www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 16(3), May-Aug 2025, 25-35. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2025.v16i3.1114

Prediction of PM10, SO2, NO2, O3, and CO Concentrations in Guadalajara Using ARIMA and Open Data with Python

Julio Cesar Salas López, Juvencio Sebastián Zarazúa Silva, Jorge A. Ruiz-Vanoye, Eric Simancas-Acevedo
Julio C. Salgado-Ramírez, Ocotlán Díaz-Parra
Universidad Politécnica de Pachuca, México.
E-mail: juve@micorreo.upp.edu.mx

Abstract. Air quality in Guadalajara has deteriorated in recent years, becoming a serious health concern for the local population. In response, this project seeks to mitigate the impact of pollution by developing a prediction platform based on ARIMA models implemented in Python. The system will analyse historical pollutant levels—including $PM_{2.5}$, PM_{10}, SO_2, NO_2, O_3 and CO—enabling the anticipation of high-pollution episodes. Armed with this information, both citizens and authorities will be able to take timely preventative measures. Given the growing interest in air quality and its implications for health, this tool will furnish valuable data for informed decision-making. Moreover, it will facilitate trend analysis and permit short-term forecasts, helping to identify potential pollution episodes before they occur.
Keywords: Air quality, Environment, ARIMA, Predictive model, Public health, Pollution, Guadalajara.

Article Info
Received April 26, 2025
Accepted Jul 1, 2025

1 Introduction

In recent decades, rapid urbanisation has led to a substantial rise in pollutant emissions, adversely affecting both public health and the environment. Guadalajara, one of Mexico's largest cities, is no exception. Its fast-paced urban and industrial expansion has driven increases in pollutants such as ozone (O_3), nitrogen dioxide (NO_2) and particulate matter ($PM_{2.5}$ and PM_{10}). Despite improvements in air-quality monitoring and regulation, a critical gap remains: the absence of predictive tools to forecast pollution episodes.

Decisions are currently informed by historical data and real-time measurements, which limits the efficacy of mitigation strategies. Against this backdrop, the development of a technology-driven platform based on predictive modelling offers a viable solution for enhancing air-quality management. AutoRegressive Integrated Moving Average (ARIMA) models have demonstrated effectiveness in forecasting various phenomena, including environmental conditions. By analysing historical records alongside real-time measurements, ARIMA can produce accurate forecasts for key pollutants in Guadalajara, furnishing invaluable insights for both authorities and citizens.

This study proposes the design and implementation of a predictive platform that integrates ARIMA models with data-science techniques to forecast air quality across the city. The platform aims to provide accessible, accurate information on atmospheric pollution, enabling stakeholders to take timely preventive measures to reduce exposure and its associated health impacts. Such a tool would bolster evidence-based decision-making.

The paper is structured as follows:

- **Section I** reviews the state of the art in air-quality prediction and the application of ARIMA models.
- **Section II** defines the primary research problem.
- **Section III** outlines the platform's development objectives.
- **Section IV** describes the methodology, including data selection, pre-processing and ARIMA configuration.
- **Section V** presents the results, evaluating their accuracy and reliability.
- **Section VI** summarises the conclusions and discusses potential avenues for future research.

2 State of the art

The use of ARIMA models in air-quality prediction is well documented. For instance, a study in Abu Dhabi employed ARIMA to forecast levels of nitrogen dioxide (NO_2) and particulate matter (PM_{10} and $PM_{2.5}$) using data from 2015 to 2023. The results indicated a marked decrease in NO_2 after 2020, alongside an increase in particulate concentrations in 2022, demonstrating the model's capacity to capture air-quality trends (Ramadan, 2024). Similarly, in Chennai, India, univariate ARIMA models were developed to predict daily mean pollutant levels, illustrating the method's versatility across diverse geographical contexts (Nadeem et al., 2020).

The ARIMA methodology follows the Box–Jenkins procedure, comprising model identification, parameter estimation and diagnostic verification. This approach has been applied not only to air quality but also to other domains such as water quality and rainfall forecasting (Hernández et al., 2017; "Short-Term and Long-Term Rainfall Forecasting Using ARIMA Model", 2023). For example, a wetland water-quality forecast based on UV–Vis spectrometry demonstrated ARIMA's effectiveness for short-term predictions (Hernández et al., 2017). Moreover, combining ARIMA with other techniques—such as artificial neural networks (ANNs)—has enhanced forecast accuracy: a Tunisian study found that a hybrid ARIMA–ANN model delivers a more efficient early-warning system for urban air quality (Ayari et al., 2012), suggesting that method integration is a sound strategy for tackling complex datasets.

Hybrid strategies pairing ANNs with statistical models like ARIMA have been proposed to further improve prediction accuracy. López et al. (2016) noted that many researchers combine ANNs and ARIMA, since pure ARIMA may not fully capture data intricacies. This synergy leverages the strengths of both approaches to boost forecasting performance.
Wireless sensor systems represent a prominent approach in air-quality monitoring. For example, the Smart Air Quality Monitoring System (SAQMS) developed by Oyo-Ita measures $PM_{2.5}$, PM_{10}, volatile organic compounds (VOCs) and toxic gases via wireless sensor nodes, enabling real-time data collection and prompt environmental responses (Oyo-Ita, 2023). Likewise, wireless sensor networks (WSNs) have proven effective in delivering continuous air-quality data, facilitating trend forecasting and pollution-control measures (Chang et al., 2016).

The integration of machine-learning algorithms into monitoring platforms has also advanced prediction precision. Akbaba (2023), for instance, developed an air-quality detection system using a feedback neural network to link sensors with a control board, permitting deeper data analysis. This hardware–software fusion not only improves pollutant detection but also optimises emergency responses to air-quality incidents.

Simulation software remains indispensable for studying airborne pollutant dispersion. Zhang and Ryu (2021) employed Airpak to model indoor airflow and moisture distribution, essential for designing effective ventilation and purification systems.
Finally, studies in South America highlight ANNs' utility in air-quality forecasting. Baena-Salazar et al. (2019) used ANN models to predict critical $PM_{2.5}$ events in Colombia's Aburrá Valley, capturing temporal and spatial pollution patterns. Similarly, Quincho and Dionicio (2022) developed ANN architectures to forecast PM_{10} concentrations in Lima using pollutant and meteorological data, demonstrating ANNs' adaptability to varied environmental conditions.

3 Aportation

Figure 1 illustrates the UML diagram of the Air Quality Prediction System, identifying its principal components and their interconnections. On the left-hand side, the user interacts with the web interface, which is designed for querying and presenting forecasts. This interface issues HTTP requests to the Flask backend, acting as the central controller, organising business logic and coordinating communication with the remaining modules.

In the persistence layer, the Flask server connects to the MySQL database, where historical pollutant measurements, user configurations and model outputs are stored. Concurrently, the backend periodically queries an external environmental data API to obtain real-time readings, which are then integrated into the system to generate up-to-date predictions. This modular architecture guarantees scalability, maintainability and rapid response to the varied requirements of both citizens and regulatory authorities.

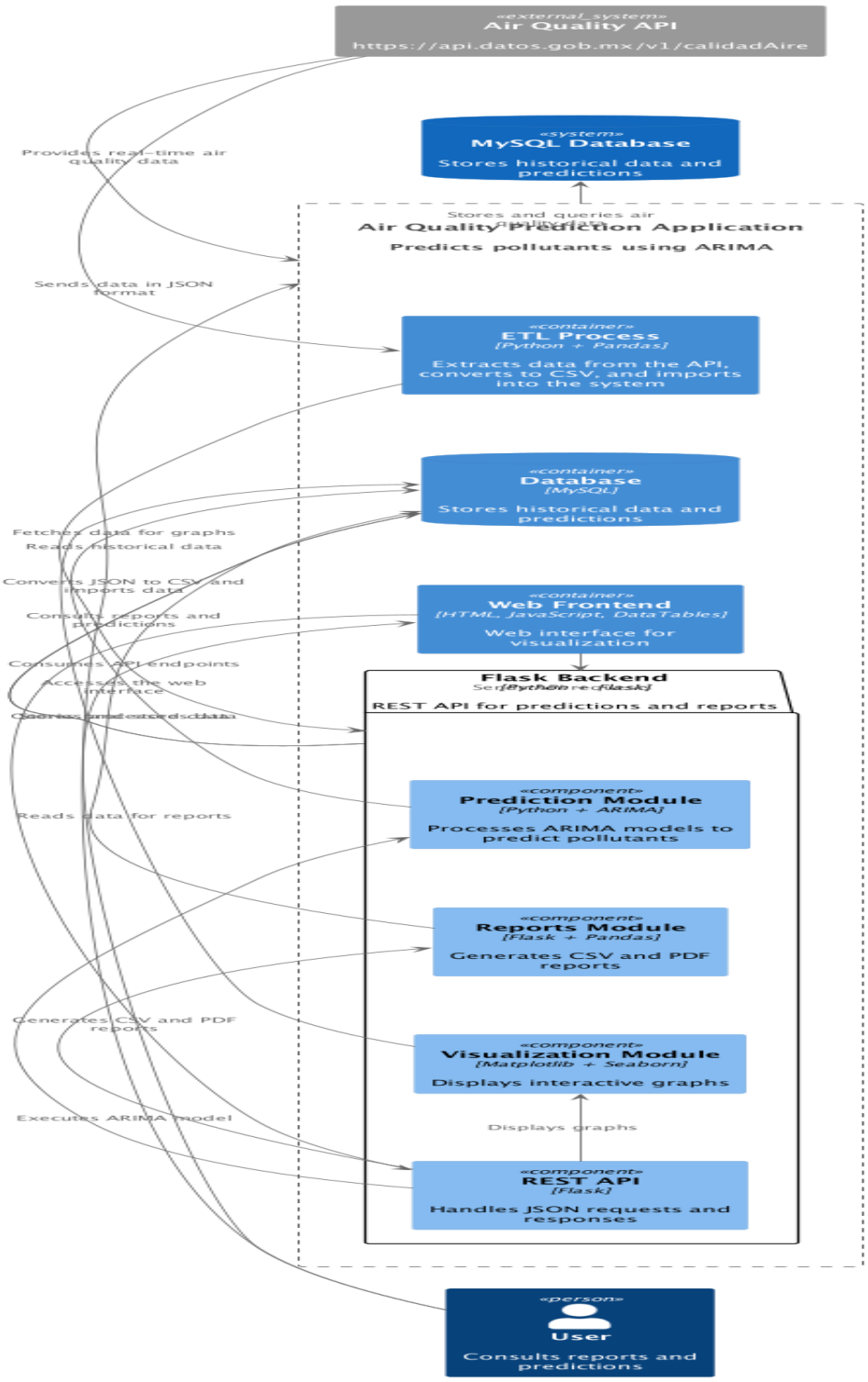


Figure 1. UML Diagram of the Air Quality Prediction System.
It represents the main components of the system, including the interaction between the user, the web interface, the Flask backend, the MySQL database, and the external environmental data API.

4 Experimentation

Level 1: General Context

The system is designed to forecast air pollution levels in various geographical regions using historical data and ARIMA predictive models.

Data Source

Data are retrieved from the Mexican Government's public API: https://api.datos.gob.mx/v1/calidadAire

This API supplies up-to-date information on air quality across multiple locations nationwide.

Pollutants Considered
Only the following pollutants and measurement units are processed:

- **CO (carbon monoxide)** → µg/m³
- **NO_2 (nitrogen dioxide)** → µg/m³
- **O_3 (ozone)** → µg/m³
- **PM_{10} (particulate matter < 10 µm)** → µg/m³
- **SO_2 (sulphur dioxide)** → µg/m³

Any additional pollutants reported by the API are disregarded.

System Actors

- **User**: Views reports and visualisations via a web browser.
- **External API (datos.gob.mx)**: Supplies raw air-quality data.

Storage
Fetched data are converted to CSV format and stored in a MySQL database, which also holds the generated forecasts.

Level 2: Containers

Flask Backend (REST API)
Implemented in Python using Flask, responsible for:

- Data upload
- Prediction generation
- Report creation
- Result visualization

Web Fronted
Built with HTML and JavaScript (DataTables), offering:

- Filters by state, municipality and pollutant
- Interactive graph visualisations
- Downloadable reports in PDF and CSV formats

ETL Process
Developed in Python with Pandas, it:

1. Extracts JSON data from the API
2. Filters for the five specified pollutants
3. Converts data to CSV format
4. Loads the CSV into the database for analysis

MySQL Database

Contains two principal tables:

- **informacion**: Daily average values by pollutant, state and municipality
- **predicciones**: ARIMA outputs with parameters (p, d, q), forecast values and confidence intervals

Level 3: Backend Components

REST API

Handles front-end requests and returns structured JSON responses.

Prediction Module

- Executes an ARIMA (2, 1, 2) model on the latest 30 days of daily averages
- Generates forecasts solely for the five defined pollutants
- Pseudocode:

```
for pollutant in [CO, NO₂, O₃, PM₁₀, SO₂]:
    data ← fetch_last_30_days(pollutant)
    model ← ARIMA(data, order=(2,1,2))
    forecast, conf_int ← model.forecast(steps, alpha=0.05)
    store_results(pollutant, forecast, conf_int)
```

This modular architecture ensures clarity, maintainability and scalability for forecasting air quality across Guadalajara and beyond. More specific pseudocode:

```
Procedure ExecutePrediction(prediction_date):

  1. Validate input
     If prediction_date is missing Then
       Exit with message "Date not provided"

  2. Open connection to storage

  3. Check for existing predictions
     If any prediction for prediction_date exists Then
       Close connection
       Exit with message "Prediction already exists"

  4. Fetch pollutant-region combinations with data in the 30 days b
prediction_date
     If none found Then
       Close connection
       Exit with message "No data available for this date"

  5. Initialize empty list "results"

  6. For each combination in fetched list Do
       a. Retrieve the last 30 daily values for that pollutant
region
       b. If fewer than 30 values Then
            Skip to next combination
       c. Compute first differences of the series
       d. Fit ARIMA(2,1,2) to the differenced series
       e. Forecast the next value and compute a 95 % confidence inte
       f. Round forecast and interval bounds to three decimal places
```

```
      g. Create a prediction record with:
         - pollutant, region, prediction_date
         - forecasted value and units
         - ARIMA order (2,1,2)
         - confidence interval
      h. Save the record to storage
      i. Add the record to "results"

  7. Commit all changes

  8. Close connection

  9. Return summary containing:
     - number of predictions made
     - list of prediction records

End Procedure
```

Reporting Module

- Generates CSV and PDF files containing:
  - Prediction date
  - Pollutant
  - Actual average value
  - Predicted value
  - Percentage change
  - 95 % confidence interval

Visualisation Module

- Creates plots using Matplotlib and Seaborn:
  - Comparative line chart (actual vs. predicted values)
  - Error histogram
  - Scatter plot

General System Flow

1. Data are extracted from the datos.gob.mx API.
2. The ETL process filters for CO, NO_2, O_3, PM_{10} and SO_2, then converts the data to CSV.
3. The CSV files are loaded into the MySQL database's informacion table.
4. The system identifies pending dates and runs ARIMA forecasts for each pollutant.
5. Forecasts are stored in the predicciones table.
6. Users view results in their web browser, applying filters by region and pollutant.
7. The system permits export of metrics, reports and graphs in CSV and PDF formats.

5 Results

This section presents the results obtained from implementing the atmospheric pollutant forecasting model in Guadalajara, Jalisco. Five principal pollutants were analysed—CO, NO_2, O_3, PM_{10} and SO_2—using historical data retrieved from the Mexican Government API and processed via an ARIMA model.

Analysis of the prediction process and potential errors

The prediction process comprises several essential steps to assess feasibility and ensure result accuracy. Below, the calculation flow, error conditions and criteria for a valid prediction are outlined.

Prediction calculation process

1. **Parameter input**: The function accepts a specific date for which a forecast is required.
2. **Existing-forecast verification**: The database is queried to determine whether a forecast already exists for the given date.
   – *Potential error*: If a forecast is found, the system issues a warning and prevents duplication.
3. **Retrieval of available pollutants**: All pollutants with records in the informacion table from the 30 days preceding the selected date are retrieved.
   – *Potential error*: If insufficient data are available for this interval, an error is returned, indicating that the forecast cannot be produced.
4. **Extraction of historical data**: For each pollutant–state–municipality combination, measured values for the last 30 days are queried.
   – *Potential error*: If fewer than 30 records are obtained, that pollutant is excluded, as the ARIMA model requires a minimum dataset for valid forecasting.
5. **ARIMA model application**: An ARIMA(2,1,2) model is fitted to the historical series, and a forecast for the specified date is generated.
   – *Potential error*: Failure to converge—due to inconsistent or insufficient data—will result in an unsuccessful forecast for that pollutant.
6. **Database storage**: Successfully generated forecasts are saved in the predicciones table, together with their 95 % confidence intervals.

Possible causes of error and their impact

Cause of Error	Description	Impact
Date not provided	A date parameter is not received in the request	The prediction is not executed
Prediction already exists	A prediction for the given date already exists	Avoids duplication of data
Insufficient data	No records found in the database for the last 30 days	Prediction cannot be generated for that date
Less than 30 records available	Data exists but is insufficient for ARIMA	The pollutant is excluded from the prediction
ARIMA model failure	Unable to fit due to data inconsistencies	Prediction for the affected pollutant is not generated

Necessary conditions for a valid prediction

To generate a prediction without errors, the following conditions must be satisfied:

- The date must be provided correctly.
- No existing prediction should be present for the selected date.
- At least 30 days' historical data must be available in the database.
- Historical data must be properly formatted and free of extreme outliers.
- The ARIMA model must fit the time series adequately.

Comparative graphs (Figure 2 and 3) illustrate the progression of actual versus predicted values for each pollutant over the study period. Overall, the model captures the underlying trend with moderate variability for certain pollutants. However, particular outliers produce significant deviations—especially for PM_{10} and CO—indicating the need for further model refinement.

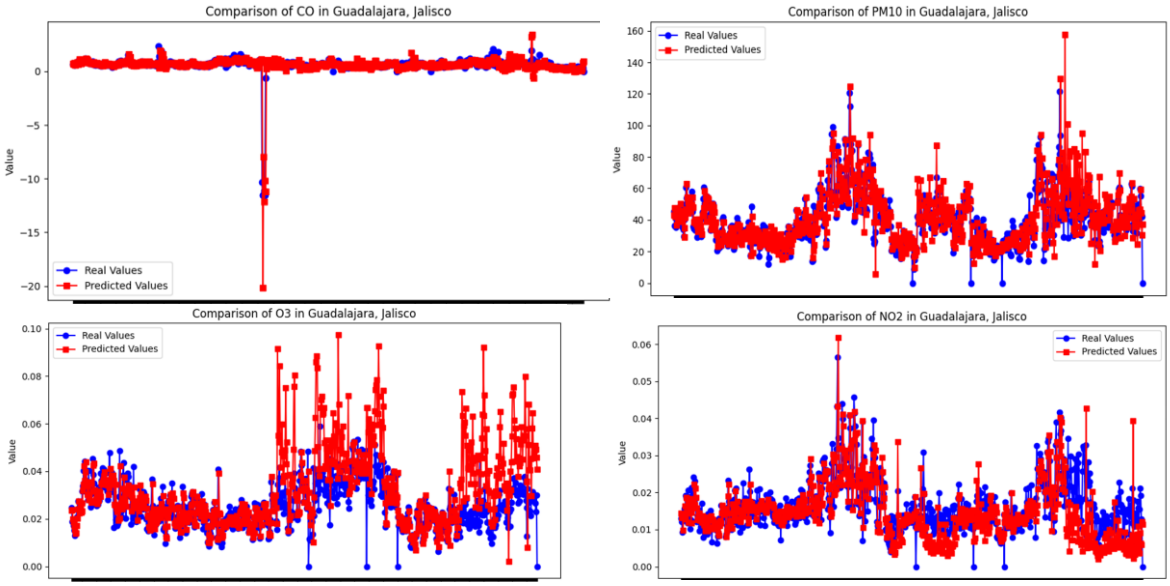


Figure 2. a) Comparison of Actual and Predicted Values for CO. b) Comparison of Actual and Predicted Values for PM10. c) Comparison of Actual and Predicted Values for O_3. d) Comparison of Actual and Predicted Values for SO_2.

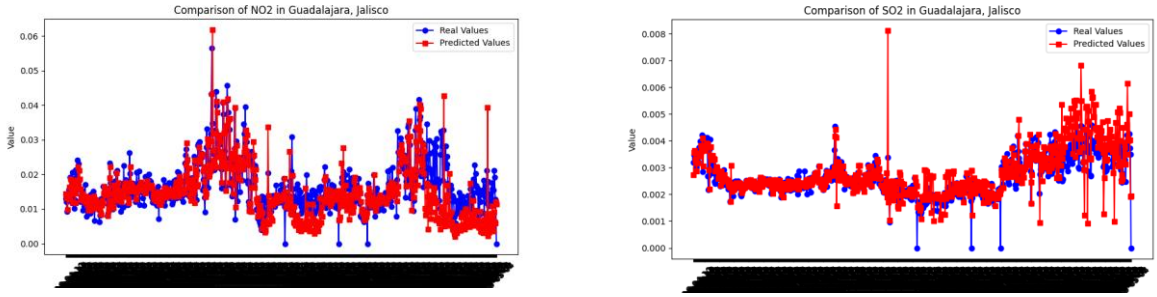


Figure 3. Comparison of Actual and Predicted Values for NO_2

Error distribution

To evaluate the model's performance, scatter plots (Figure 2) and error histograms (Figures 4 and 5) were produced. The scatter plots show that predicted values generally correspond with actual values, albeit with wider dispersion for CO and PM_{10}. The error histograms demonstrate that most errors cluster around zero, although notable deviations are observed for some pollutants.

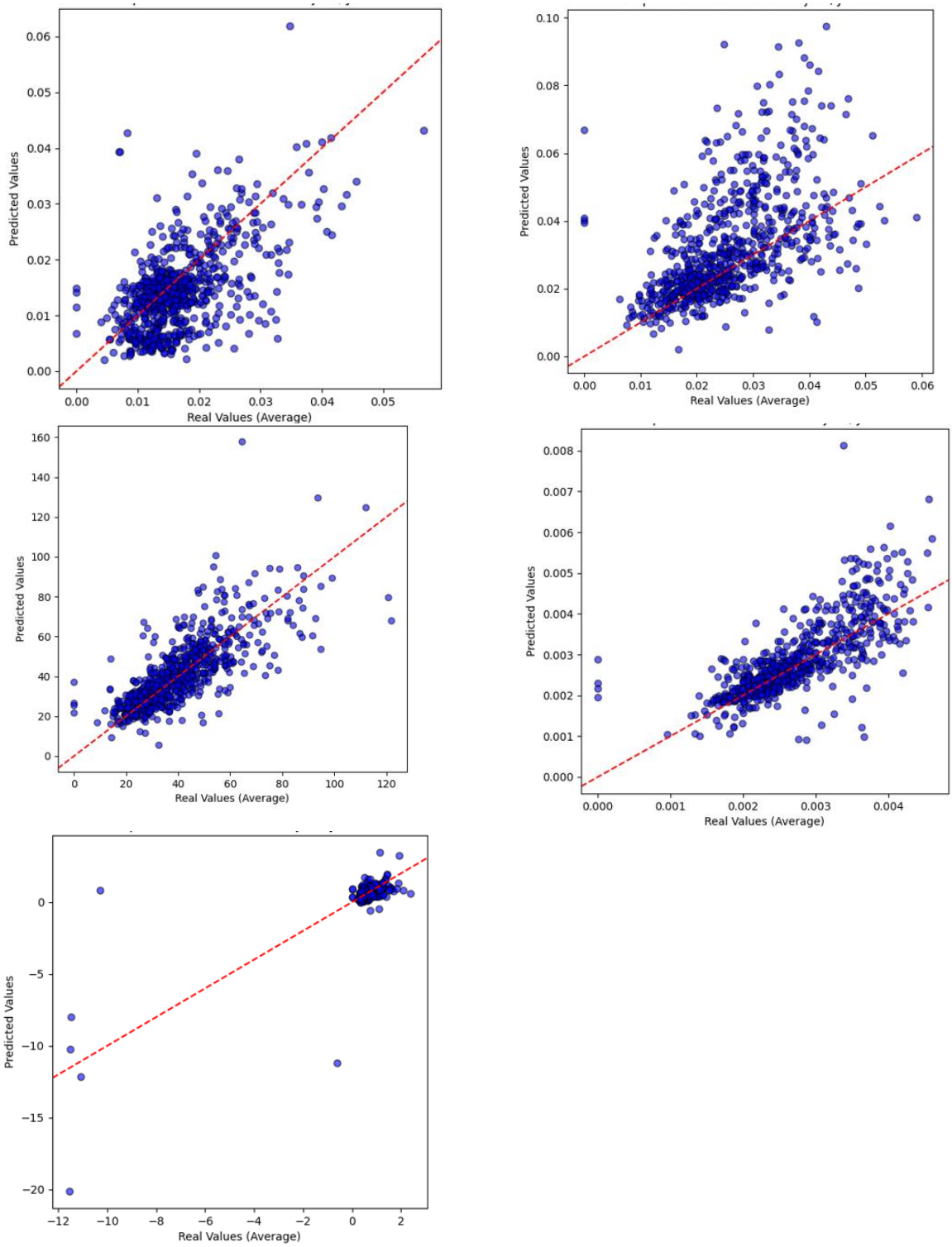
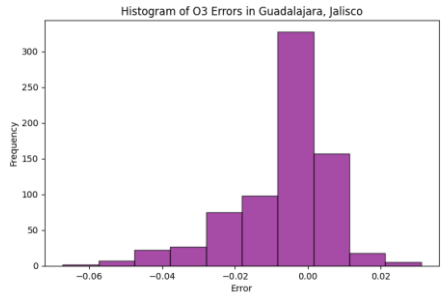
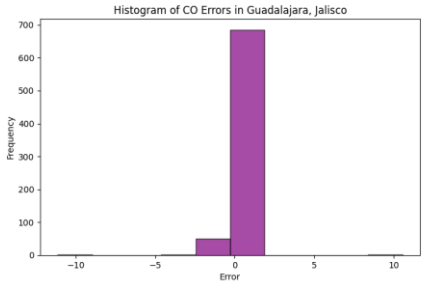
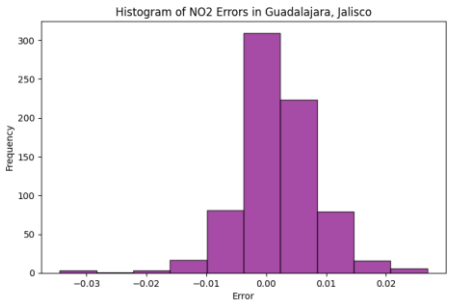
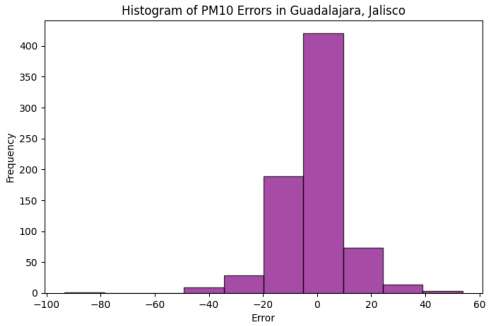


Figure 4. Scatter Plot of Actual vs Predicted Values for:
a) Scatter Plot of Actual vs Predicted Values for NO_2. b) Scatter Plot of Actual vs Predicted Values for O_3.
c) Scatter Plot of Actual vs Predicted Values for PM10. d) Scatter Plot of Actual vs Predicted Values for SO_2.
e) Scatter Plot of Actual vs Predicted Values for CO.









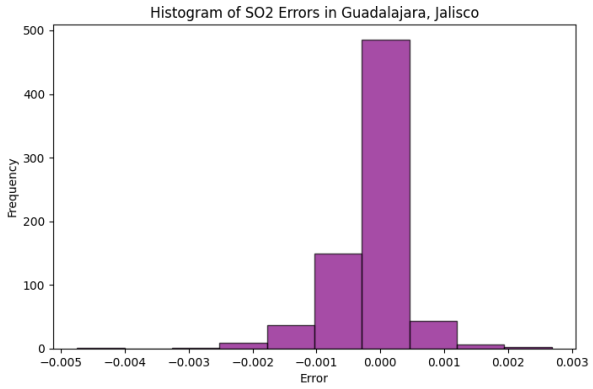


Figure 5. Error Histogram for:
a) Error Histogram for O_3. b) Error Histogram for CO.
c) Error Histogram for NO_2. d) Error Histogram for PM10
e) Error Histogram for SO_2

Error metrics: RMSE and MAE
To quantify the model's accuracy, root mean square error (RMSE) and mean absolute error (MAE) metrics were calculated for each pollutant (Figure 4). The results are:

- CO: RMSE = 0.72, MAE = 0.23
- NO_2: RMSE = 0.01, MAE = 0.01

- O_3: RMSE = 0.01, MAE = 0.01
- PM10: RMSE = 11.66, MAE = 8
- SO_2: RMSE = 0, MAE = 0

6 Conclusions

The results of this study indicate that the ARIMA model performs robustly in forecasting air quality for Guadalajara, Jalisco. Analysis of time series for CO, NO_2, O_3, PM_{10} and SO_2 revealed that the model adapts well to sufficiently stable datasets, demonstrating particularly high accuracy for NO_2 and O_3, where forecasts correlate strongly with observed values. Conversely, predictions for CO and PM_{10} exhibited significant shortcomings—evident in elevated RMSE and MAE metrics and increased variance—likely due to greater volatility and outliers in the historical records. Accordingly, more rigorous data preprocessing and exploration of alternative or hybrid modelling approaches that incorporate non-linear dynamics or additional external variables are recommended.

These findings underscore the necessity of maintaining a robust database with a minimum of 30 daily records per pollutant, state and municipality—the threshold for valid ARIMA implementation. Data insufficiency was a principal factor behind several unfulfilled forecasts, emphasising the importance of ensuring consistent, high-quality data streams to enhance model coverage. Furthermore, this work demonstrates that integrating technologies such as Flask, Pandas, MySQL and standard statistical libraries can yield an effective urban air-quality monitoring and prediction platform. While the ARIMA model proved capable of accurately forecasting the trajectories of NO_2 and O_3, further optimisation—such as parameter tuning or hybridisation—is advisable to improve accuracy for CO and PM_{10}.

References

Ramadan, M. S., Abuelgasim, A., & Al Hosani, N. (2024). Advancing air quality forecasting in Abu Dhabi, UAE, using time series models. *Frontiers in Environmental Science, 12*. https://doi.org/10.3389/fenvs.2024.1393878

Nadeem, I., Ilyas, A. M., & Uduman, P. S. S. (2020). Analyzing and forecasting ambient air quality of Chennai city in India. *Geography, Environment, Sustainability, 13*(3), 13–21. https://doi.org/10.24057/2071-9388-2019-97

Hernández, N., Camargo, J. A., Moreno, F., Torres, A., & Nossa, L. P. (2017). ARIMA as a forecasting tool for water quality time series measured with UV–Vis spectrometers in a constructed wetland. *Tecnología y Ciencias del Agua, 8*(5), 127–139. https://doi.org/10.24850/j-tyca-2017-05-09

Khan, M. M. H., Mustafa, M. R. U., Hossain, M. S., Shams, S., & Julius, A. D. (2023). Short-term and long-term rainfall forecasting using ARIMA model. *International Journal of Environmental Science and Development, 14*(5), 292–298. https://doi.org/10.18178/ijesd.2023.14.5.1447

Oyo-Ita, E. U., Ekah, U. J., Ana, P., & Ewona, I. O. (2023). Development of a smart air quality monitoring system using wireless sensors. *Advances in Research, 24*(6), 50–59. https://doi.org/10.9734/air/2023/v24i6984

Chang, B., Zhang, X., & Xv, B. (2016). Air quality monitoring network localisation algorithm based on RSSI ranging. In *Proceedings of the 7th International Conference on Mechatronics, Control and Materials (ICMCM 2016)*. https://doi.org/10.2991/icmcm-16.2016.68

Akbaba, C. E., & Dişken, G. (2023). Feedforward neural network-based indoor air quality detection system. *International Journal of Applied Methods in Electronics and Computers*. https://doi.org/10.58190/ijamec.2023.64

Jiménez, S. P. B., Ramírez, C. L., Hernández, D. A. G., Rodríguez, V. M. Z., & Araiza, M. Á. C. (2019). Red neuronal artificial para la clasificación y predicción de la calidad del aire. *Programación Matemática y Software, 11*(2). https://doi.org/10.30973/progmat/2019.11.2/7

González, C. J. T., Landassuri-Moreno, V., Hernández, J. J. C., & Albino, J. M. F. (2016). Predicción de oxígeno disuelto en acuacultura semi-intensiva con redes neuronales artificiales. *Research in Computing Science, 120*(1), 159–168. https://doi.org/10.13053/rcs-120-1-14

González, M., Balsategui, S. D. G., & Sarasibar, X. A. (2021). La monitorización de la calidad del aire interior como herramienta de evaluación y mejora de la salubridad de un espacio. *Anales de Edificación, 6*(3), 13. https://doi.org/10.20868/ade.2020.4610

Baena-Salazar, D., Jiménez, J., Zapata, C., & Ramírez-Cardona, Á. (2019). Red neuronal artificial aplicado para el pronóstico de eventos críticos de $PM_{2.5}$ en el Valle de Aburrá. *Dyna, 86*(209), 347–356. https://doi.org/10.15446/dyna.v86n209.63228

Lino-Ramírez, C., Bautista-Sánchez, R., & Bombela-Jiménez, S. (2019). Utilización de un sistema en tiempo real para la predicción de contaminación del aire. *Research in Computing Science, 148*(7), 441–453. https://doi.org/10.13053/rcs-148-7-33