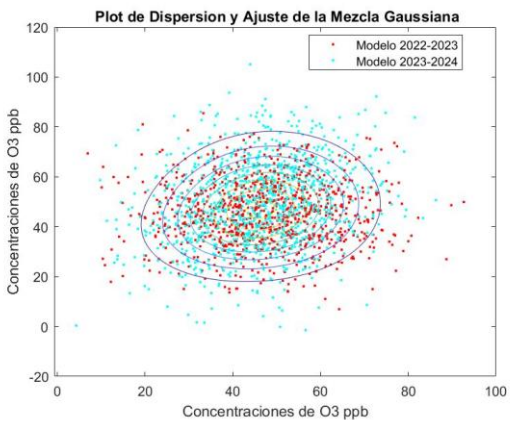


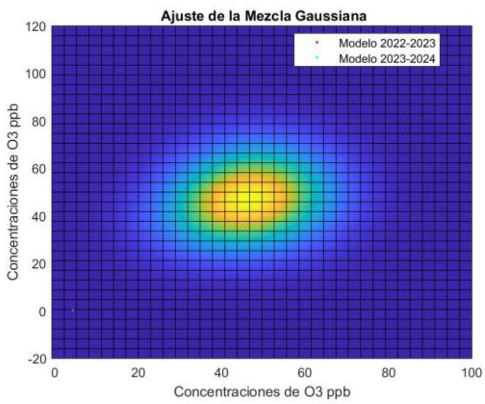
Sigma =

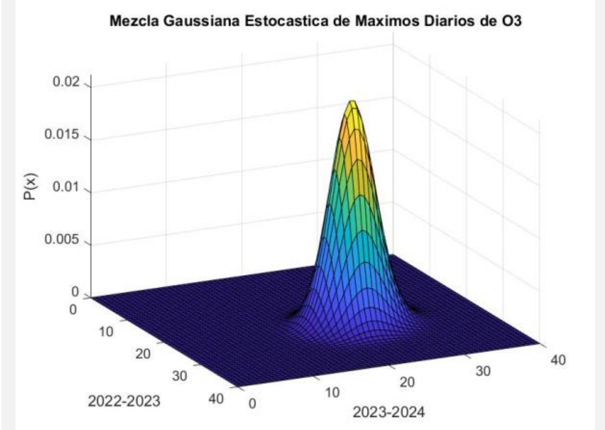
163.3078 36.0968
36.0968 274.0855

R =
1.0000 0.1706
0.1706 1.0000



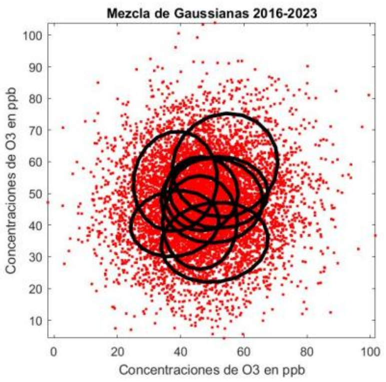


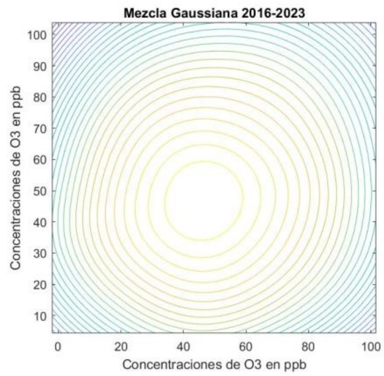


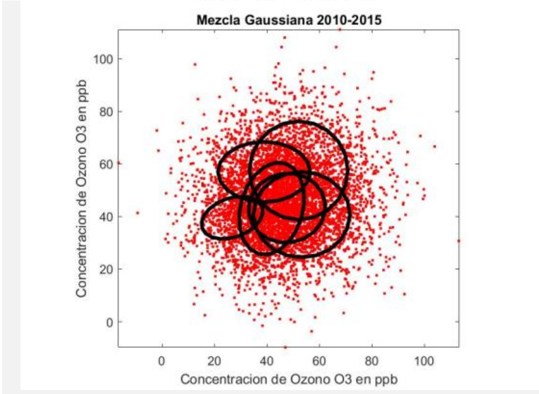


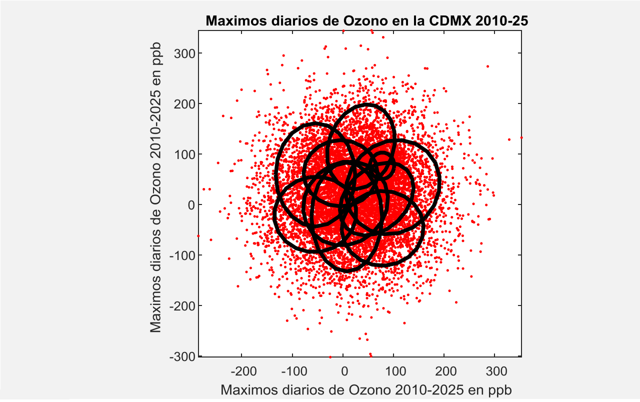
Now using the 2010 – 2024 trend with the conditions of Total Covariance and without Kmeans, [Ref http://mirlab.org/jang/books/dcpr/]

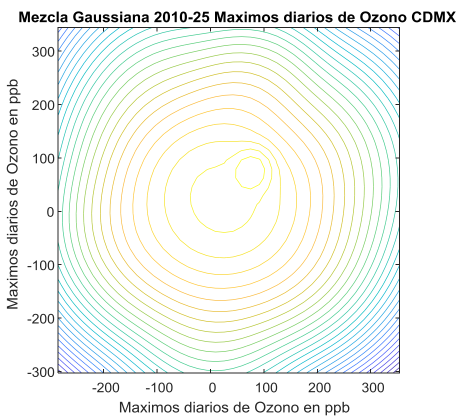
General Results











Using with Gaussian mixture 2010-2025 with the last one without Gaussian mixture only the common normal with kmean = 10

Use of ARIMA Method

ARIMA is fundamentally a statistical method designed for time series analysis and forecasting. Its main objective is to model time dependence in data, that is, how past values influence future values. It is based on statistical concepts such as autocorrelation and stationarity.

This component models the linear dependence between an observation and a certain number of lagged observations. It is denoted as AR(p), where "p" is the order of the autoregressive model.

Mathematically, an AR(p) model is expressed as:
$Xt = c + \varphi 1Xt\text{-}1 + \varphi 2Xt\text{-}2 + ... + \varphi pXt\text{-}p + \varepsilon t$
Where:

Xt is the value of the time series at time t.
c is a constant.
φi are the model parameters.
εt is the error term (white noise).

Diffing (I):

This component is used to make a non-stationary time series stationary. Diffing consists of calculating the differences between consecutive observations.

It is denoted as I(d), where "d" is the order of differencing.

Moving Average (MA):

This component models the linear dependence between an observation and the forecast errors of previous observations. It is denoted as MA(q), where "q" is the order of the moving average model.

Mathematically, an MA(q) model is expressed as:

$Xt = \mu + \varepsilon t + \theta 1\varepsilon t\text{-}1 + \theta 2\varepsilon t\text{-}2 + ... + \theta q\varepsilon t\text{-}q$

Where:

Xt is the value of the time series at time t.
μ is the mean of the series.
θi are the model parameters.
εt is the error term (white noise).

Methodology:

Stationarity: Check whether the time series is stationary. If not, apply differencing until it is.

Estimation: Estimate the ARIMA model parameters using methods such as maximum likelihood.

Table 1

ARIMA Daily Maximum O3 Concentrations				
Stationarity Test with O3 Maxima for 2024				

Null hypothesis: The time series has a unit root (it is not stationary).				
Test statistic (ADF): -2.0275				
p-value: 0.0410				
Critical values: -1.9415				
The null hypothesis is rejected. The time series is likely stationary.				
ARIMA (10,0,5) Model (Gaussian Distribution):				
	Value	StandardError	TStatistic	PValue
Constant	30.535	14.144	2.1588	0.030865

AR{1}	0.92392	0.33516	2.7566	0.0058401
AR{2}	-1.4419	0.33435	-4.3127	1.613e-05
AR{3}	1	0.4608	2.1701	0.029998
AR{4}	-0.68952	0.36557	-1.8862	0.059273
AR{5}	0.46996	0.29111	1.6143	0.10645
AR{6}	0.082719	0.21156	0.391	0.6958
AR{7}	0.1295	0.11582	1.1182	0.26349
AR{8}	0.045646	0.12419	0.36755	0.71321
AR{9}	0.058676	0.08526	0.68821	0.49132
AR{10}	0.070485	0.069949	1.0077	0.31362
MA{1}	-0.2455	0.34443	-0.71276	0.47599
MA{2}	1.2457	0.13711	9.0859	1.0284e-19
MA{3}	-0.083436	0.38978	-0.21406	0.8305
MA{4}	0.63545	0.1282	4.9566	7.1736e-07
MA{5}	0.041583	0.20114	0.20674	0.83621
Variance	321.34	25.772	12.469	1.1076e-35

Results

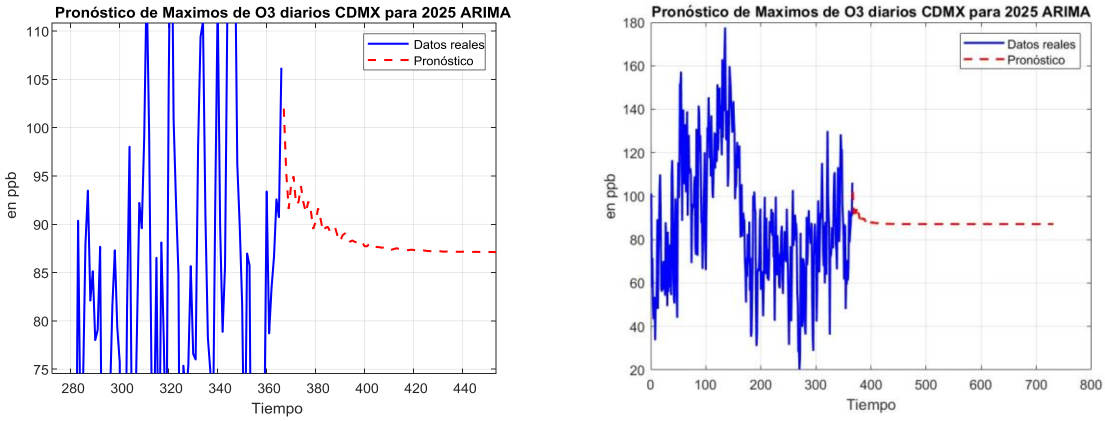


Figure 5. ARIMA 90-Day Forecast of Daily Maximum Concentrations.

Daily Ozone O3 concentrations in Mexico City

Table 2

ARIMA Daily O3 concentrations

Null hypothesis: The time series has a unit root (is not stationary).
Test statistic (ADF): -9.6446
p-value: 0.0010
Critical values: -1.9416
The null hypothesis is rejected. The time series is likely stationary.
ARIMA (10,1,5) Model (Gaussian Distribution):

	Value	StandardError	TStatistic	PValue
Constant	0.00041828	0.0020819	0.20091	0.84077
AR{1}	2	0.067913	29.449	1.2825e-190
AR{2}	-0.98463	0.16267	-6.0528	1.4231e-09
AR{3}	-0.1843	0.17714	-1.0405	0.29813
AR{4}	-0.43641	0.20141	-2.1668	0.030249
AR{5}	0.949	0.13045	7.275	3.4654e-13
AR{6}	-0.52161	0.033149	-15.735	8.6387e-56
AR{7}	0.089786	0.035182	2.552	0.010709
AR{8}	-0.086427	0.0359	-2.4075	0.016064

AR{9}	0.25784	0.033305	7.7417	9.8122e-15
AR{10}	-0.21148	0.016752	-12.624	1.5491e-36
MA{1}	-1.5979	0.068733	-23.247	1.512e-119
MA{2}	0.21109	0.13834	1.5259	0.12704
MA{3}	0.35623	0.13927	2.5579	0.010532
MA{4}	0.48776	0.1548	3.1508	0.001628
MA{5}	-0.42776	0.063801	-6.7047	2.0183e-11
Variance	36.724	0.30119	121.93	0

Results

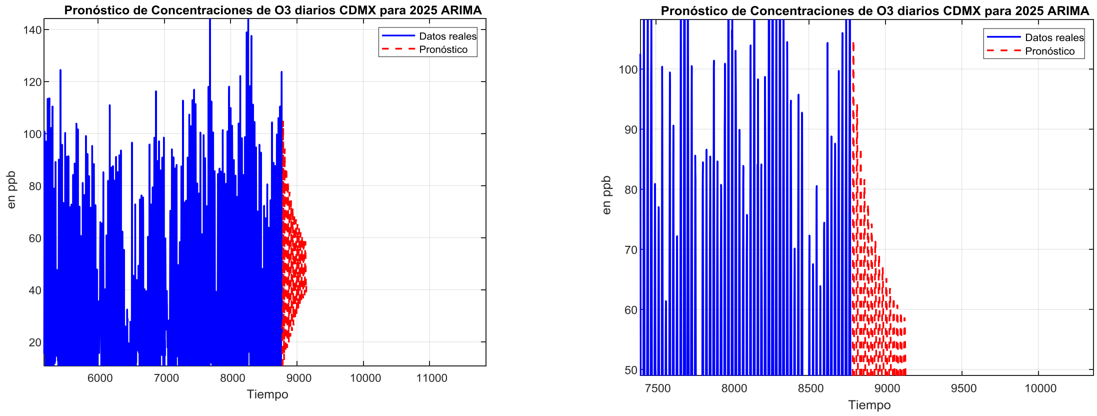


Figure 6. ARIMA's 90-day forecast of daily concentrations.

Decision Trees:

Decision trees can capture nonlinear relationships in data. They can be used to predict the next value in a time series based on previous values. The choice of algorithm will depend on the characteristics of the time series, such as the presence of trends, seasonality, and noise.

Data Preparation

First, we need to convert our time series into a format that the decision tree can understand. To do this, we will create sliding windows of data, where each window represents a set of features and the next observation is the target variable.

Decision Tree Basics

A decision tree is a machine learning model that makes decisions based on rules learned from the data.

Algorithm Steps

1. **Data Preparation:**
   - We create data "windows": we take a sequence of past values (the window) and use it to predict the next value.
   - This turns the time series into a regression problem, where we want to predict a numerical value.

2. **Tree Construction:**
   - The algorithm starts with a root node containing all the training data.
   - It then searches for the best way to split the data into two or more subsets, based on a characteristic (in our case, the past values of the time series).
   - This splitting process is repeated recursively for each subset, creating new branches and nodes in the tree.
   - The tree grows until a stopping condition is met, such as reaching a maximum depth or having a minimum number of samples at a node.

3. **Prediction:**

   a. To make a prediction, the algorithm takes a new window of data and passes it through the tree.
   b. At each node, it follows the branch corresponding to the answer to the question at that node.
   c. Finally, it reaches a leaf node, which contains the model's prediction.

In our case, the "features" the tree uses to make decisions are the past values of the time series in the window. For example, if the window size is 3, the tree can use the values from the last 3 time points to predict the next value. The tree learns patterns in the data, such as trends and seasonality, by finding the best splits that minimize prediction error.

Daily Ozone Maximums 2024 Results

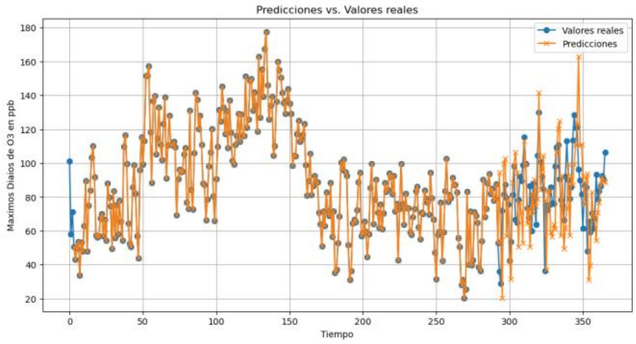


Figure 7. Capture of the behavior of the daily Ozone Maximum time series in México City in 2024

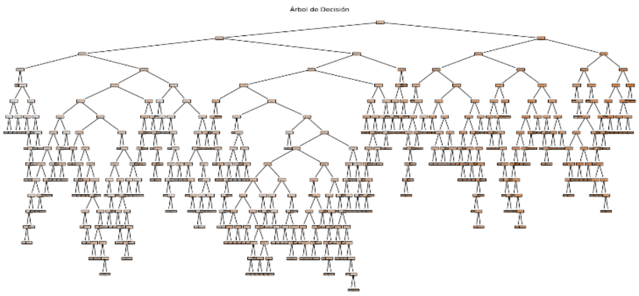


Figure 8.Decision-making for the Time Series Tree of Daily Ozone Maximums in México City in 2024

Daily Ozone Concentration Results 2024

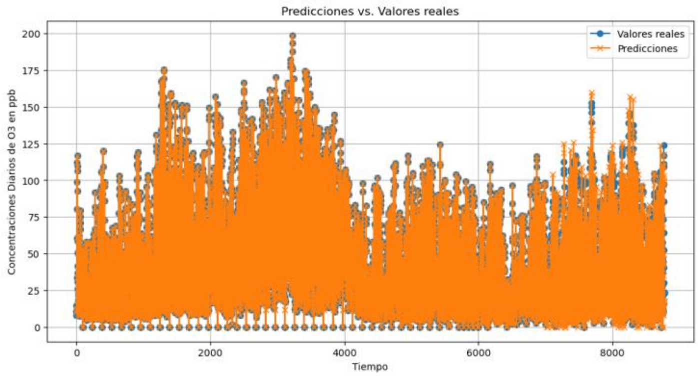


Figure 9. Capture of the behavior of the daily Ozone Concentrations time series in México City in 2024

Forecasts of daily maximum ozone concentrations in México City.

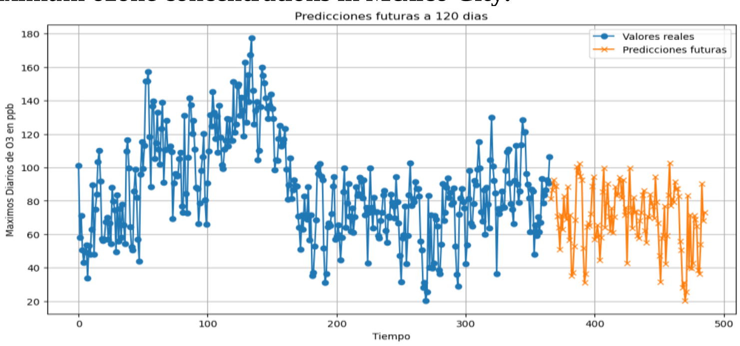


Figure 10. Forecast of the daily Ozone Maximum time series in México City from 2024 to 120 days

Table 3. Adjustment Data and Values

Values	Comments
MSE: 2.698 RMSE: 0.0859 MAE: 0.0679 MAPE: 0.1123 Scaled due to observed variability	Due to their seasonality, variability, and the presence of outliers, the time series of daily ozone maxima present higher results in the Adjustment Estimators. This is due to extreme events, such as heat waves or episodes of high pollution, which generate very high ozone peaks. These outliers can have a significant impact on the error metrics.

Daily Ozone Concentration Forecasts for México City.

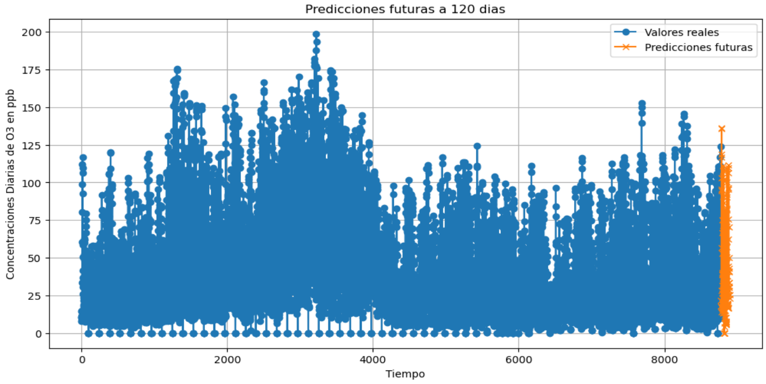


Figure 11. Forecast of the daily Ozone Concentrations time series in México City from 2024 to 120 days

Table 4. Adjustment Data and Values

Values	Comments
MSE: 0.0979 RMSE: 0.003455 MAE: 0.002743 Scaled due to observed variability	Due to their seasonality, variability, and the presence of outliers, the time series of daily ozone maxima present higher results in the Adjustment Estimators. This is due to extreme events, such as heat waves or episodes of high pollution, which generate very high ozone peaks. These outliers can have a significant impact on the error metrics.

Table 5of the Error Metrics used

Métrics	Meaning
Mean Square Error (MSE): $MSE = (1/n) * \Sigma(yi - \hat{y}i)^2$ **Root mean square error (RMSE):** $RMSE = \sqrt{[(1/n) * \Sigma(yi - \hat{y}i)^2]}$	n is the number of observations. yi is the true value of the ith observation. ŷi is the value predicted by the model for the ith observation. It is sensitive to outliers. The lower the MSE value, the better the model's performance. An MSE of zero indicates that the model perfectly predicts the true values. The closer the RMSE result is to zero, the better the prediction model.
Mean absolute error (MAE): $MAE = (1/n) * \Sigma\|yi - \hat{y}i\|$	n is the number of observations. yi is the actual value of the ith observation. ŷi is the value predicted by the model for the ith observation. |...| indicates the absolute value.

	It is less sensitive to outliers than the MSE. The closer the MAE result is to zero, the better the prediction model.
Mean Absolute Percentage Error (MAPE): MAPE = (1/n) * Σ(|(yi - ŷi) / yi|) * 100%	n is the number of observations. yi is the actual value of the ith observation. ŷi is the value predicted by the model for the ith observation. |...| indicates the absolute value. This can be problematic if the data contains values close to zero. The closer the MAPE result is to zero, the better the prediction model.

Using Deep Learning:
Recurrent Neural Networks (RNN):

Designed specifically for sequential data, such as time series. They can remember information from previous time steps.

A recurrent neural network (RNN) is a deep learning model that is trained to process and convert a sequential data input into a specific sequential data output. Sequential data is data, such as words, sentences, or time series data in our case.

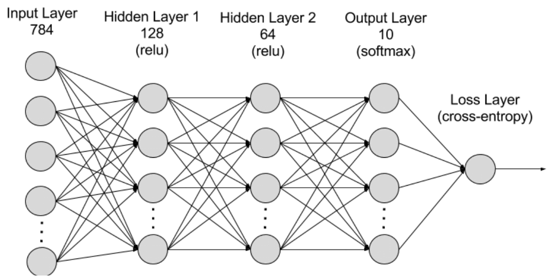


Figure 11. A recurrent neural network (RNN) (Source Internet)

RNNs work by passing the sequential data they receive to the hidden layers step by step. However, they also have a recursive or self-looping workflow: the hidden layer can remember and use previous inputs to make future predictions in a short-term memory component. It uses the current input and stored memory to predict the next sequence.

Methodology:

RNN Structure:

A typical RNN consists of a sequence of recurrent cells. Each recurrent cell receives an input and the previous hidden state, and generates an output and a new hidden state.

Forward Propagation:

In forward propagation, inputs are processed sequentially. At each time step, the recurrent cell calculates the output and the new hidden state. The RNN output can be a sequence of outputs or a single output at the end of the sequence.

Back Propagation Through Time (BPTT):

RNN training is performed using the BPTT algorithm. BPTT deploys the RNN over time and applies the standard backpropagation algorithm. BPTT calculates the gradients of the loss function with respect to the RNN parameters. It is an extended variant of the original Backpropagation (Rumelhart, Hinton et al., 1986). BPTT is based on deploying the network for eachtime interval, turning it into a network with forward connections, with shared weights

Fading Problem:

Standard RNNs can suffer from the problem of fading or gradient explosion, which makes training long sequences difficult. This can be seen in both cases of the two time series, each trained with 80% of the data in the series.

With the Daily Maximum O3 Concentrations in Mexico City trending for 2024, Extended Forecast

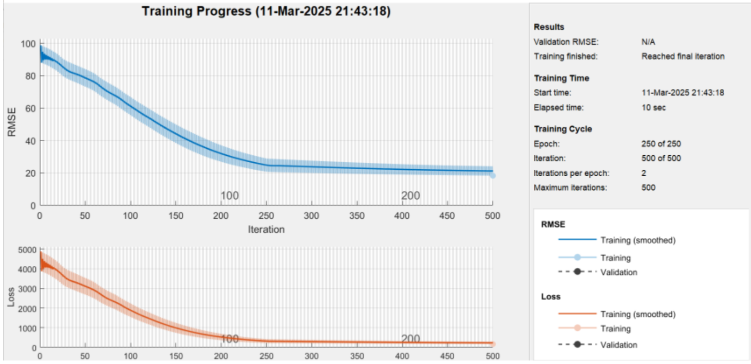
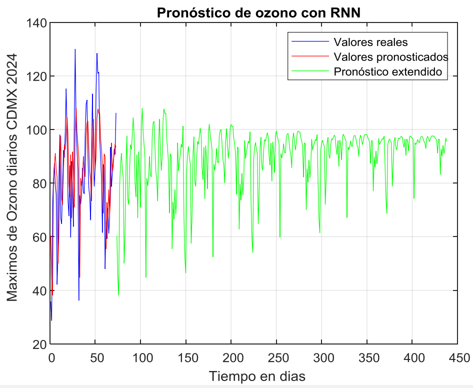
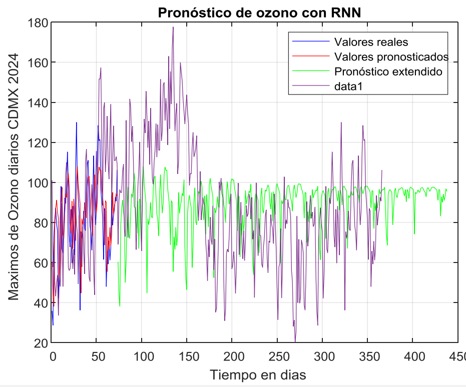


Figure 12. Training the recurrent neural network data (RNN)





Values	Comments
MSE: 0.8728 RMSE: 0.9342 MAE: 0.0395 Scaled due to observed variability	Due to their seasonality, variability, and the presence of outliers, the time series of daily ozone maxima present higher results in the Adjustment Estimators. This is due to extreme events, such as heat waves or episodes of high pollution, which generate very high ozone peaks. These outliers can have a significant impact on the error metrics. The green line is the forecast, which shows this variability over 120 days.

With the Daily Concentrations of O3 in México City 2024 for 2025

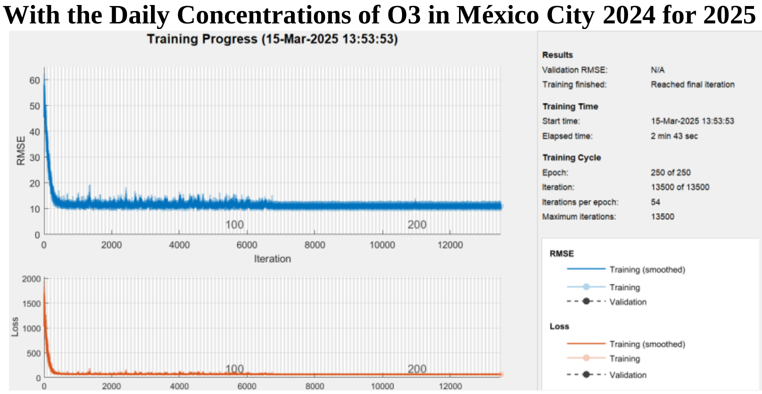


Figure 14. Training the recurrent neural network data (RNN)

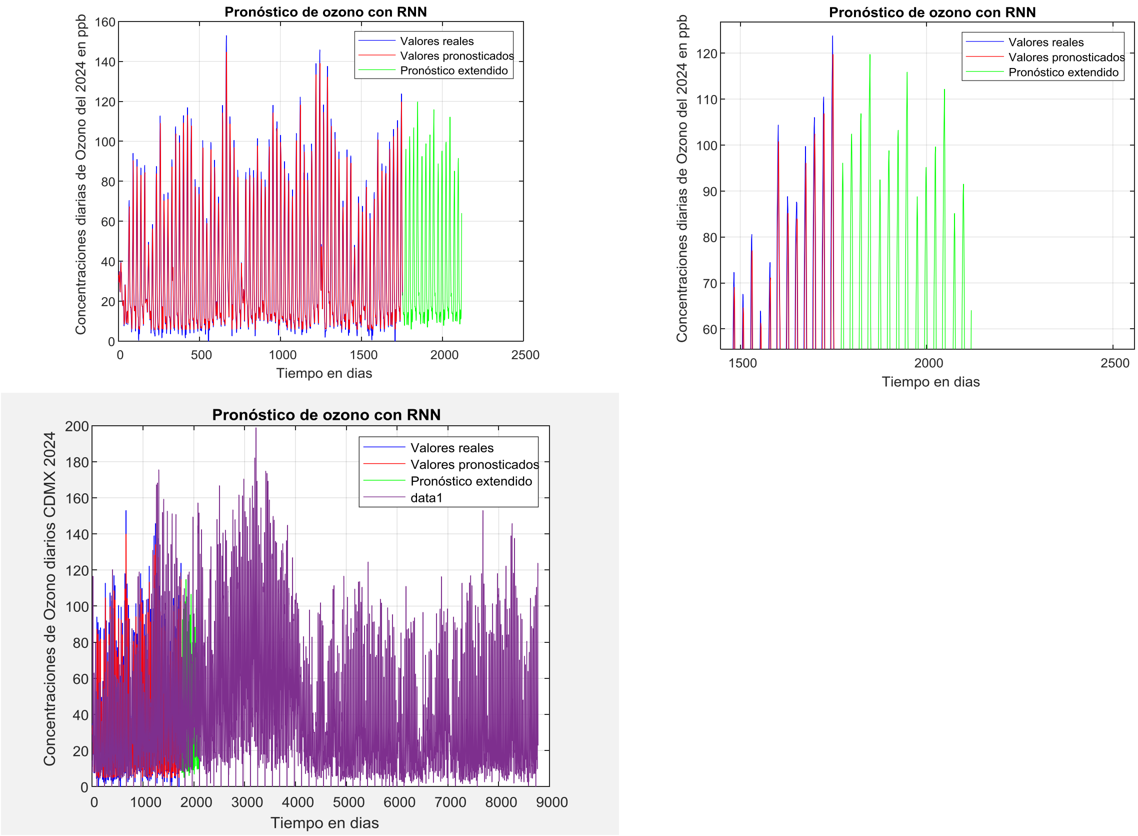


Figure 15. Results and Forecast of Recurrent Neural Network (RNN) Training Data for Daily Ozone Concentrations in Mexico City.

Values	Comments
MSE: 0.0542 RMSE: 0.2329 MAE: 7.5152 Scaled due to observed variability	Due to extreme events, such as heat waves or episodes of high pollution, which generate very high ozone peaks. These outliers can have a significant impact on error metrics. The green line is the forecast, which shows the learned pattern.

With the Daily O3 Concentrations in Mexico City from 2024 to 2025, using the latest 460 data points in the time series, with a forecast extended to 120 days from 2025.

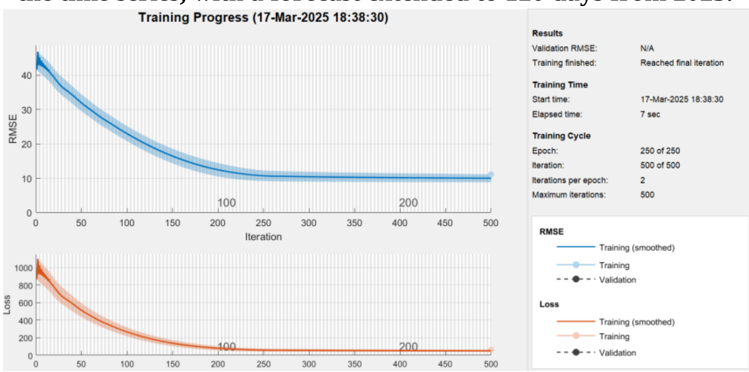
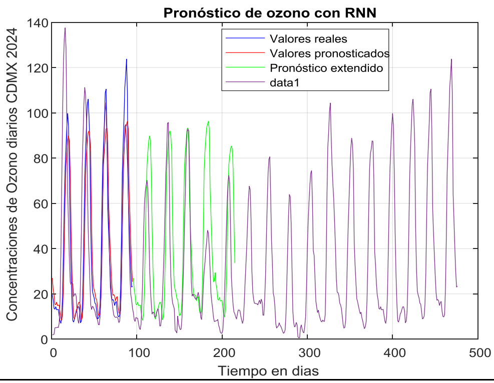
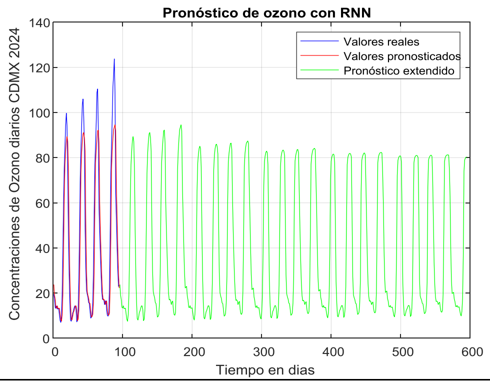


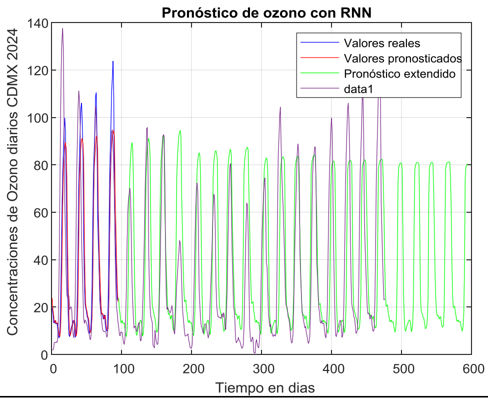
Figure 16. Training the recurrent neural network data (RNN)

Values	Comments
MSE: 0.347 RMSE: 0.589 MAE: 0.0194 Scaled due to observed variability	Due to extreme events, such as heat waves or episodes of high pollution, which generate very high ozone peaks. These outliers can have a significant impact on error metrics. The green line is the forecast, which shows the learned pattern.

Behavior and Forecast







The green curve is the extended forecast and the purple curve is the actual data.

Conclusions

We can make the following observations and conclusions: we can see that the pattern of both time series is actually being identified with the different techniques. By scaling the error metrics, this can be seen in the graphs. Now, we can also see the forecast for each one: the GMM shows an increasing trend, but the trend is falling. ARIMA shows similar behavior in the first few days for the daily Ozone Maximums, with an average of approximately 90 ppb. While the Decision Tree and RNN algorithm maintain the same trend and value for the daily Ozone concentrations based on ARIMA, given that the GMM admits Gaussian behavior, and the daily Ozone concentrations do not exhibit such behavior, but rather exponential behavior, or another pdf with that behavior, such as Gamma pdf or GEV. ARIMA provides an approximate forecasting behavior of 80 ppb, the Decision Tree with a very similar trend, and the RNN algorithm provides very similar behavior when training the data with a given number of samples from a portion of the time series.

The problem that accumulates in errors over time and outliers can also be seen. It would be necessary to segment and analyze subsets into days with high versus low concentrations for certain concentration ranges, weekdays versus weekends, and perform a detailed analysis of the errors in the residual distributions.

Software Used

Both Matlab 2020b and Python with the Anaconda 2025 Browser were used with the advanced trainNetwork algorithms that train a neural network for sequence and time series classification and regression tasks. https://la.mathworks.com/help/deeplearning/ref/trainnetwork.html for Python with Scikit-learn for integrated machine learning models.

Results until April 4, 2025 With Daily Ozone Maximums O3

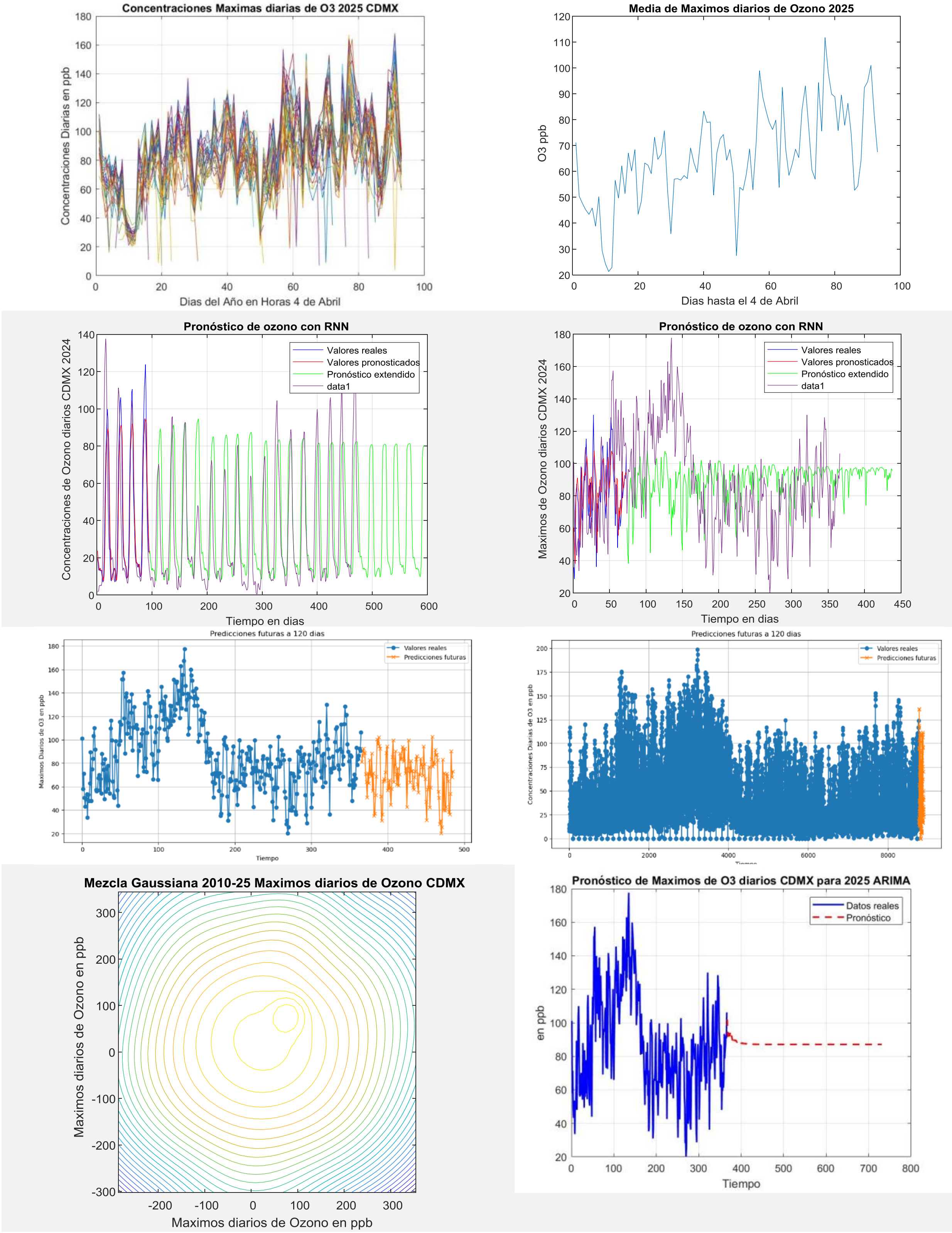


Figure 17. We can see that so far the observed average is in line with the Machine Learning techniques used, for the next 100 days and the behavior is increasing with the GMM.

Appendix

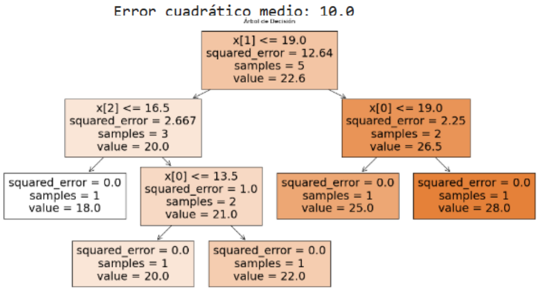
Example code for a Decision Tree

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree

def crear_ventanas(serie, tamaño_ventana):
  X, y = [], []
  for i in range(len(serie) - tamaño_ventana):
    ventana = serie[i:i + tamaño_ventana]
    etiqueta = serie[i + tamaño_ventana]
    X.append(ventana)
    y.append(etiqueta)
  return np.array(X), np.array(y)

serie_temporal = np.array([10, 12, 15, 18, 20, 22, 25, 28, 30, 32])
tamaño_ventana = 3
X, y = crear_ventanas(serie_temporal, tamaño_ventana)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

modelo = DecisionTreeRegressor()
modelo.fit(X_train, y_train)
predicciones = modelo.predict(X_test)
mse = mean_squared_error(y_test, predicciones)
print(f"Error cuadrático medio: {mse}")

# Visualizar las predicciones vs. valores reales
plt.figure(figsize=(12, 6))
plt.plot(y_test, label='Valores reales', marker='o')
```

```
from sklearn.tree import plot_tree

def crear_ventanas(serie, tamaño_ventana):
  X, y = [], []
  for i in range(len(serie) - tamaño_ventana):
    ventana = serie[i:i + tamaño_ventana]
    etiqueta = serie[i + tamaño_ventana]
    X.append(ventana)
    y.append(etiqueta)
  return np.array(X), np.array(y)

serie_temporal = np.array([10, 12, 15, 18, 20, 22, 25, 28, 30, 32])
tamaño_ventana = 3
X, y = crear_ventanas(serie_temporal, tamaño_ventana)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

modelo = DecisionTreeRegressor()
modelo.fit(X_train, y_train)
predicciones = modelo.predict(X_test)
mse = mean_squared_error(y_test, predicciones)
print(f"Error cuadrático medio: {mse}")

# Visualizar las predicciones vs. valores reales
plt.figure(figsize=(12, 6))
plt.plot(y_test, label='Valores reales', marker='o')
plt.plot(predicciones, label='Predicciones', marker='x')
plt.title('Predicciones vs. Valores reales')
plt.xlabel('Tiempo')
plt.ylabel('Valor')
plt.legend()
plt.grid(True)
plt.show()
```

```
# Visualizar el árbol de decisión
plt.figure(figsize=(20, 10))
plot_tree(modelo, filled=True)
plt.title('Árbol de Decisión')
plt.show()
```

```
Error cuadrático medio: 10.0
```



References

Online Resources

[1]. http://mirlab.org/jang/books/dcpr/Data Clustering and Pattern Recognition
[2]. https://codificandobits.com/blog/redes-neuronales-recurrentes-explicacion-detallada/
[3]. "Deep Learning" de Ian Goodfellow, Yoshua Bengio y Aaron Courville
[4]. "Pattern Recognition and Machine Learning" de Christopher M. Bishop
[5]. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" de Aurélien Géron
[6]. "Grokking Deep Learning" de Andrew Trask
[7]. Deep Learning: A Practitioners Approach, de Josh Patterson y Adam Gibson
[8]. "Data Mining: Practical Machine Learning Tools and Techniques" de Ian H. Witten, Eibe Frank y Mark A. Hall

[9]. International Journal of Latest Research in Engineering and Technology (IJLRET) ISSN: 2454-5031 www.ijlret.com || Volume 08 - Issue 08 || August 2022 || PP. 01-20 Multivariate and Mixed Gaussian Analysis of the Maxima of Ozone, Temperatures, Nitrogen Dioxides and Total Suspended Particulate from 2019-2022 in México City http://www.ijlret.com/Papers/Vol-08-issue-08/1.B2022274.pdf

[10]. International Journal of Latest Research in Engineering and Technology (IJLRET) ISSN: 2454-5031 www.ijlret.com || Volume 10 - Issue 04 || April 2024 || PP. 01-15 Gaussian Mixture of Several Components (Machine Learning) of Daily Ozone and Temperature Maximums in Mexico City Trend 2010-2023 http://www.ijlret.com/Papers/Vol-10-issue-04/1.C2024362.pdf

[11]. International Journal of Latest Research in Engineering and Technology (IJLRET) ISSN: 2454-5031 www.ijlret.com || Volume 07 - Issue 09 || September 2021 || PP. 15-30 http://www.ijlret.com/Papers/Vol-07-issue-09/3.B2021187.pdf Fundamentals for Obtaining New Probability Distribution Functions with Bayesian Inference with Gaussian and NearGaussian Behavior

[12]. International Journal of Latest Research in Engineering and Technology (IJLRET) ISSN: 2454-5031 www.ijlret.com || Volume 04 - Issue 11 || November 2018 || PP. 15-35 A Methodology for Obtaining news Probability Distributions Functions Normal and Extreme Value for Bayesian Inference and Stochastic Mixed Gaussian http://www.ijlret.com/Papers/Vol-04-issue-11/3.B2018131.pdf