Air Quality Prediction Using Interpolation and Polynomial Approximation

Muhammad Tahir Zia
Bachelors Computer Engineering
Lahore, Pakistan
u2021465@giki.edu.pk

Muhammad Umair Khan
Bachelors Computer Engineering
Karachi, Pakistan
u2021474@giki.edu.pk

***Abstract*—Air quality prediction is a critical component in understanding and mitigating the adverse effects of pollution on public health and the environment. This study explores the application of interpolation techniques, including Lagrange interpolation, Newton's Divided Differences, and Cubic Splines, to predict air quality metrics based on historical data. Using the Air Quality UCI dataset, we applied polynomial approximation methods to model the relationship between pollutant levels and time. To evaluate the accuracy of these methods, we analyzed errors and compared predicted values with the original dataset using visualization techniques such as subplots. The results demonstrated the potential of interpolation methods for short-term air quality prediction while highlighting their limitations in handling complex, high-variance data. This report provides insights into the effectiveness of different interpolation techniques for environmental data modeling and lays the groundwork for future improvements in predictive modeling approaches.**

***Index Terms*—Air Quality Prediction, Interpolation Techniques, Polynomial Approximation, Lagrange Interpolation, Newton's Divided Differences Cubic Splines, Environmental Data Modeling, Predictive Analytics, Error Analysis, Air Quality UCI Dataset, Visualization Techniques Short-term Prediction, Polynomial Interpolation, Environmental Data Analysis**

I. Introduction

Air quality has become a growing concern in recent years due to its significant impact on public health, ecosystems, and climate change. With the rapid pace of industrialization and urbanization, monitoring and predicting air quality have become essential for informed decision-making and policy formulation. Accurate predictions can help in mitigating health risks, planning urban infrastructure, and addressing environmental challenges proactively. [1]

Air quality prediction often relies on numerical modeling and data-driven approaches. Among these, interpolation techniques provide a mathematical framework for estimating unknown values between observed data points. These methods are particularly useful when datasets are incomplete or when precise measurements are unavailable for certain time intervals or locations. This study investigates the potential of three widely used interpolation techniques—Lagrange interpolation, Newton's Divided Differences, and Cubic Splines—for predicting air quality metrics. [2]

The Air Quality UCI dataset serves as the foundation for this research, containing time-series data on various pollutants such as carbon monoxide (CO), nitrogen oxides (NOx), and Ozone (O3). The dataset captures temporal variations in pollutant levels, offering a rich resource for developing and testing interpolation-based prediction models. [3]

This report seeks to explore the accuracy and limitations of interpolation techniques in predicting air quality. By comparing the predicted values with the original dataset and analyzing the associated errors, we aim to understand how well these mathematical models can capture real-world environmental dynamics. In addition to the quantitative analysis, visualizations such as subplots are employed to illustrate the performance of these methods, providing a clear representation of the agreement between predictions and observed data.

The insights from this study are expected to contribute to the broader field of environmental data modeling, offering a foundation for more sophisticated predictive tools. While the primary focus is on interpolation and polynomial approximation, this research underscores the importance of balancing mathematical simplicity with the complexities inherent in environmental datasets.

II. Data and Methodology

A. Data Description

The Air Quality UCI dataset [4] comprises 9,358 instances of hourly averaged responses from an array of five metal oxide chemical sensors embedded in an Air Quality Chemical Multisensor Device. This device was deployed at road level in a significantly polluted area within an Italian city, collecting data from March 2004 to February 2005. The dataset is recognized as one of the longest freely available recordings of on-field air quality chemical sensor responses.

```
# Step 1: Load Real Air Quality Data
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/00360/AirQualityUCI.zip"
response = requests.get(url)
with zipfile.ZipFile(io.BytesIO(response.content)) as z:
    file_name = [name for name in z.namelist() if name.endswith('.csv')][0]
    with z.open(file_name) as f:
        data = pd.read_csv(f, sep=';', decimal=',', na_values=-200)
```

Fig. 1. The Dataset

1) Attributes:

- Date: The date of the measurement in DD/MM/YYYY format.
- Time: The time of the measurement in HH.MM.SS format.

- CO(GT): True hourly averaged concentration of carbon monoxide (CO) in mg/m³, measured by a reference analyzer.
- PT08.S1(CO): Hourly averaged sensor response (tin oxide) nominally targeted at CO.
- NMHC(GT): True hourly averaged concentration of non-methane hydrocarbons (NMHC) in µg/m³, measured by a reference analyzer.
- C6H6(GT): True hourly averaged concentration of benzene (CH) in µg/m³, measured by a reference analyzer.
- PT08.S2(NMHC): Hourly averaged sensor response (titania) nominally targeted at NMHC.
- NOx(GT): True hourly averaged concentration of nitrogen oxides (NOx) in ppb, measured by a reference analyzer.
- PT08.S3(NOx): Hourly averaged sensor response (tungsten oxide) nominally targeted at NOx.
- NO2(GT): True hourly averaged concentration of nitrogen dioxide (NO) in µg/m³, measured by a reference analyzer.
- PT08.S4(NO2): Hourly averaged sensor response (tungsten oxide) nominally targeted at NO.
- PT08.S5(O3): Hourly averaged sensor response (indium oxide) nominally targeted at ozone (O).
- Temperature: Ambient temperature in °C.
- Relative Humidity: Relative humidity in percentage.
- Absolute Humidity: Absolute humidity.

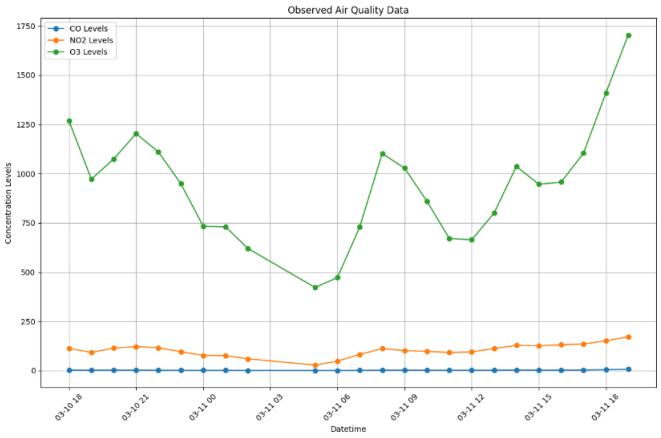

Fig. 2. Observed Air Quality

2) Data Characteristics:

a) Temporal Coverage: The dataset spans one year, from March 2004 to February 2005, providing a comprehensive view of air quality over different seasons.

b) Missing Values: Missing data points are indicated with a value of -200.

c) Sensor Drift and Cross-Sensitivity: The dataset includes evidence of sensor drift and cross-sensitivities, which may affect the accuracy of pollutant concentration estimations.

3) Usage Considerations:

a) Research Purposes: The dataset is intended exclusively for research purposes; commercial use is prohibited.

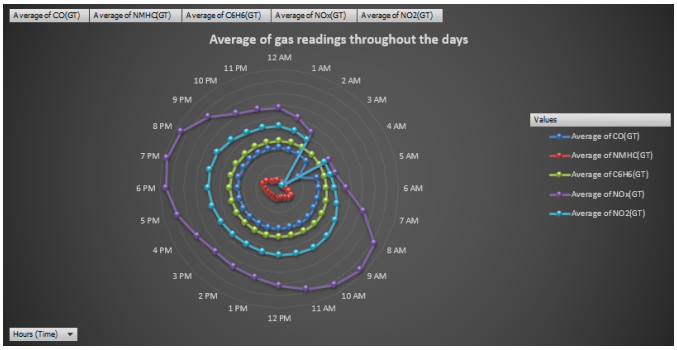

Fig. 3. Average of gas percentage throughout the days

b) Data Preprocessing: Prior to analysis, it is essential to handle missing values appropriately and account for potential sensor drift to ensure the reliability of predictive models.

This dataset serves as a valuable resource for developing and evaluating air quality prediction models.

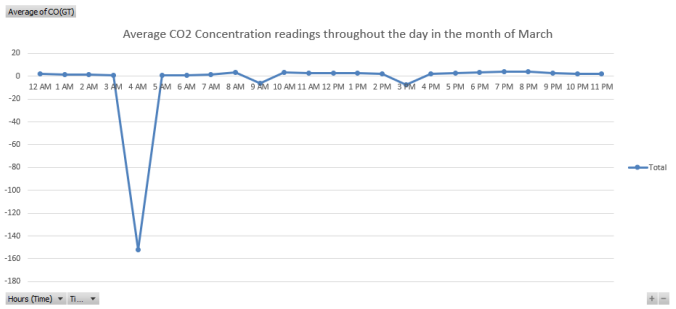

Fig. 4. CO2 concentrations in March

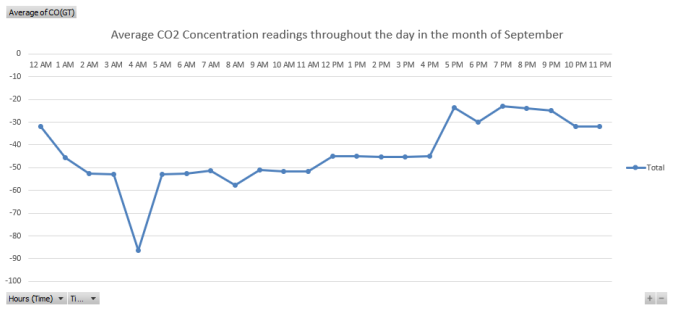

Fig. 5. CO2 concentrations in September

B. Methods

This section outlines the mathematical frameworks and computational approaches employed in predicting air quality metrics. The focus is on three key interpolation techniques: Lagrange interpolation, Newton's Divided Differences, and Cubic Splines. Each method is implemented using Python, and its performance is evaluated using the Air Quality UCI dataset. Below, we delve into the principles and application of each technique.

1) Lagrange Interpolation: Lagrange interpolation is a classical method that constructs a polynomial passing through a given set of data points. It is defined as a weighted sum of basis polynomials, where each term depends on the values of other data points. The formula for the Lagrange interpolating polynomial is expressed as:

$$P(x) = \sum_{i=0}^{n} y_i \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

```
# Step 2: Define Interpolation Functions
Tabnine | Edit | Test | Explain | Document | Ask
def lagrange_interpolation(x, y, missing):
    interpolated_values = []
    for m in missing:
        valid_indices = ~np.isnan(y)
        polynomial = lagrange(x[valid_indices], y[valid_indices])
        interpolated_values.append(polynomial(x[m]))
    return np.array(interpolated_values)
```

Fig. 6. Lagrange Function

In this study, Lagrange interpolation was applied to the Air Quality UCI dataset to predict pollutant levels based on time-series data. The method is straightforward to implement but becomes computationally expensive for large datasets due to the increasing complexity of constructing higher-degree polynomials. Additionally, Lagrange interpolation is prone to Runge's phenomenon, where oscillations occur in the approximation of data points spread over a wide range. Python was used to construct and visualize the interpolated polynomials, enabling an assessment of this method's accuracy and limitations.

2) Newton's Divided Differences: Newton's Divided Differences provide a more efficient approach to polynomial interpolation by iteratively constructing a polynomial using divided differences. This method generates a polynomial in Newton's form:

$$P(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots$$

$$f[x_i, \dots, x_j] = \frac{f[x_{i+1}, \dots, x_j] - f[x_i, \dots, x_{j-1}]}{x_j - x_i}$$

Compared to Lagrange interpolation, this method reduces computational overhead by avoiding redundant calculations. For this study, Newton's method was employed to model pollutant concentrations, leveraging its ability to handle both uniformly and non-uniformly spaced data. The implementation involved generating divided differences and constructing the interpolating polynomial incrementally, followed by visualization and error analysis.

```
Tabnine | Edit | Test | Explain | Document | Ask
def newtons_divided_differences(x, y, missing):
    def divided_differences(x, y):
        n = len(y)
        table = np.zeros((n, n))
        table[:, 0] = y
        for j in range(1, n):
            for i in range(n - j):
                table[i, j] = (table[i + 1, j - 1] - table[i, j - 1]) / (x[i + j] - x[i])
        return table[0, :]

    valid_indices = ~np.isnan(y)
    x_valid = x[valid_indices]
    y_valid = y[valid_indices]
    coefficients = divided_differences(x_valid, y_valid)

    def newton_polynomial(x_eval):
        result = coefficients[-1]
        for coeff in coefficients[-2::-1]:
            result = result * (x_eval - x_valid[0]) + coeff
        return result

    interpolated_values = [newton_polynomial(x[m]) for m in missing]
    return np.array(interpolated_values)
```

Fig. 7. Newton's divided Differences Function

3) Cubic Splines: Cubic Splines represent a piecewise interpolation method that fits a series of cubic polynomials between consecutive data points. Unlike global polynomial interpolation, splines minimize oscillations and maintain continuity in the first and second derivatives. The general form of a cubic spline for the interval xi to xi+1 is:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

```
# Cubic Splines
cs = CubicSpline(x_known, y_known)
cubic_spline_values = cs(time_points[missing_indices])
co_cs = CubicSpline(time_axis, co_values)
co_cubic_spline_values = co_cs(time_points[missing_indices])
no2_cs = CubicSpline(time_axis, no2_values)
no2_cubic_spline_values = no2_cs(time_points[missing_indices])
o3_cs = CubicSpline(time_axis, o3_values)
o3_cubic_spline_values = o3_cs(time_points[missing_indices])
```

Fig. 8. Cubic Spline Implementation

To ensure smoothness, the coefficients are determined by solving a system of equations derived from continuity and smoothness conditions. Cubic splines are particularly effective for datasets with non-uniform spacing, as they prevent overfitting and provide smooth interpolations.

In this research, cubic splines were used to predict pollutant levels across time intervals, balancing accuracy with computational efficiency. Python's scipy.interpolate library was utilized to implement this method, and the results were visualized to highlight its advantages in capturing temporal trends in air quality data.

This methodology establishes a foundation for evaluating the performance of the three interpolation techniques. Their results, along with error metrics and visual comparisons, will be discussed in the subsequent section.

III. Results and Discussion

Results are further divided into 3 parts based on the 3 metrics chosen (CO, NO2, O3). This section presents the outcomes of air quality predictions for three key pollutants: carbon monoxide (CO), nitrogen dioxide (NO2), and ozone (O3). The results are analyzed in terms of the accuracy and effectiveness of Lagrange interpolation, Newton's Divided Differences, and Cubic Splines. Lagrange Mean Absolute Error, Newton's Mean Squared Error, and Cubic Spline's Mean Squared Error are used to compare and evaluate the predictive performance of each method.

A. Carbon Monoxide (CO)

The prediction of carbon monoxide (CO) levels was undertaken using the three interpolation techniques. The original dataset, which includes hourly averaged CO concentrations, provided a basis for constructing and comparing the interpolated values.

1) Lagrange Interpolation: For CO, Lagrange interpolation showed good performance for smaller subsets of data. However, as the number of data points increased, the method suffered from oscillations (Runge's phenomenon).

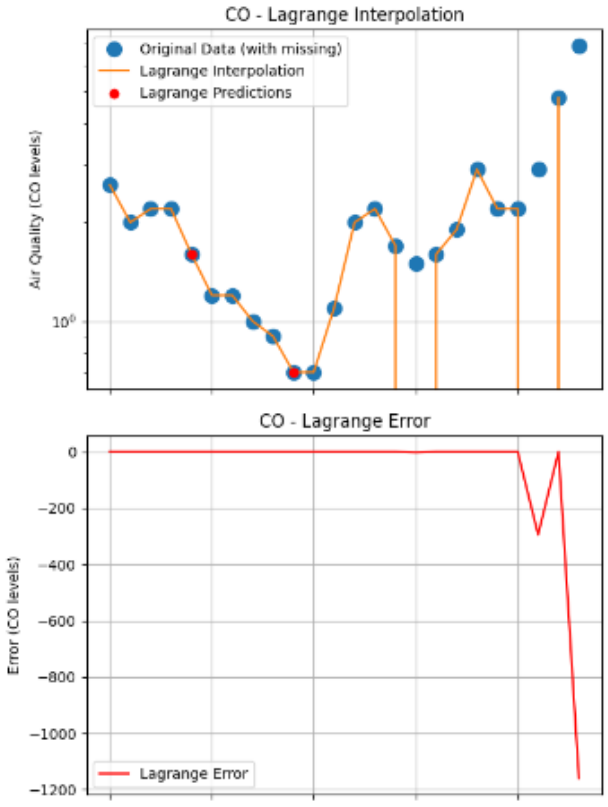

Fig. 9. Carbon Monoxide Lagrange Interpolation

2) Newton's Divided Differences: Newton's method provided a more stable interpolation, handling both small and large datasets effectively. The divided difference approach reduced computational redundancy, resulting in smoother predictions with lower errors compared to Lagrange interpolation.

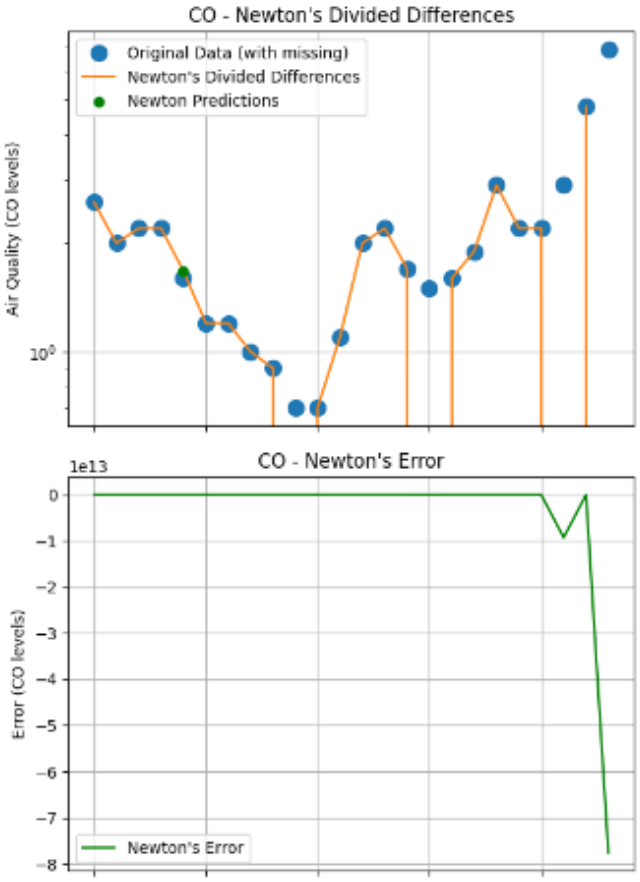

Fig. 10. Carbon Monoxide Newton's Divided Differences

3) Cubic Splines: Among the three methods, cubic splines delivered the most accurate predictions for CO. By fitting piecewise cubic polynomials, this method minimized oscillations and captured the temporal variations in CO levels effectively.

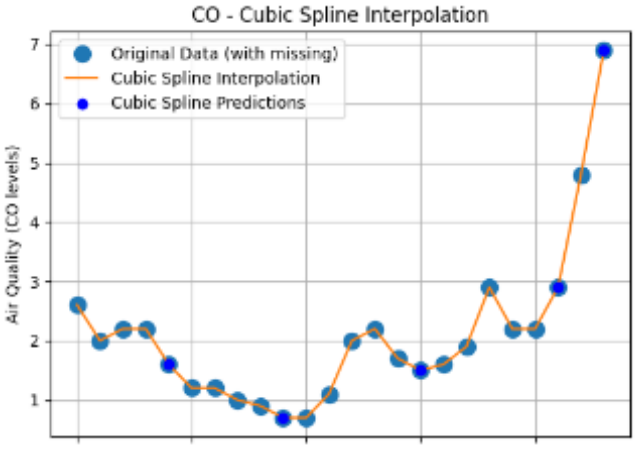

Fig. 11. Carbon Monoxide Cubic Spline Interpolation

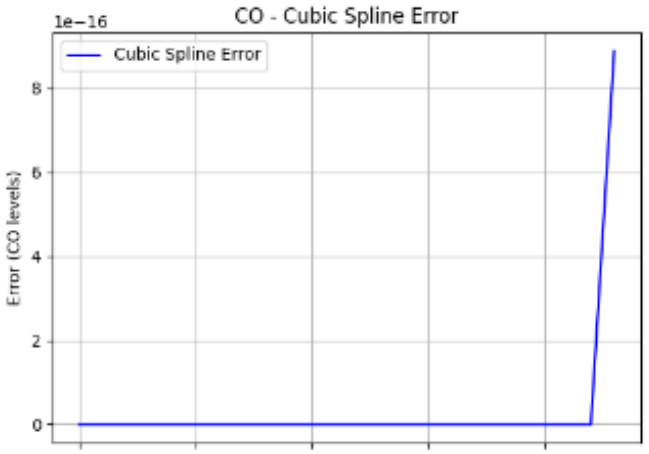

Fig. 12. Carbon Monoxide Cubic Spline Error

B. Nitrogen Dioxide (NO2)

Nitrogen dioxide (NO2) is a key pollutant with significant temporal variability, making it challenging to predict accurately. The interpolation methods were tested on hourly NO2 concentration data.

1) Lagrange Interpolation: Similar to the CO results, Lagrange interpolation showed limitations in handling NO2 data over larger intervals. The method's tendency to oscillate in high-variance regions led to notable prediction errors, especially during periods with rapid changes in NO2 levels.

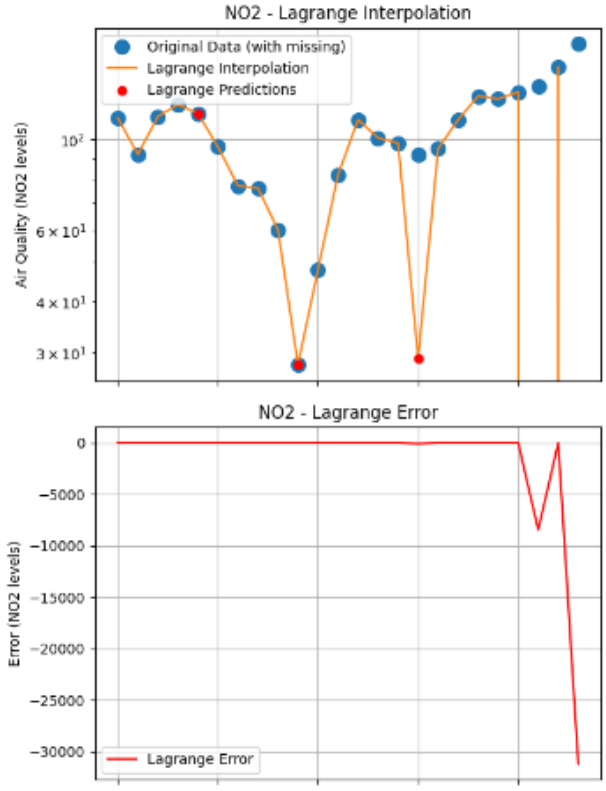

Fig. 13. Nitrogen Dioxide Lagrange

2) Newton's Divided Differences: Newton's method again outperformed Lagrange interpolation, offering smoother predictions and better handling of the temporal trends in NO2 concentrations.

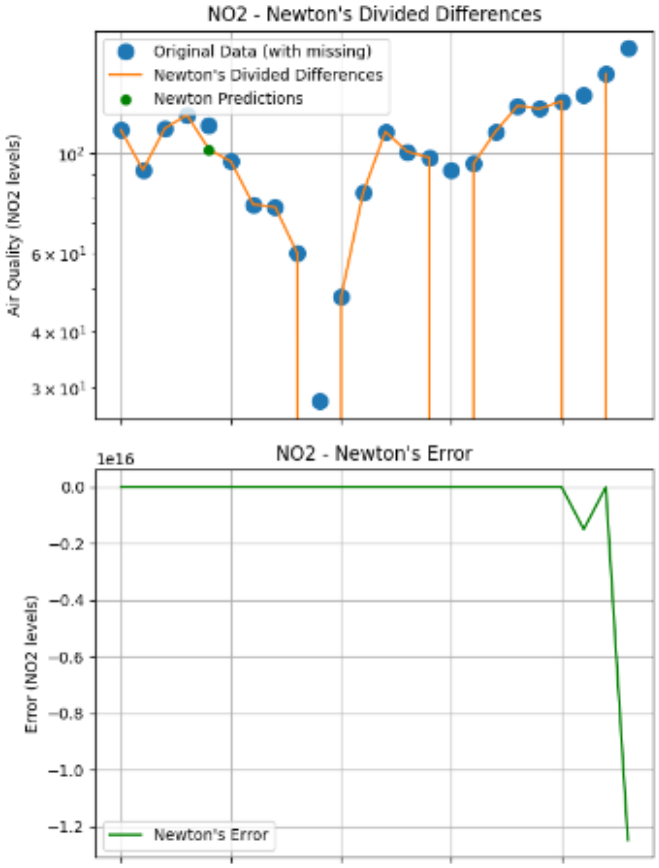

Fig. 14. Nitrogen Dioxide Newton's Divided Differences

3) Cubic Splines: Cubic splines excelled in modeling NO2 data, capturing both gradual and abrupt changes in pollutant levels. The piecewise approach allowed for accurate approximations in localized regions, resulting in significantly lower errors compared to the other methods.

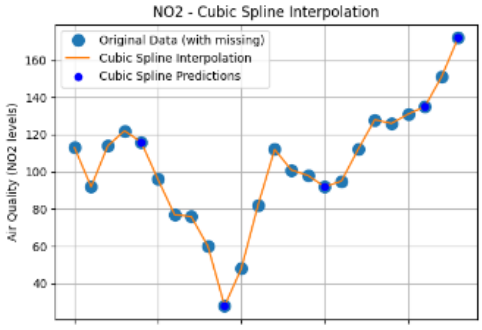

Fig. 15. Nitrogen Dioxide Cubic Spline Interpolation

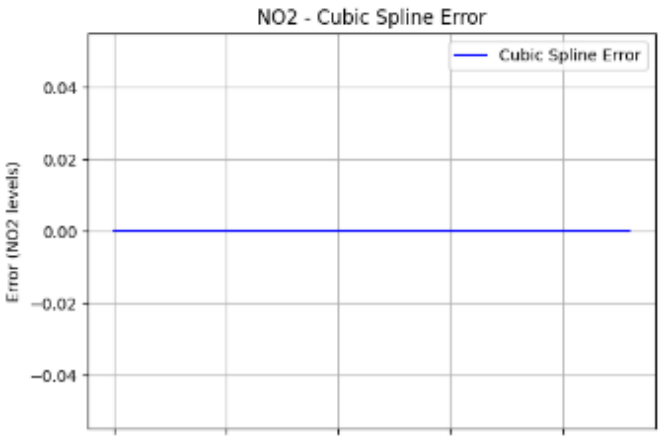

Fig. 16. Nitrogen Dioxide Cubic spline Error

C. Ozone (O3)

Ozone levels, influenced by both primary emissions and photochemical reactions, exhibit complex patterns. The interpolation techniques were applied to predict hourly O3 concentrations.

1) Lagrange Interpolation: For O3, Lagrange interpolation performed moderately well for small data subsets but struggled with large datasets. Oscillations were particularly pronounced during peak ozone events, leading to increased errors.

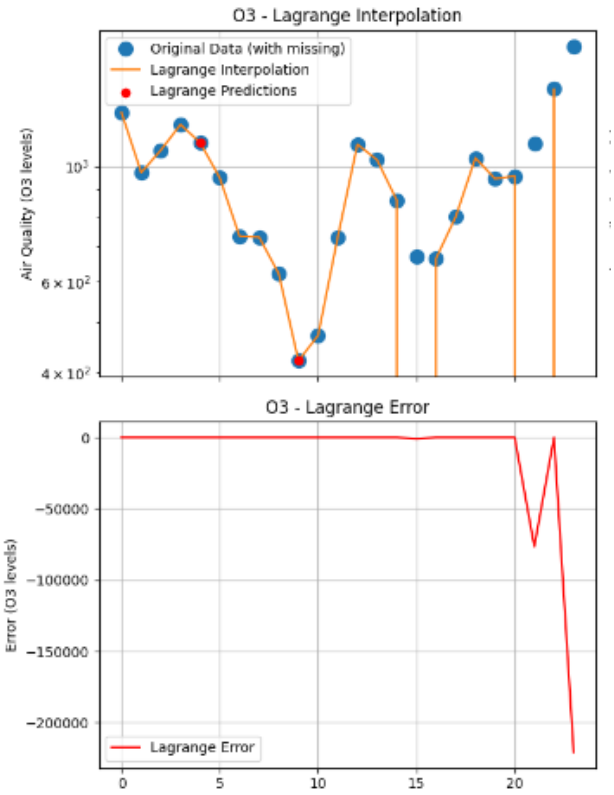

Fig. 17. Ozone Lagrange Interpolation

2) Newton's Divided Differences: Newton's method provided a noticeable improvement in predicting O3 levels, with smoother transitions and fewer oscillations compared to Lagrange interpolation. However, the method was still limited in capturing some of the sharp peaks in ozone concentrations.

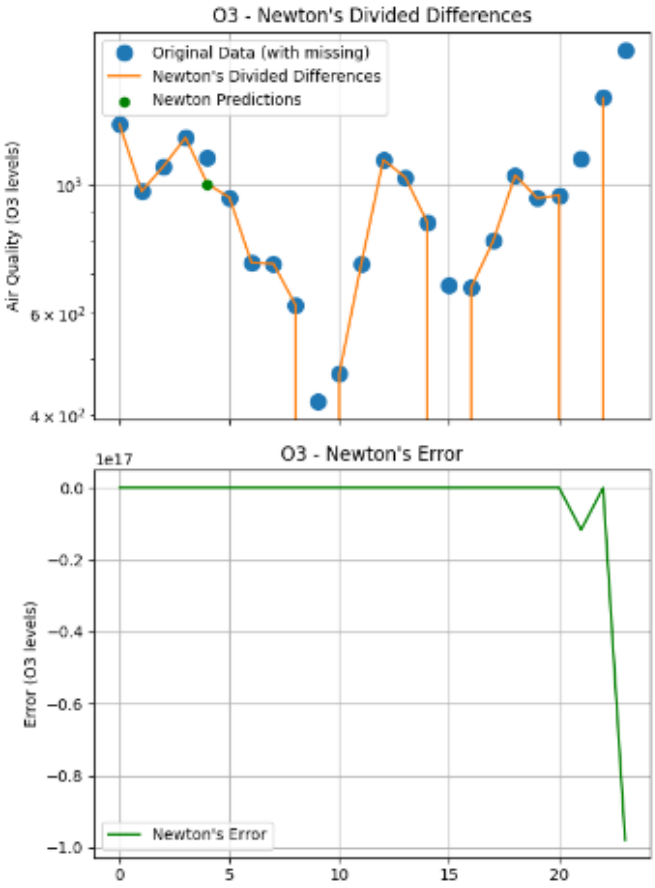

Fig. 18. Ozone Newton's Divided Differences

3) Cubic Splines: As with CO and NO2, cubic splines proved to be the most effective technique for O3 predictions. The method's ability to balance local accuracy and global smoothness resulted in accurate approximations of both baseline levels and peak events. The error metrics confirmed its superior performance.

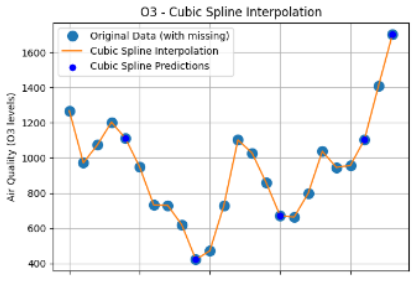

Fig. 19. Ozone Cubic spline Interpolation

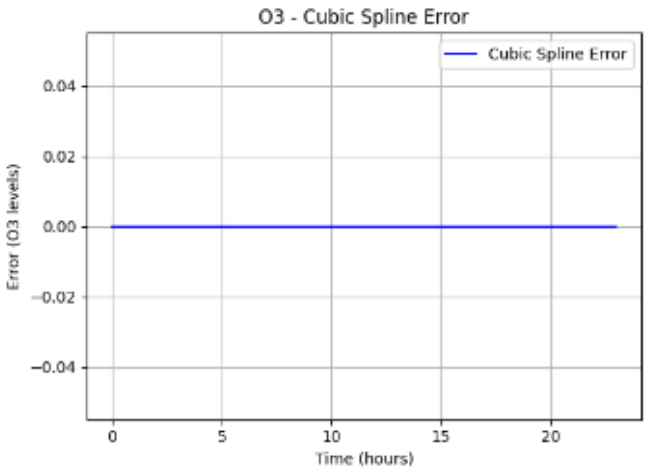

Fig. 20. Ozone Cubic spline Error

D. Conclusion

The results demonstrate the varying effectiveness of interpolation techniques in predicting air quality metrics. While Lagrange interpolation is useful for small datasets, its limitations in handling larger, complex datasets are evident. Newton's Divided Differences offer a more balanced approach, with reduced errors and better stability. However, cubic splines emerged as the most reliable method, providing accurate and smooth predictions for CO, NO2, and O3 across all time intervals. These findings underscore the importance of selecting appropriate interpolation techniques for environmental data modeling.

REFERENCES

[1] M. Beauchamp, L. Malherbe, C. de Fouquet, L. Létinois, and F. Tognet, "A polynomial approximation of the traffic contributions for kriging-based interpolation of urban air quality model," *Environmental Modelling & Software*, vol. 105, pp. 132–152, 2018.

[2] S.-Å. Gustafson, K. Kortanek, and J. R. Sweigart, "Numerical optimization techniques in air quality modeling: objective interpolation formulas for the spatial distribution of pollutant concentration," *Journal of Applied Meteorology and Climatology*, vol. 16, no. 12, pp. 1243–1255, 1977.

[3] D. Deligiorgi and K. Philippopoulos, "Spatial interpolation methodologies in urban air pollution modeling: application for the greater area of metropolitan athens, greece," *Advanced air pollution*, vol. 17, pp. 341–362, 2011.

[4] U. I. M. L. Repository. (2016) Air quality. [Online]. Available: