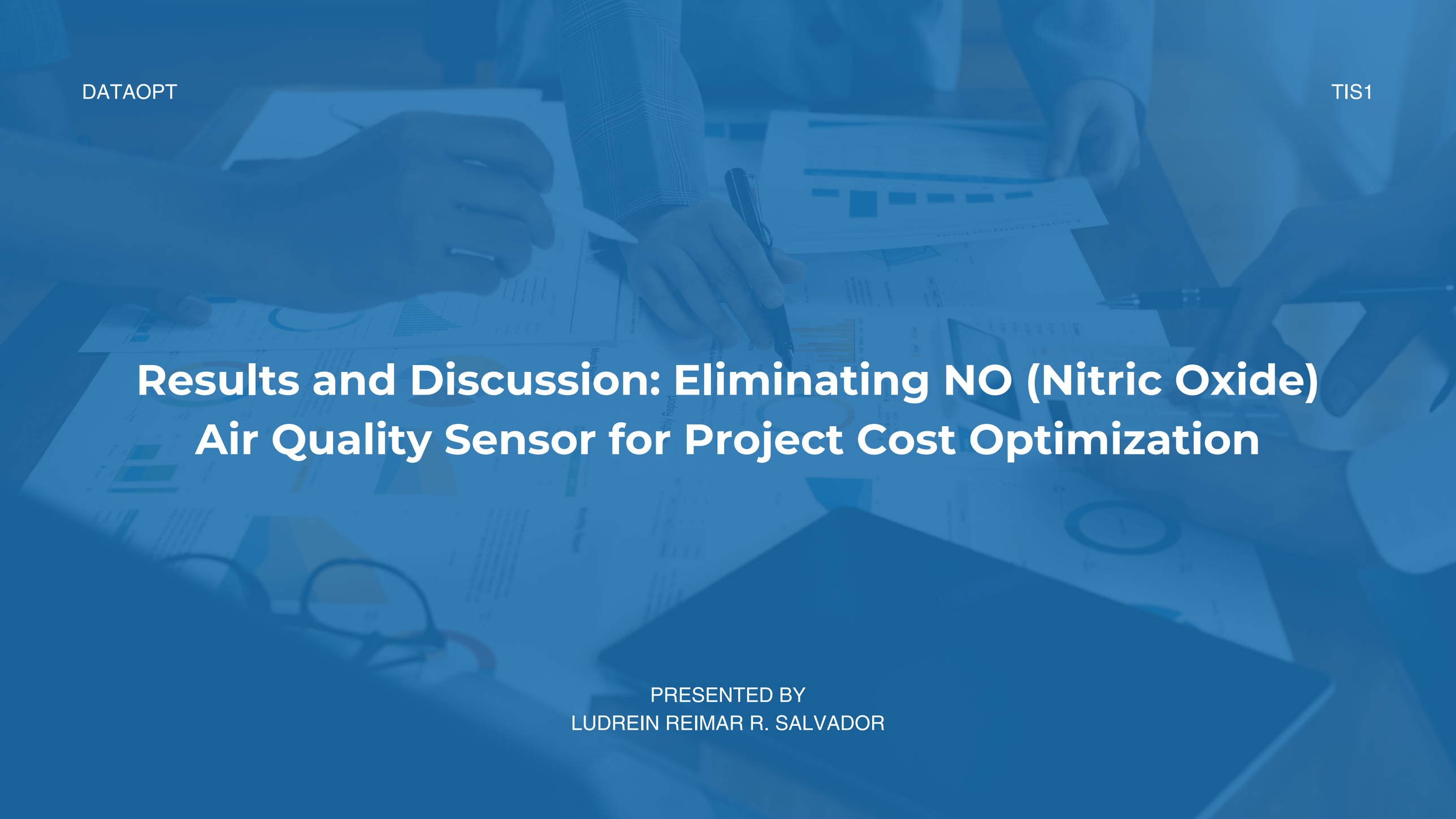
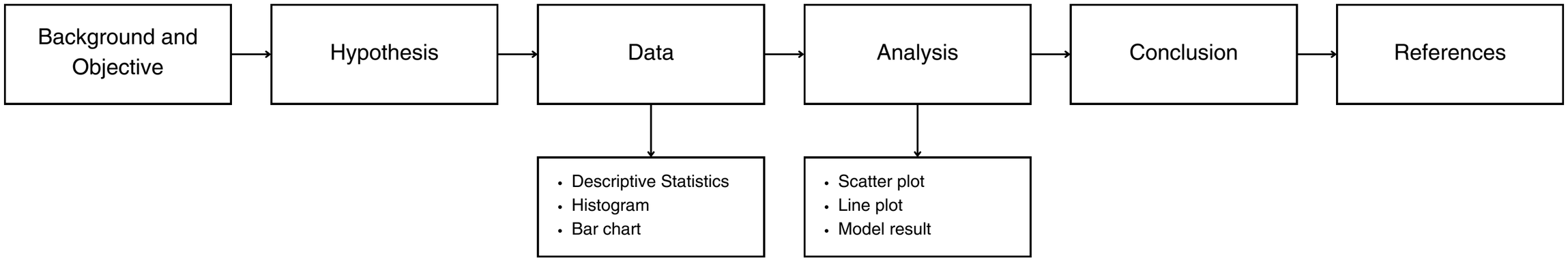


Results and Discussion: Eliminating NO (Nitric Oxide) Air Quality Sensor for Project Cost Optimization

PRESENTED BY
LUDREIN REIMAR R. SALVADOR

OVERVIEW


Background and Objective
Hypothesis
Data
Analysis
Conclusion
References
Descriptive Statistics
Histogram
Bar chart
Scatter plot
Line plot
Model result

BACKGROUND AND OBJECTIVE

Low-cost air quality sensors **can replace traditional monitoring systems**, which are often costly and require significant infrastructure (Great Basin Unified Air Pollution Control District, n.d.).

These sensors **provide real-time data** on various pollutants at a fraction of the cost of regulatory-grade monitors, making them **accessible for community use** (Davda, 2024).

When properly validated, low-cost sensors can achieve 80% to **90% accuracy** compared to reference monitors, though they **may not fully replace traditional sensors** due to potential data discrepancies (Kang et al., 2021).

Their portability allows for dense monitoring networks in underserved areas, **enhancing public health initiatives** (World Meteorological Organization, 2024; Kunak Technologies S.L., 2023).

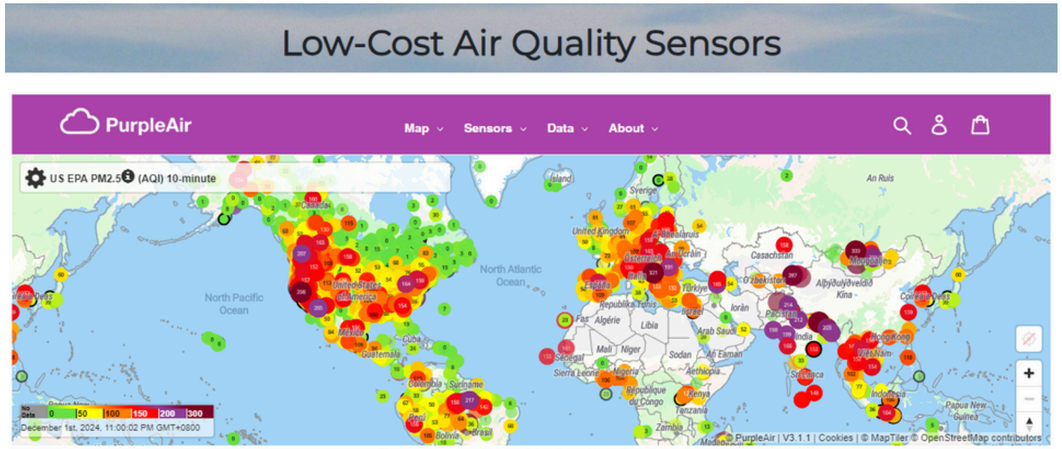


Fig. 1. Low-Cost Air Quality Sensors
Image Source: GBUAPCD

Fig. 2. GAW report n° 293 - cover image
Image Source: World Meteorological Organization, 2024

Fig. 3. Low-cost Air Quality Monitoring
Image Source: Kruti Davda, 2024

BACKGROUND AND OBJECTIVE

- Significant project cost savings are possible using **inexpensive air quality sensors**, which can be up to **40 times less expensive than conventional monitors**.

- ‘no’ levels can be calculated from ‘no2’ and ‘nox’, indicating that deleting the **‘no’** sensor will have **no effect on data integrity** (Kang et al., 2021).

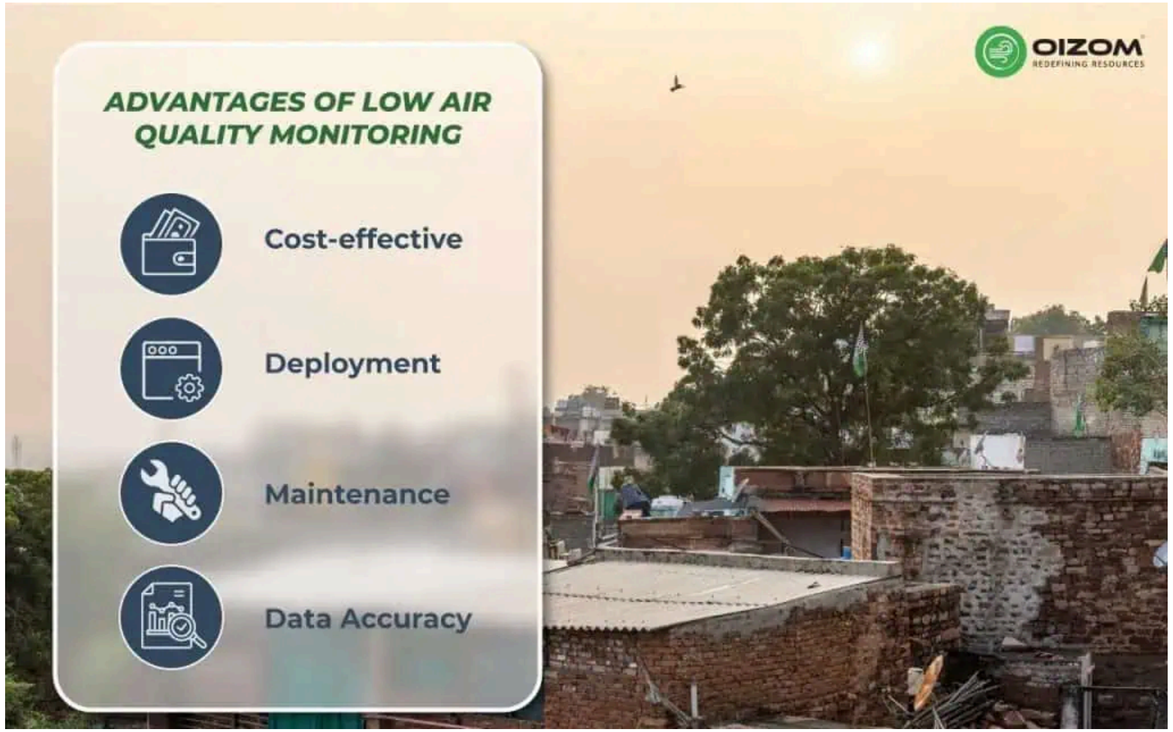


Fig. 4. Advantages of Low Air Quality Monitoring
Image Source: Kruti Davda, 2024

BACKGROUND AND OBJECTIVE

- By enabling extensive monitoring networks, these sensors **efficiently track pollution without the need for separate sensors**.

- Accessible air quality data from inexpensive sensors supports public health campaigns and **keeps the community informed even in the event that one sensor is removed** (Kunak Technologies S.L., 2023).

Objective

- Identify which sensor can be eliminated to optimally reduce cost.

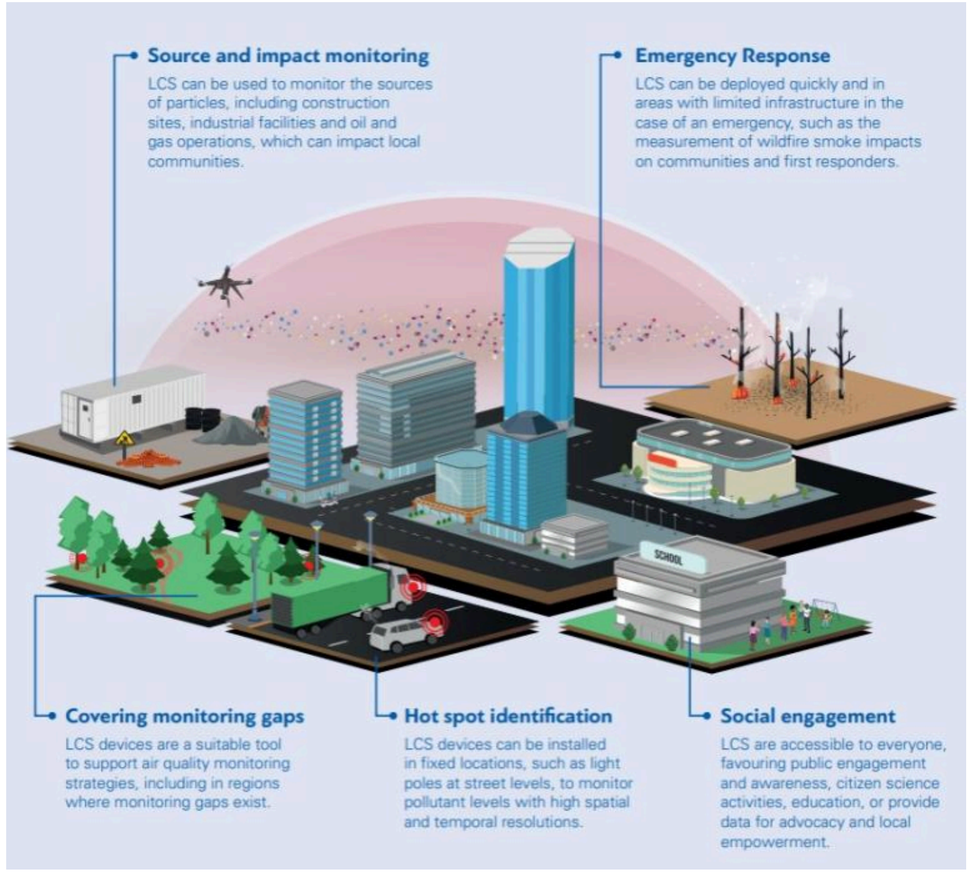


Fig. 5. Low-cost Sensor Systems (LCS)
Image Source: World Meteorological Organization, 2024

HYPOTHESIS

Eliminating the ‘no’ sensor (Nitric Oxide) in N. Mai, Los Angeles California (CA), will have a minimal impact on overall air quality monitoring. This is based on the strong correlation, interdependence, or redundancy of ‘no’ with other related pollutants, such as ‘no2’ and ‘nox’. By leveraging data from these sensors, it can effectively derive ‘no’ levels, thereby optimally reducing project expenses while maintaining the integrity of air quality data.

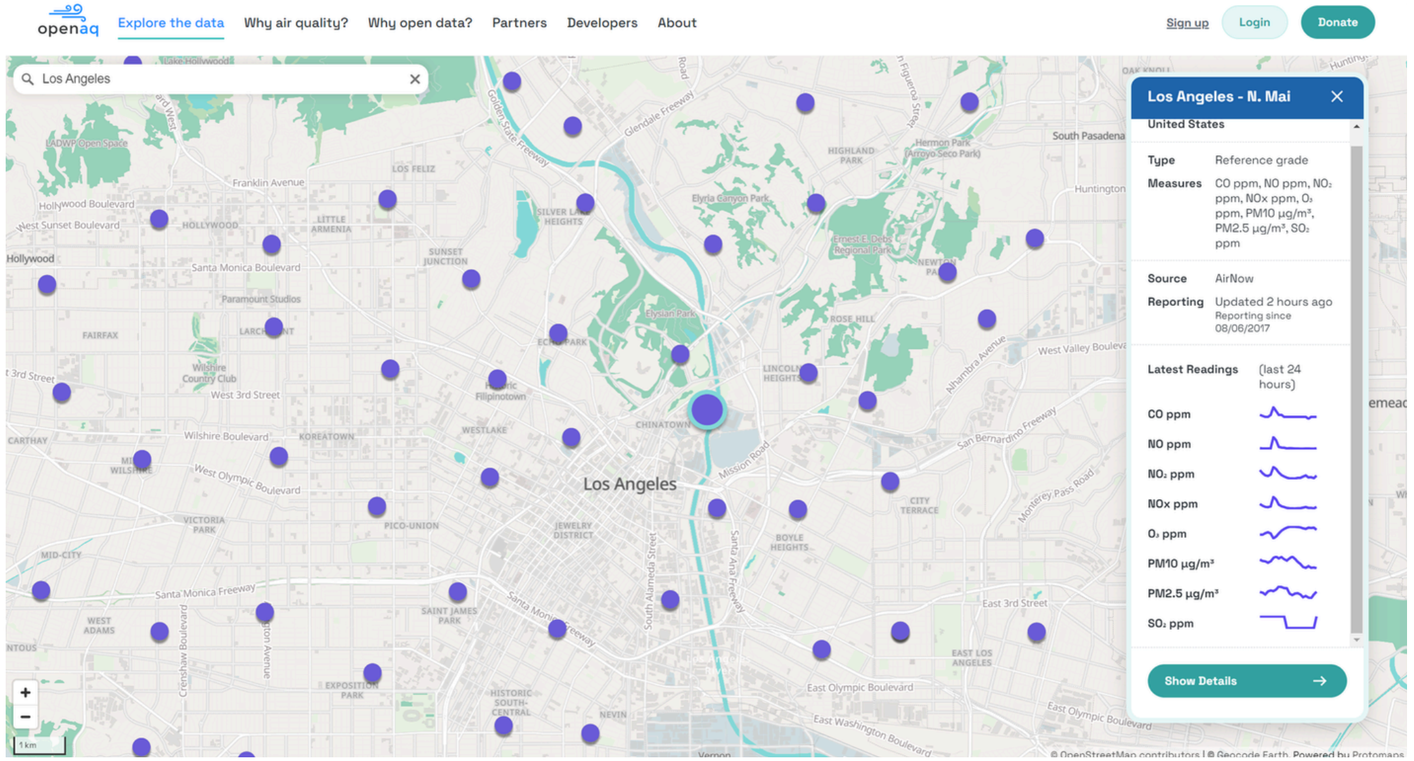


Fig. 6. N. Mai, Los Angeles, CA
Image Source: OpenAQ Explorer

DATA

- Installed the necessary libraries such as pyspark, findspark, and installed awscii or the AWS 3 for data pulling from my chosen **location ID (7936)** which is from **Los Angeles, CA**, specifically at **N. Mai**.

```
[9]  # Created a directory for the data
     !mkdir raw_7936
     !ls
```

```
raw_7936  sample_data
```

- After creating the directory for the dataset, I extracted data from AWS S3 openaq-data-archive similar with previous coding exercises

```
[10] # Extracted data from AWS S3 openaq-data-archive similar with previous coding exercises.
     !aws s3 cp --recursive --no-sign-request s3://openaq-data-archive/records/csv.gz/locationid=7936/ raw_7936
```

```
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170610.
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170611.
```

DATA

- The dataset contains columns such as location_id, sensors_id, datetime, parameter, units, and value.
- The primary focus is on the parameter (type of pollutant, e.g., so2, pm10) and value (measured pollutant concentration).

```
[11] # Defined the 7938 and displayed top 5 rows from the dataset
     df_7936 = spark.read.csv('/content/raw_7936/*/*/', inferSchema=True, header=True)
     df_7936.show(5)
```

```
+-----------+----------+--------------------+--------------------+---------+--------------------+---------+-----+-----+
|location_id|sensors_id|            location|            datetime|      lat|                 lon|parameter|units|value|
+-----------+----------+--------------------+--------------------+---------+--------------------+---------+-----+-----+
|       7936|     25195|Los Angeles - N. ...|2024-10-25T01:00:...|34.066429|-118.22675500000001|     pm10|µg/m³| 30.0|
|       7936|     25195|Los Angeles - N. ...|2024-10-25T02:00:...|34.066429|-118.22675500000001|     pm10|µg/m³| 31.0|
|       7936|     25195|Los Angeles - N. ...|2024-10-25T03:00:...|34.066429|-118.22675500000001|     pm10|µg/m³| 22.0|
|       7936|     25195|Los Angeles - N. ...|2024-10-25T04:00:...|34.066429|-118.22675500000001|     pm10|µg/m³| 28.0|
|       7936|     25195|Los Angeles - N. ...|2024-10-25T05:00:...|34.066429|-118.22675500000001|     pm10|µg/m³| 27.0|
+-----------+----------+--------------------+--------------------+---------+--------------------+---------+-----+-----+
only showing top 5 rows
```

```
[12] # Displayed the number of rows from the dataset
     df_7936.count()
```

```
301798
```

DATA

Summary Statistics

- The dataset contains columns such as location_id, sensors_id, datetime, parameter, units, and value.

```
[14] # Displayed the summary of the data and its fields
     df_7936.summary().show()
```

summary	location_id	sensors_id	location	datetime	lat	lon	parameter	units	value
count	301798	301798	301798	301798	301798	301798	301798	301798	301798
mean	7936.0	419100.41071842756	NULL	NULL	34.06642899999882	-118.22675500000167	NULL	NULL	6.79443538128152
stddev	0.0	1232511.5536995134	NULL	NULL	0.0	1.164382201354442...	NULL	NULL	13.225674480539892
min	7936	23019	Los Angeles - N. ...	2017-06-07T14:00:...	34.066429	-118.226755	co	ppm	-0.0001
25%	7936.0	25192.0	NULL	NULL	34.066429	-118.22675500000001	NULL	NULL	0.006
50%	7936.0	25194.0	NULL	NULL	34.066429	-118.226755	NULL	NULL	0.038
75%	7936.0	25195.0	NULL	NULL	34.066429	-118.226755	NULL	NULL	9.4
max	7936	4272361	Los Angeles - N. ...	2024-11-28T00:00:...	34.066429	-118.22675500000001	so2	µg/m³	99.0

DATA

Summary Statistics

parameter	count	avg	stddev	min	p25	median	p75	max	count_null	count_zero
so2	46170	2.160298895386617E-4	4.041309755105069E-4	-0.001	0.0	0.0	2.0E-4	0.01	0	28635
co	41529	0.39307977557851115	0.25275618507603265	0.0	0.2	0.3	0.5	2.0	0	2
nox	14004	0.02213774635818337	0.020635147511218806	8.0E-4	0.0079	0.0144	0.0288	0.16219999999999998	0	0
o3	46634	0.02498601878457775	0.018367949607866096	0.0	0.008	0.025	0.038	0.138	0	1352
pm10	46675	29.853519014461703	16.26855114320419	-4.0	19.0	28.0	38.0	588.0	0	27
no2	46689	0.017487442438261672	0.011311562242212183	6.0E-4	0.008	0.0147	0.025	0.08	0	0
no	14008	0.006463370930896629	0.012450976576445411	-9.0E-4	3.0E-4	0.0013	0.006	0.12290000000000001	0	631
pm25	46089	13.85184317299139	9.22071562513049	-3.8	8.0	12.0	17.4	508.0	0	91

Based on the low mean and limited range of no, no2, and nox, as well as their good correlation, these sensors might provide overlapping data or **potential redundancy**.

DATA

I. Descriptive Statistics

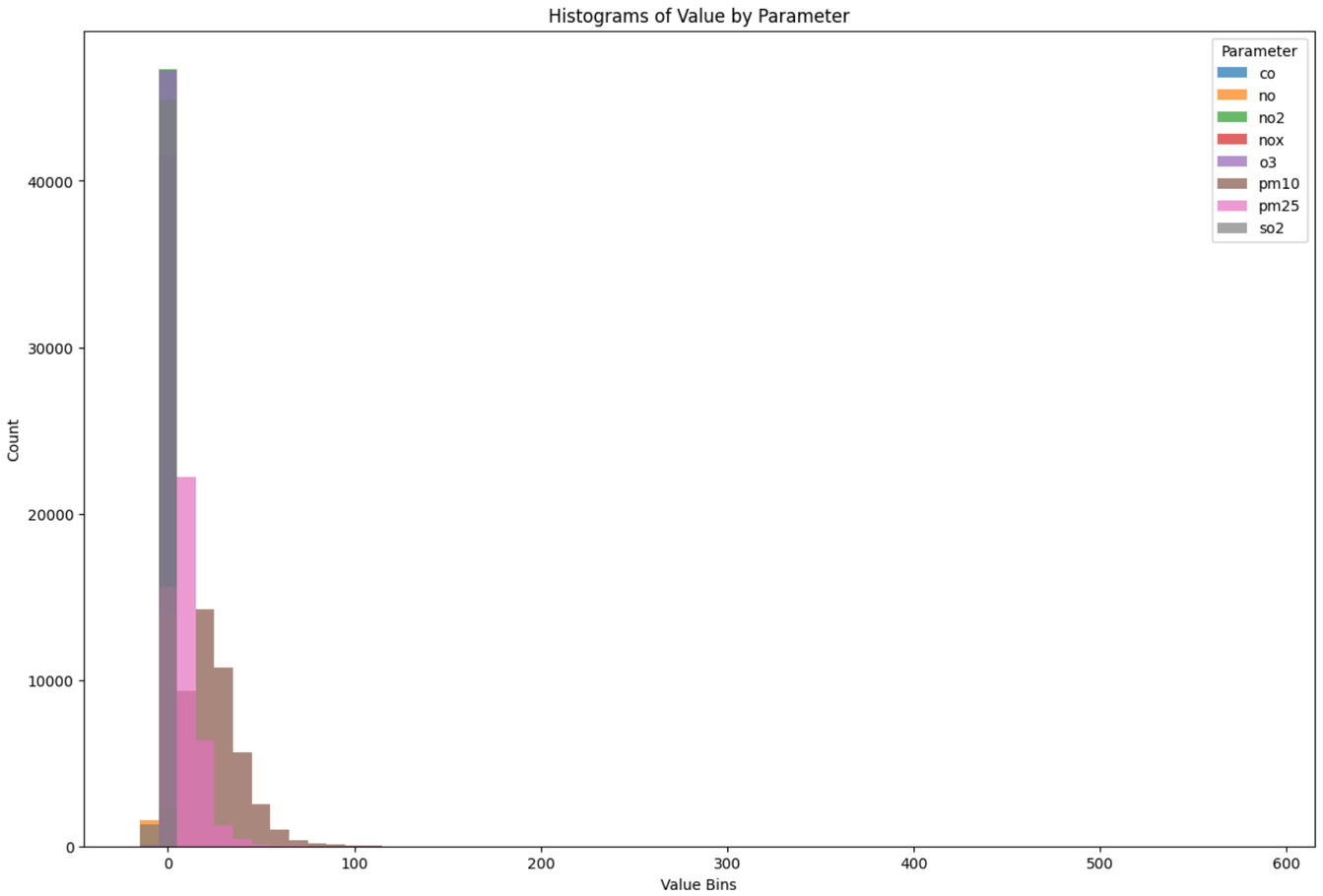
```
[28] from pyspark.sql.functions import col, count, avg, stddev, min, max, percentile_approx

     # This will be the descriptive statistics for each sensor
     descriptive_stats = df_7936.groupBy("parameter").agg(
         count("value").alias("count"),
         avg("value").alias("mean"),
         stddev("value").alias("stddev"),
         min("value").alias("min"),
         percentile_approx("value", 0.25).alias("25th_percentile"),
         percentile_approx("value", 0.5).alias("median"),
         percentile_approx("value", 0.75).alias("75th_percentile"),
         max("value").alias("max")
     )

     # Displaying the actual descriptive statistics
     descriptive_stats.show(truncate=False)
```

parameter	count	mean	stddev	min	25th_percentile	median	75th_percentile	max
so2	46170	2.160298895386617E-4	4.041309755105069E-4	-0.001	0.0	0.0	2.0E-4	0.01
co	41529	0.39307977557851115	0.25275618507603265	0.0	0.2	0.3	0.5	2.0
nox	14004	0.02213774635818337	0.020635147511218806	8.0E-4	0.0079	0.0144	0.0288	0.16219999999999998
o3	46634	0.02498601878457775	0.018367949607866096	0.0	0.008	0.025	0.038	0.138
pm10	46675	29.853519014461703	16.26855114320419	-4.0	19.0	28.0	38.0	588.0
no2	46689	0.017487442438261672	0.011311562242212183	6.0E-4	0.008	0.0147	0.025	0.08
no	14008	0.006463370930896629	0.012450976576445411	-9.0E-4	3.0E-4	0.0013	0.006	0.12290000000000001
pm25	46089	13.85184317299139	9.22071562513049	-3.8	8.0	12.0	17.4	508.0

- ‘no’ sensor count is 14,008 from the data.
- **‘no’** levels are **continuously low** and **redundant** with no2 and nox.

DATA

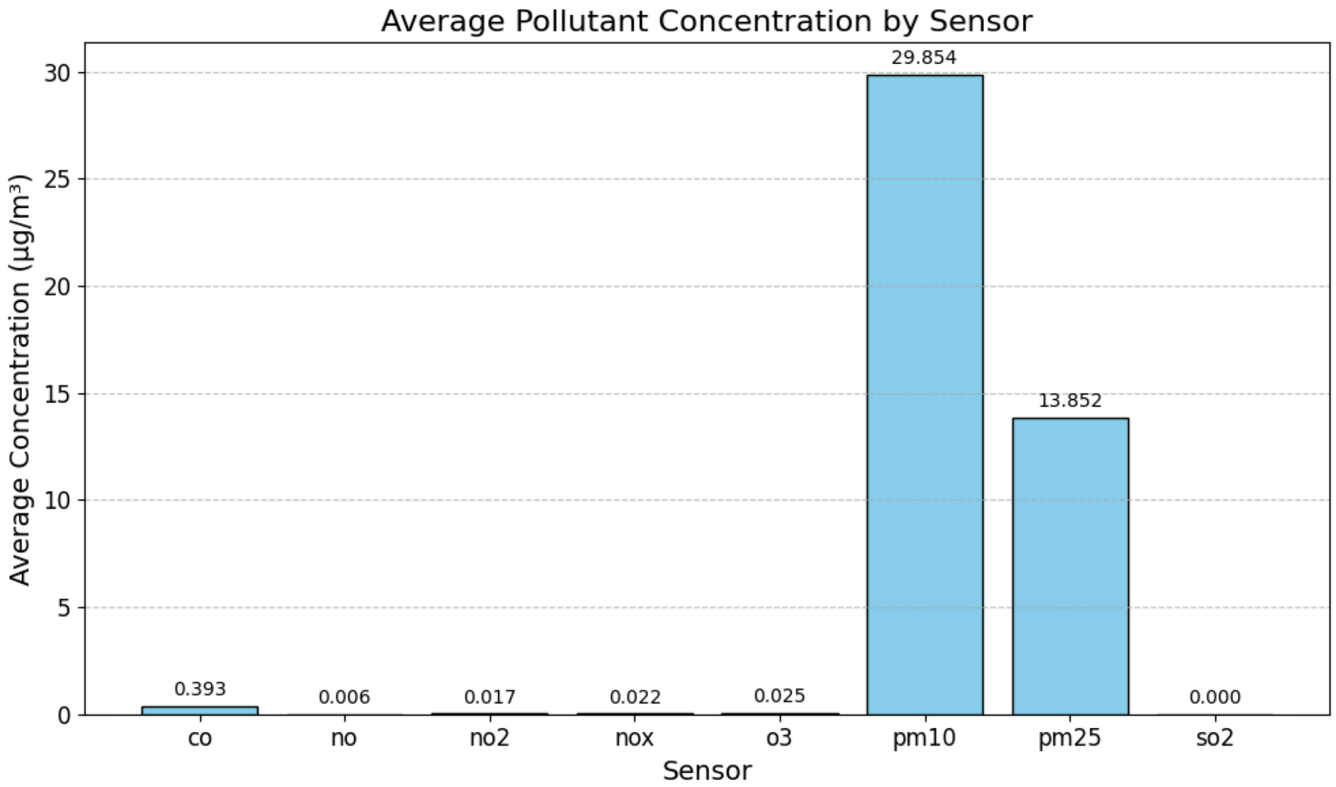
II. Histogram



- Each bar's height indicates the count.
- Majority of measurements for ‘no’ cluster is just a **small percentage in the range**.

DATA

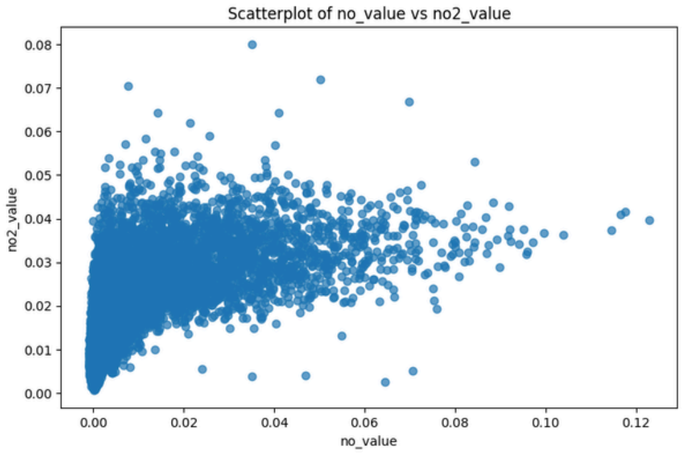
III. Bar Chart

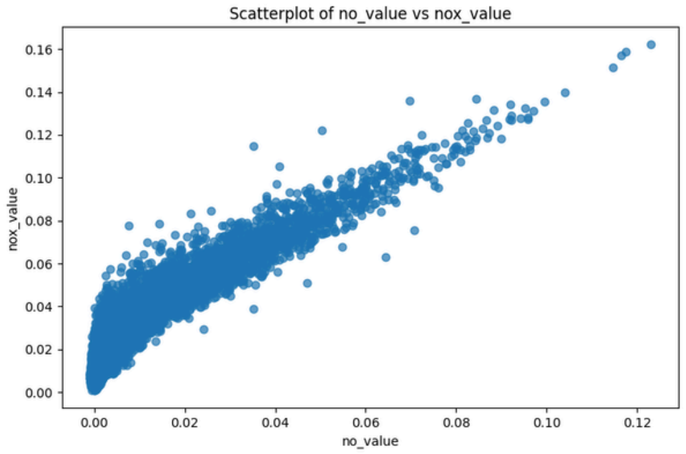


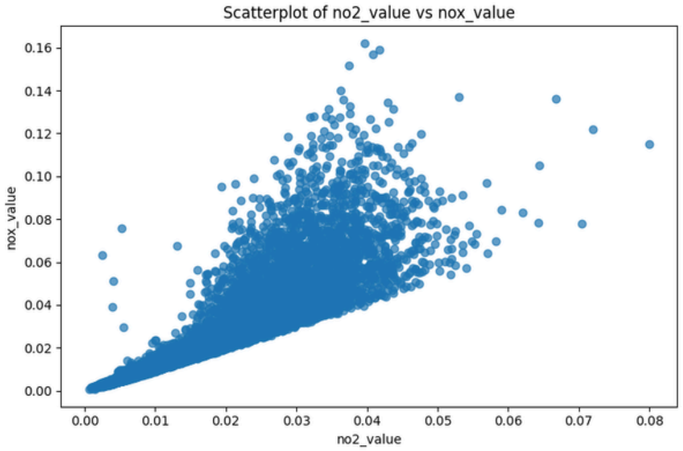
- no (Nitric Oxide) - one of the lowest average concentration, suggests that it might **have less effect on the environment**.
- no2 (Nitrogen Dioxide) - Linked to combustion or burning processes (like automobiles, industries), **higher concentration than 'no'**.
- nox (Nitrogen Oxides) - total of no and no2, it has a larger value than 'no' but is **still very low overall**.

ANALYSIS

I. Scatter Plot







[1] no vs no2

- There is **some correlation** between ‘no’ and ‘no2’, according to the data points. Little disorganized connection.
- Not entirely dependent on one another.

[2] no vs. nox

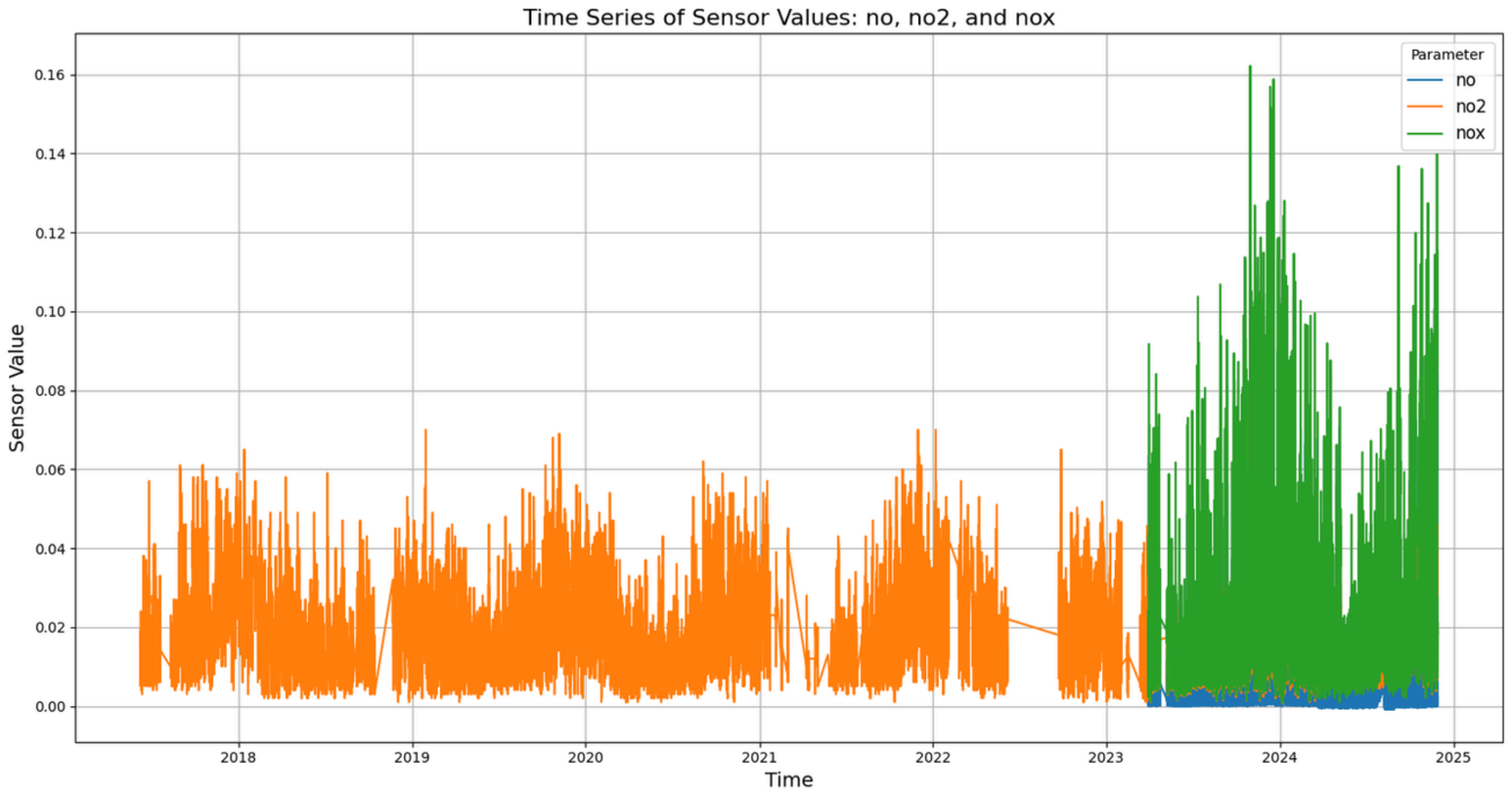
- Tight rising trend, **strong positive correlation** between ‘no’ and ‘nox’.
- ‘nox’ is sum of ‘no’ and ‘no2’, this implies that ‘no’ levels significantly contribute to ‘nox’ levels.

[3] no2 vs. nox

- ‘no2' and ‘nox’ have a **high positive correlation**, just like [2].
- This shows how ‘no2' and ’nox’ are interdependent. Changes in ‘no2’ result in ‘nox’.

ANALYSIS

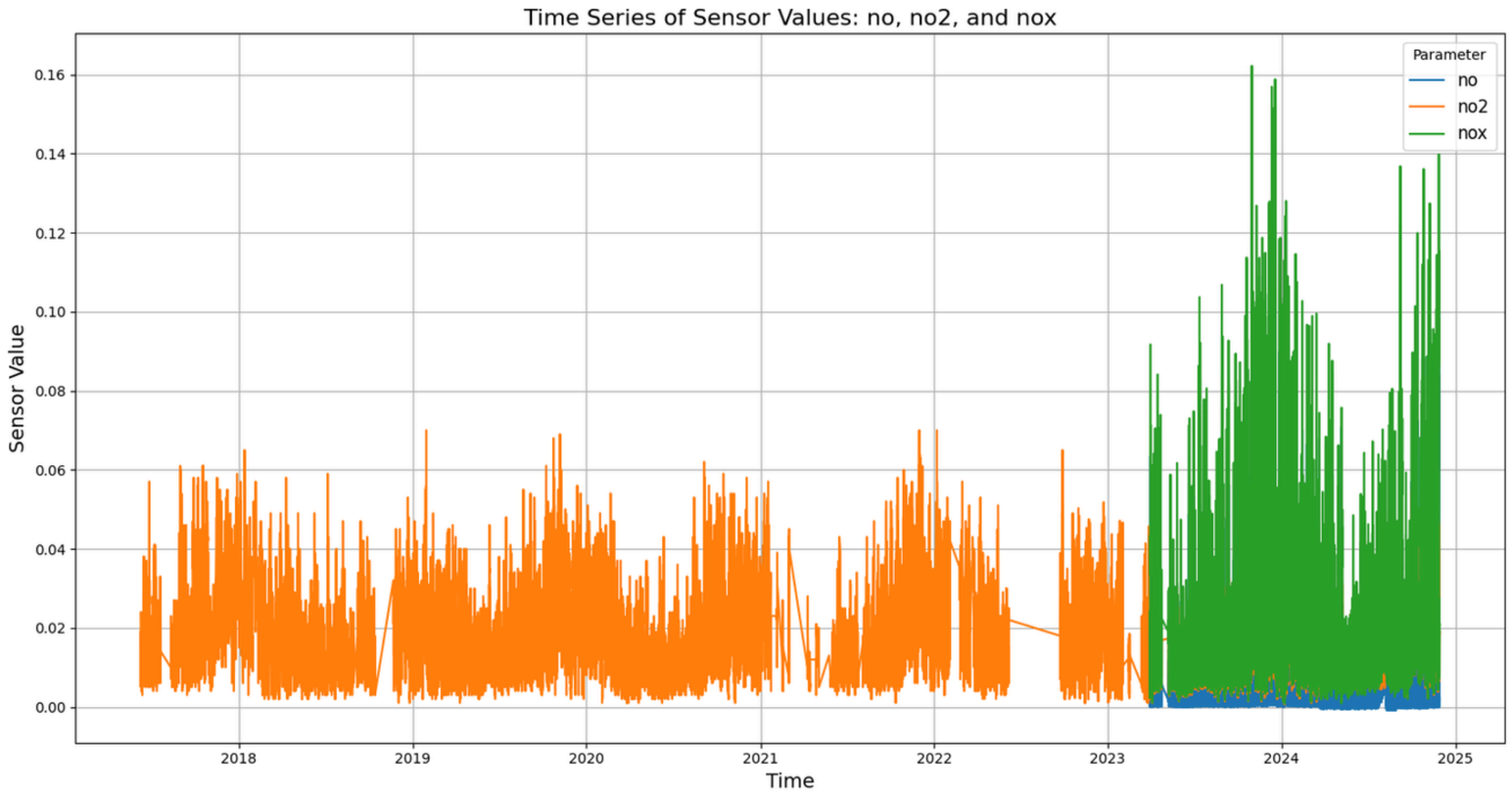
II. Line Plot



- Between 2022 and 2023, there is a **noticeable gap for every metric**, a sign of **missing data** or **sensor failure**.
- ‘nox’ has seen a significant increase in results following 2023, indicating either changes to the environment, mistakes in measurement, or adjustments to the data collection process.
- **Possibly related to environmental cycles** like temperature or human activity, ‘no2’ shows consistent seasonal or periodic behavior.

ANALYSIS

II. Line Plot



- Changes in the real world, including **increased industrial activity**, **traffic**, or **changing seasons in pollutant concentrations**, may be the cause of the ‘no2’ moves and the ‘nox’ increase.
- Data gap can be sensors were not working or when problem gathering the data.
- Correlations - the patterns in ‘no2’ and ‘nox’ may be connected.

ANALYSIS

III. Model Result

```
[72]  # Creating and training the linear regression model
      lr = LinearRegression(featuresCol="features", labelCol="label")
      lr_model = lr.fit(train_data)

[73]  # Evaluating the model
      test_results = lr_model.evaluate(test_data)

[74]  # This will display and print evaluation metrics
      print(f"R²: {test_results.r2}")
      print(f"RMSE: {test_results.rootMeanSquaredError}")

      R²: 0.9998760560412906
      RMSE: 0.00013389460874487974

[75]  # Showing coefficients and intercept
      print(f"Coefficients: {lr_model.coefficients}")
      print(f"Intercept: {lr_model.intercept}")

      Coefficients: [-1.0005450970376846,1.0026874550856768]
      Intercept: -2.683898681278864e-05
```

- Using ‘no2’ and nox, the model can virtually exactly predict ‘no’ values. This shows that there is not much unique information being added by the sensor.
- It is simple to estimate without the ‘no’ sensor since the coefficients demonstrate that ‘no’ has a distinct and expected relationship with ‘no2’ and ‘nox’.
- Possibility of reducing cost - Removing the sensor **could result in lower maintenance** or **sensor purchases** because it is **not necessary for accurate forecasts**.
- Reliable model - It is feasible to rely on forecasts rather than the actual sensor because the model is strong and **fits the data** very well.

CONCLUSION

Eliminating the ‘no’ (Nitric Oxide) sensor will not have a major effect on air quality monitoring, according to the procedure and analysis. A reliable linear regression model can accurately predict ‘no’ values due to the strong linear relationship between ‘no’, ‘no2’, and ‘nox’.

The regression model's strong and incredibly low RMSE which supports the claim that the ‘**no’ sensor** is **redundant** and that its values can be derived from ‘no2’ and ‘nox’. These findings suggest a significant connection between ‘no’, ‘no2’, and ‘nox’.

Without sacrificing the accuracy of the data from air quality monitoring, removing the ‘no’ sensor from the designated area (N. Mai, Los Angeles, CA) **will lower expenses** associated with **equipment purchase** or **maintenance**. This is in line with the project's goal.

This shows how correlations and predictive modelling can be used to optimize sensors in air quality monitoring systems. This method can be expanded to assess more sensors in comparable setups.

REFERENCES

Davda, K. (2024, June 27). *What is low-cost air quality monitoring, and what are its Working principles?* Oizom. https://oizom.com/what-is-low-cost-air-quality-monitoring/

DD-Scientific. (n.d.). *GS+7NO Nitric Oxide (NO) Sensor | Industrial specification*. https://ddscientific.com/products/gs-7no-electrochemical-sensor-nitric-oxide-no

Great Basin Unified Air Pollution Control District (n.d.). *Low-Cost Air Quality Sensors*. https://www.gbuapcd.org/AirMonitoringData/LowCostSensors/

Kang, Y., Aye, L., Ngo, T. D., & Zhou, J. (2021). Performance evaluation of low-cost air quality sensors: A review. The Science of the Total Environment, 818, 151769–151769. https://doi.org/10.1016/j.scitotenv.2021.151769

Kunak Technologies S.L. (2023, June 30). *The power of low-cost air quality sensors for cleaner environments*. Kunak. https://kunakair.com/low-cost-air-quality-sensors/

OpenAQ Location ID 7936. (n.d.). OpenAQ Explorer. https://explore.openaq.org/locations/7936

World Meteorological Organization. (2024, June 13). *Low-cost sensors can improve air quality monitoring and people's health*. https://wmo.int/media/news/low-cost-sensors-can-improve-air-quality-monitoring-and-peoples-health

---

Others

- Google Colab Notebook Link
- salvadorl_week11.pdf

Thank You!

PRESENTED BY
LUDREIN REIMAR R. SALVADOR