SLIIT Datathon 2024

Optimizing Solar Power Efficiency through Advanced Anomaly Detection

---

By Cognic AI

1.1 Data Understanding and Preprocessing

Dataset Overview File: **`solar_sensor_data.csv`**

Preprocessing Steps

	type	count	nunique	null	mean	min	max
LOCATION	object	136476	2	0	NaN	NaN	NaN
DATE_TIME	object	136476	6417	0	NaN	NaN	NaN
SENSOR_ID	object	136476	44	0	NaN	NaN	NaN
DC_POWER	float64	136476	63581	0	1708.541497	0.000000	14471.125000
AC_POWER	float64	136476	62872	0	274.803511	0.000000	1410.950000
DAILY_YIELD	float64	136476	59249	0	3295.433783	0.000000	9873.000000
TOTAL_YIELD	float64	136476	70381	0	330382090.068492	0.000000	2247916295.000000

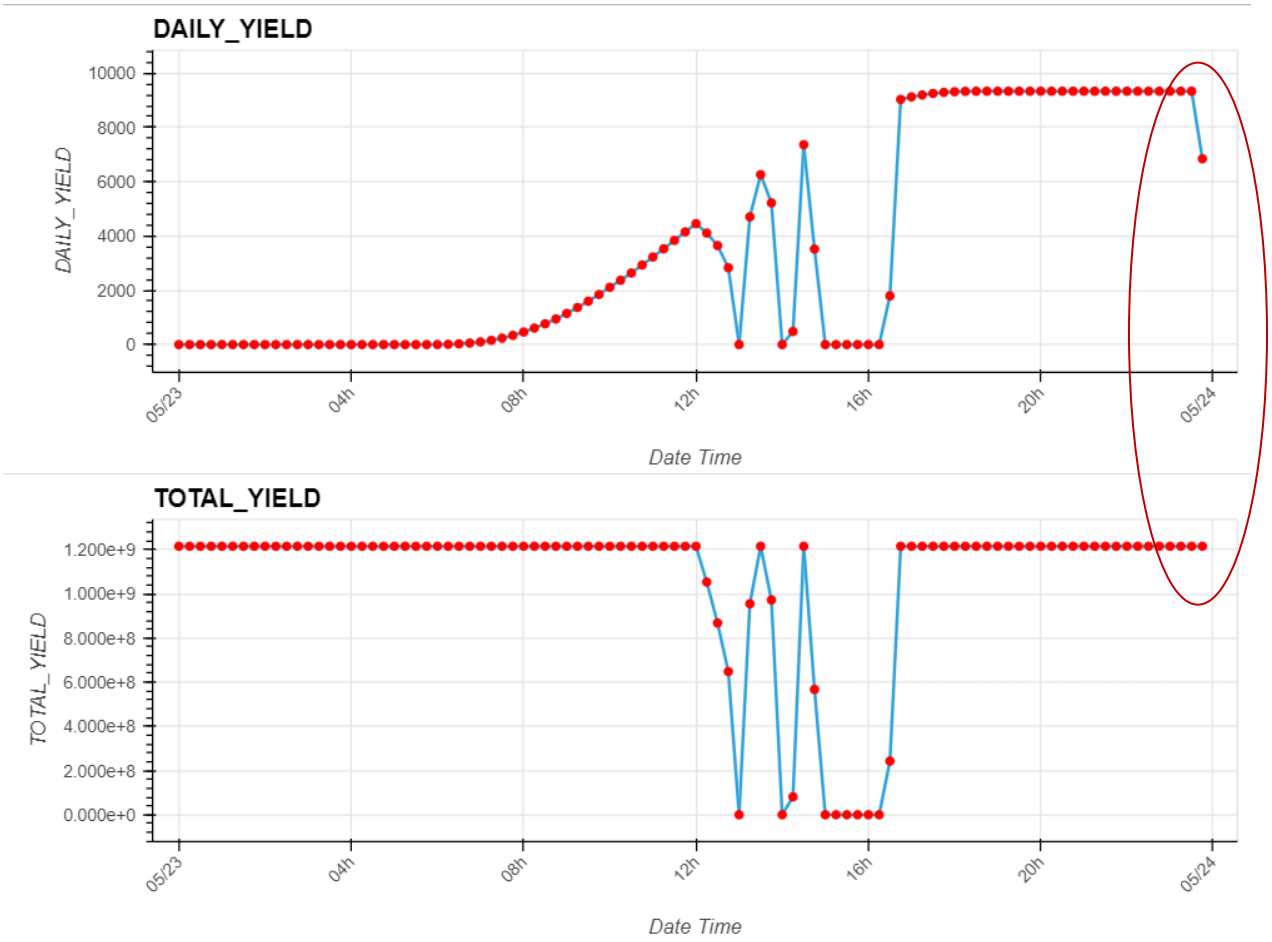
We notice that **DATE_TIME** in two locations are in two different formats. So, we took them both into '%Y-%m-%d %H:%M:%S' format.

Dataset Overview File: **`weather_sensor_data.csv`**

	type	count	nunique	null	mean	min	max
LOCATION	object	6441	2	0	NaN	NaN	NaN
DATE_TIME	object	6441	3262	0	NaN	NaN	NaN
AMBIENT_TEMPERATURE	float64	6441	6441	0	26.815672	20.398505	39.181638
MODULE_TEMPERATURE	float64	6441	6441	0	31.941762	18.140415	66.635953
IRRADIATION	float64	6441	3620	0	0.230551	0.000000	1.221652

Upon merging the weather and solar sensor datasets, we identified that the weather dataset contained four missing records compared to the time intervals in the solar sensor dataset.

	LOCATION	DATE_TIME	SENSOR_ID	DC_POWER	AC_POWER	DAILY_YIELD	TOTAL_YIELD	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION
38544	A	2020-06-03 14:00:00	sensor 2	7003.000000	685.800000	5601.000000	6330385.000000	NaN	NaN	NaN
38545	A	2020-06-03 14:00:00	sensor 25	7204.000000	705.400000	5685.000000	6419961.000000	NaN	NaN	NaN
38546	A	2020-06-03 14:00:00	sensor 39	7545.000000	738.700000	5579.000000	6928448.000000	NaN	NaN	NaN
38547	A	2020-06-03 14:00:00	sensor 42	7946.000000	777.800000	5541.000000	7152815.000000	NaN	NaN	NaN



Example : Sensor 6 on 2020-05-23

We observed that, in some instances, the DAILY_YIELD value changes while the TOTAL_YIELD remains unchanged. This issue is particularly noticeable when manually resetting the DAILY_YIELD to 0 at the start of a new day's recording. In these cases, the previous row's DAILY_YIELD is often lower than expected, likely due to delays in the reset process, resulting in incorrect data.

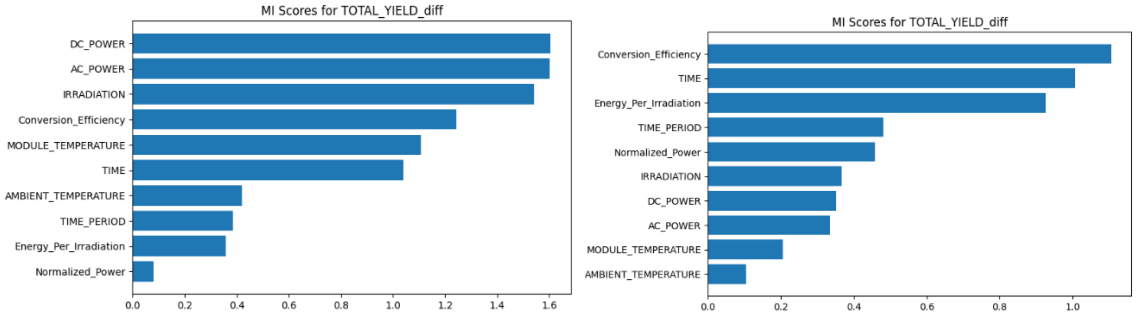
To address this, we identified the affected points and corrected the discrepancies by retaining the maximum DAILY_YIELD value from the previous day to ensure consistency with the TOTAL_YIELD.

1.2 Feature Selection and Engineering

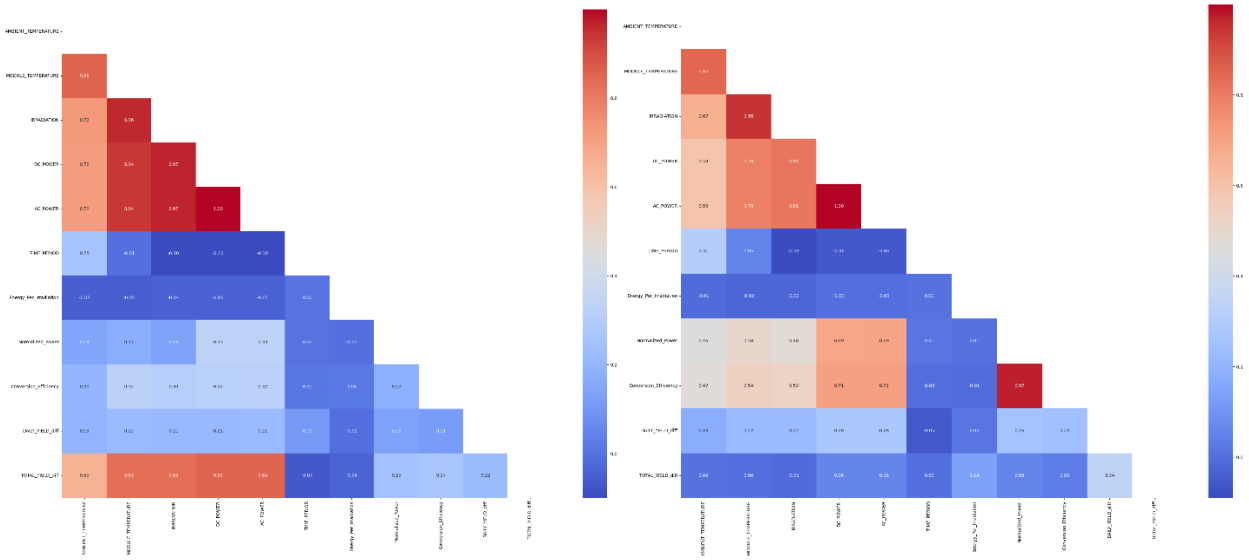
we implemented several transformations and calculated new features to enhance the dataset's predictive power and account for unique sensor-level behaviors:

1. **Sensor-Specific Features**:
    - For each unique SENSOR_ID, the following calculations were performed:
        - **Energy Per Irradiation**: Computed as the difference in TOTAL_YIELD divided by the difference in IRRADIATION.
        - **Normalized Power**: Calculated as DC_POWER divided by IRRADIATION.
        - **Conversion Efficiency**: Derived as AC_POWER divided by DC_POWER.
2. **Seasonal Classification**:
    - Months were grouped into four seasons:
        - **Season 1**: December to February
        - **Season 2**: March to May
        - **Season 3**: June to August
        - **Season 4**: September to November
3. **Time Period Segmentation**:
    - A TIME_PERIOD feature was added, based on the hour in the DATE_TIME column:
        - **Period 1**: Midnight to 6 AM
        - **Period 2**: 6 AM to Noon
        - **Period 3**: Noon to 3 PM
        - **Period 4**: 3 PM to 8 PM
        - **Period 5**: 8 PM to Midnight
4. **Correlation and MI Scores Analysis**:
    - We performed correlation analysis and calculated Mutual Information (MI) scores to understand relationships between features and target variables.
    - These analyses revealed variations in feature importance across different sensors, highlighting the need for sensor-specific modeling approaches.

Below is an example plot for MI scores and correlation analysis, showcasing sensor-level differences:



MI features on sensor 1 and 4



Correlation plots on sensor 12 and 18

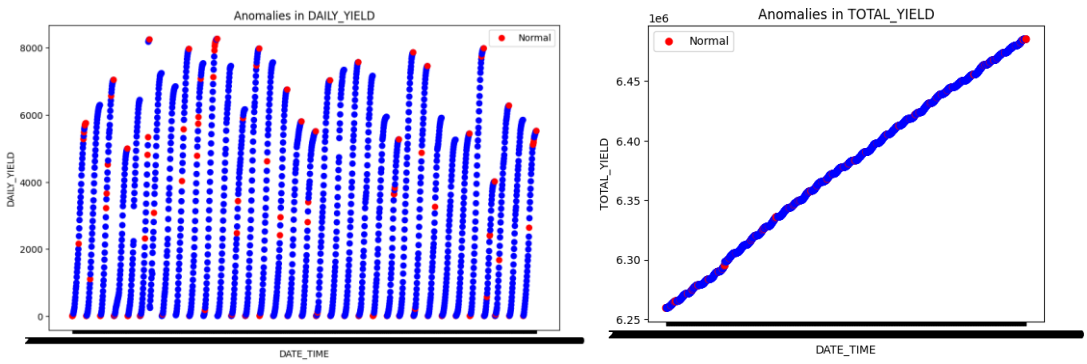
1.3 Anomalies Detection

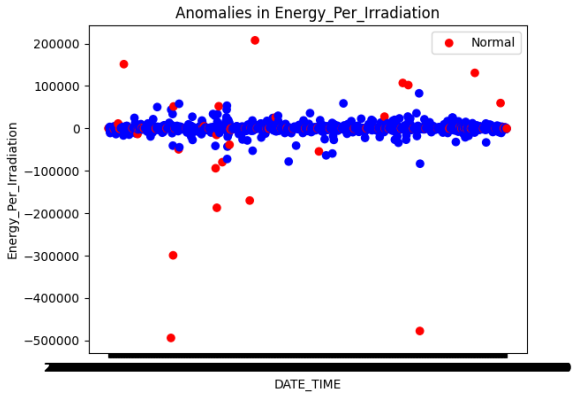
We implemented an anomaly detection process for all sensors using the Isolation Forest algorithm. This method helps identify and exclude anomalous data points, ensuring the final dataset contains only valid observations for subsequent analysis.

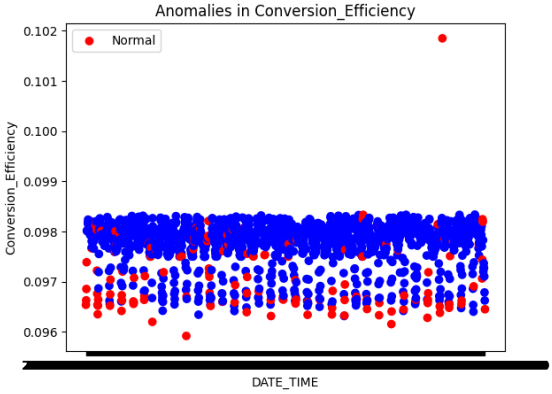
Steps Performed:

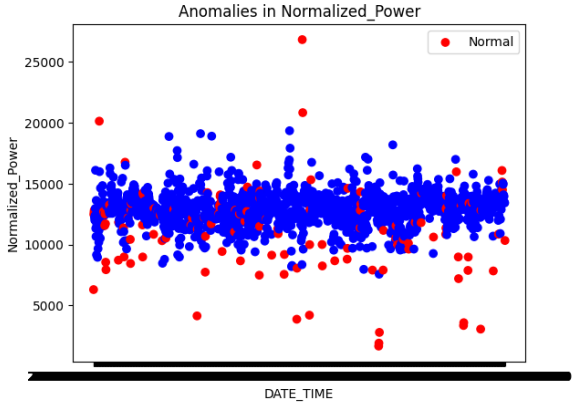
1. **Feature Selection for Anomaly Detection**:
   We selected the following features for identifying anomalies:
   - DAILY_YIELD
   - TOTAL_YIELD
   - TIME_PERIOD
   - AMBIENT_TEMPERATURE
   - MODULE_TEMPERATURE
   - IRRADIATION
   - Energy_Per_Irradiation
   - Normalized_Power
   - Conversion_Efficiency
2. **Model Configuration**:
   - Used an Isolation Forest model with the following parameters:
     - **n_estimators**: 200 (number of trees in the forest)
     - **contamination**: 0.05 (proportion of anomalies in the data)
     - **max_samples**: 70% of the data points
     - **random_state**: 0 (to ensure reproducibility)

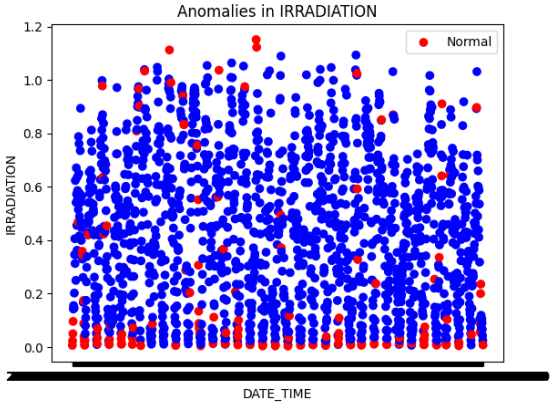
Example Visualization:
Below are example plots illustrating the anomaly detection results for one sensor.











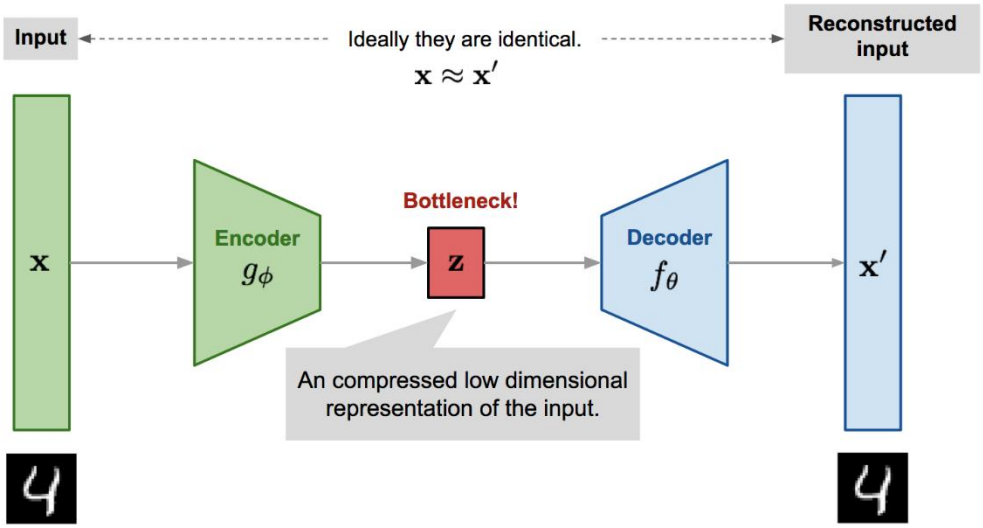
Anomalies Detection Model

Models Evaluated

1. CNN Autoencoders
2. Linear Regression
3. Random Forest Regressor
4. Gradient Boosting Regressor

1. **CNN Autoencoders**

The CNN Autoencoder model was designed to learn compact representations of the input data and reconstruct it. Anomalies were identified based on reconstruction errors, with higher errors indicating potential anomalies.

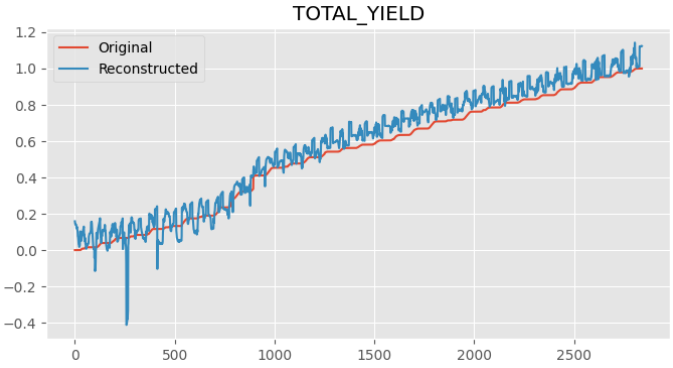


Results and Evaluation (Example: Sensor 6)

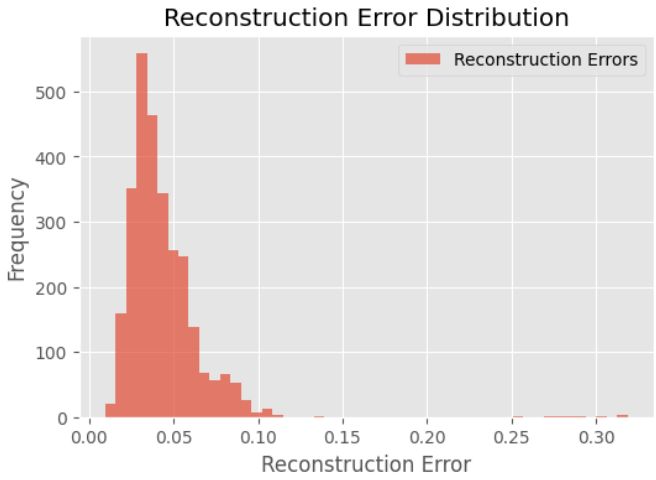
To assess the performance of the CNN Autoencoder for anomaly detection, we focused on individual sensor data. Below are the evaluation details for one sensor, as a representative example:

1. **Model Training Graph**:

2. **Original vs. Reconstructed Data**:

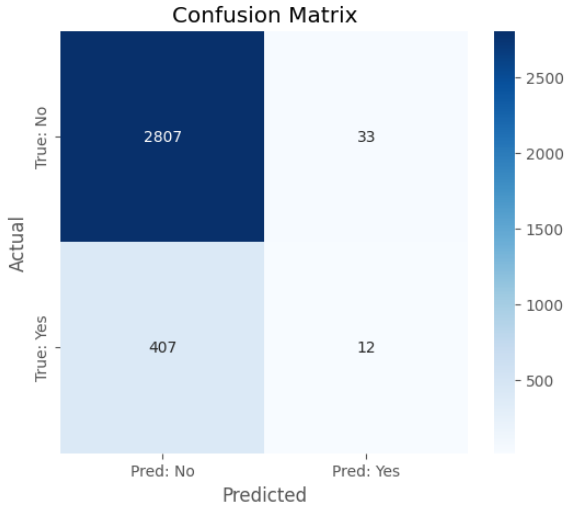


3. **Reconstruction Errors Distribution**:



4. **Accuracy and Confusion Matrix**:

   - **Accuracy**: 86.50%
   - **Confusion Matrix**:



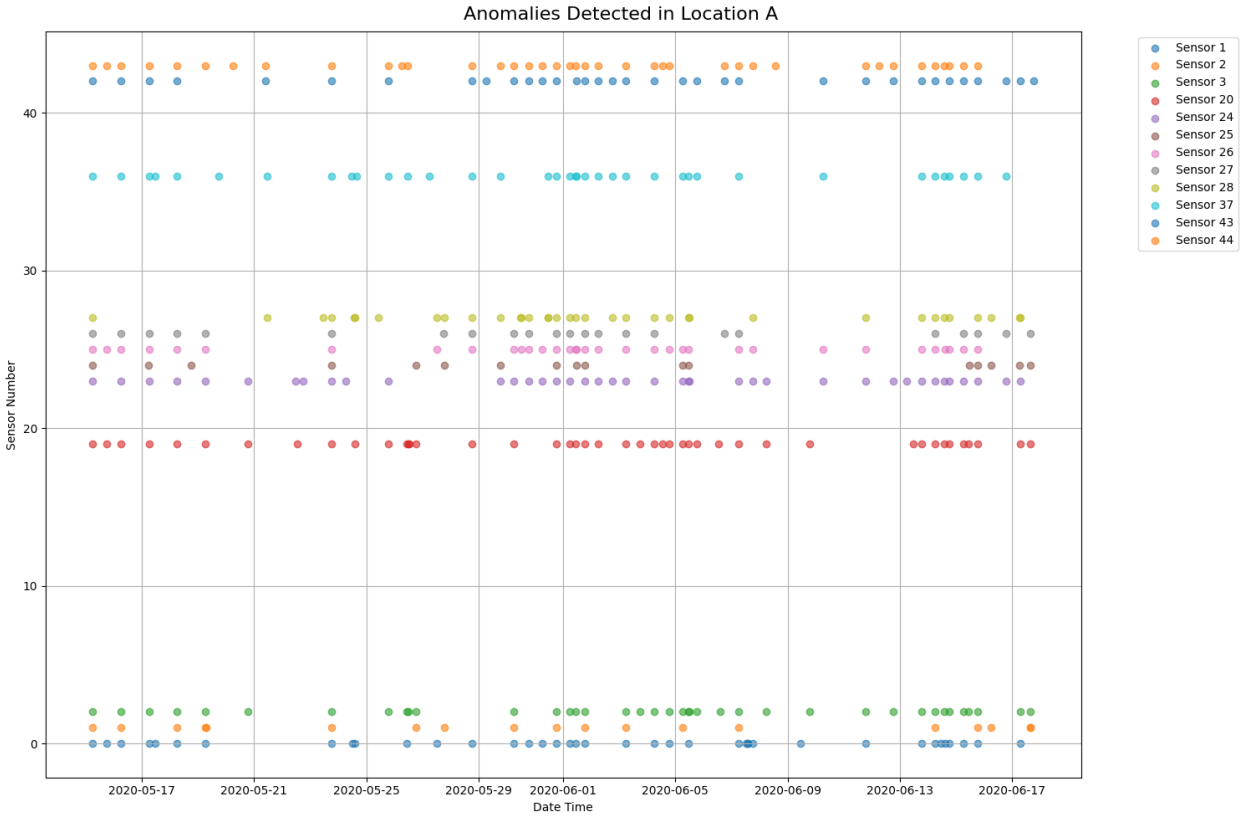
The CNN Autoencoder showed suboptimal results for this sensor. The reconstruction errors did not consistently differentiate between normal and anomalous data points for all the sensors, leading to a higher false negative rate. This pattern was observed across other sensors, prompting us to explore alternative models.

2. Linear Regression

Linear Regression served as a baseline for modeling the relationships between features and detecting anomalies.

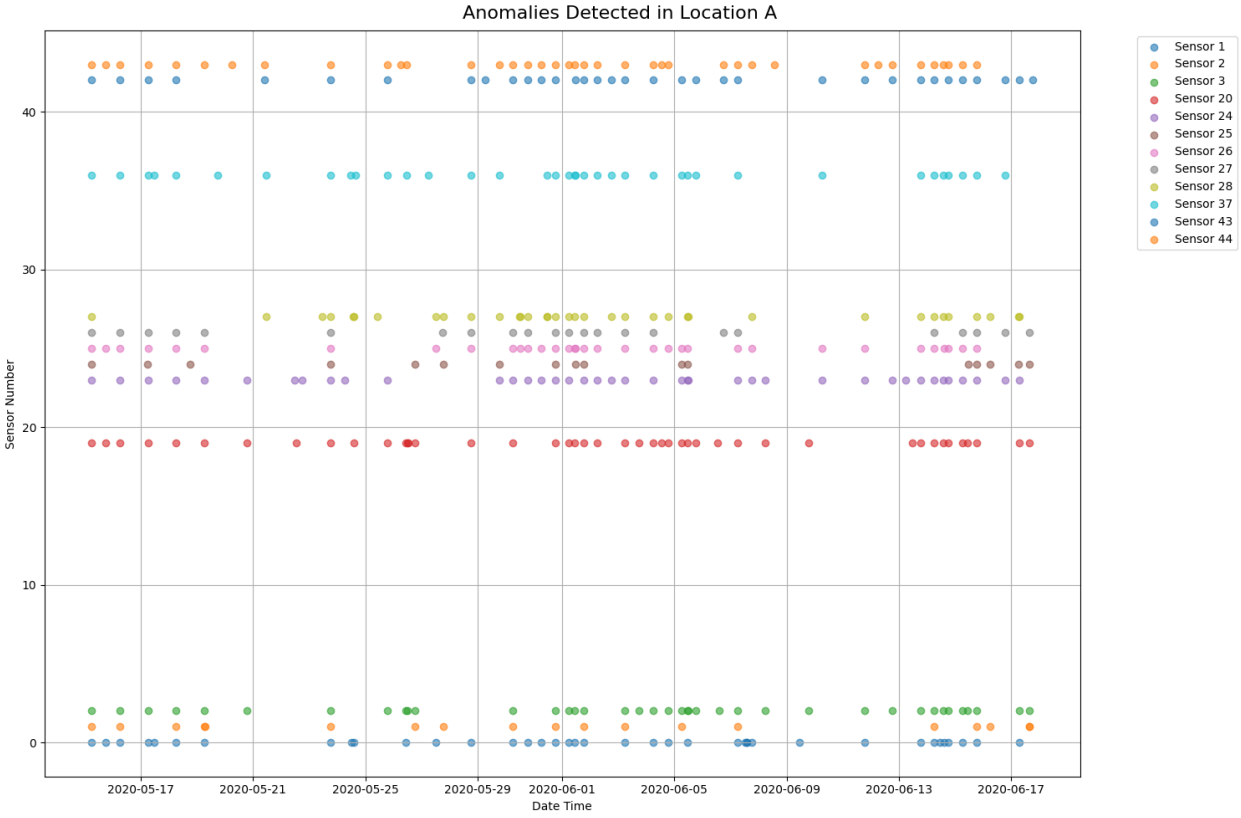
Location A:

- **Average Mean Squared Error (MSE)**: 986.0062
- **Average R^2 Score**: 0.9923



Location B:

- **Average Mean Squared Error (MSE)**: 15839.6510
- **Average R^2 Score**: 0.8606



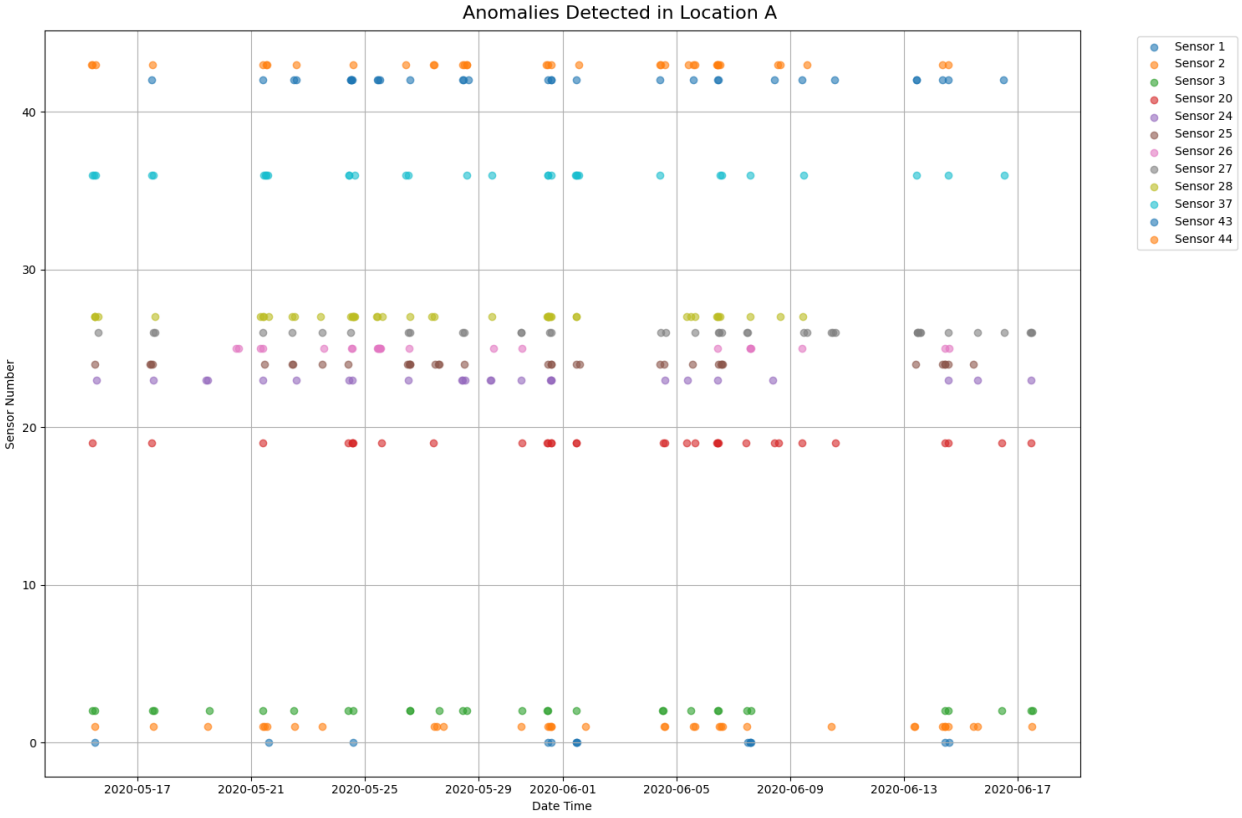
Linear Regression showed good performance for Location A but struggled with higher variance and errors at Location B, likely due to the non-linear relationships in the data.

3. Random Forest Regression

Random Forest Regression, a robust ensemble method, provided significant improvements in accuracy and error reduction.

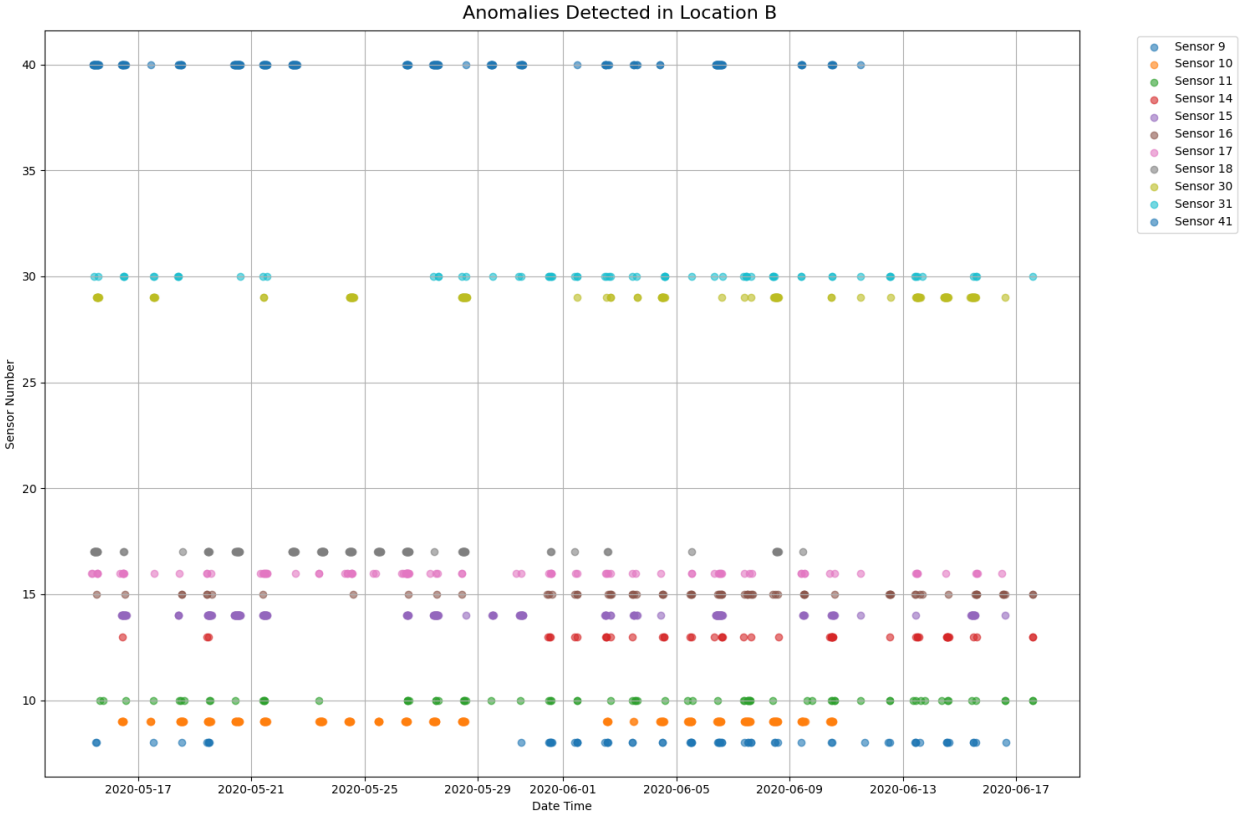
Location A:

- **Average Mean Squared Error (MSE)**: 417.4443
- **Average R^2 Score**: 0.9967



Location B:

- **Average Mean Squared Error (MSE)**: 317.8780
- **Average R^2 Score**: 0.9974



This model demonstrated strong performance across both locations, effectively capturing non-linear patterns and reducing prediction errors.

4. Gradient Boosting Regression

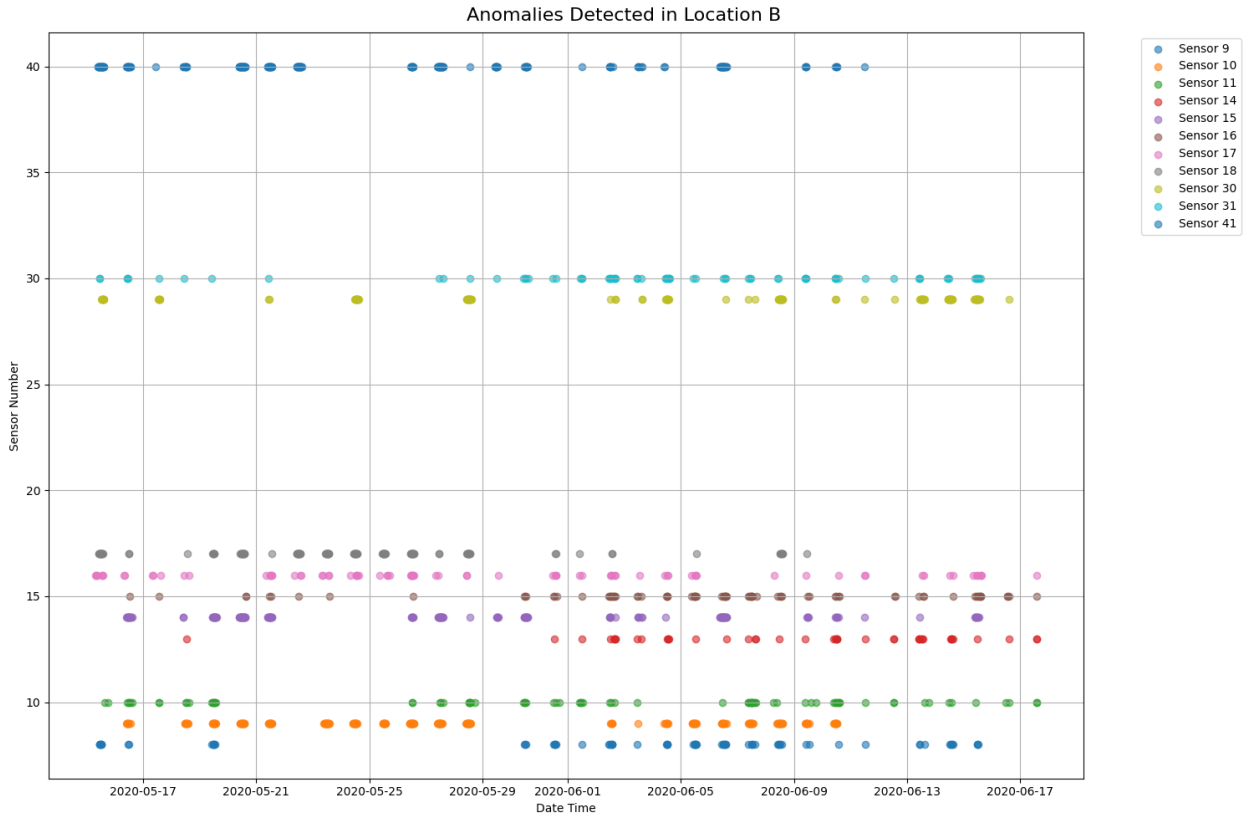
Gradient Boosting Regression further refined anomaly detection with even lower errors and higher R^2 scores, particularly for Location B.

Location A:

- **Average Mean Squared Error (MSE)**: 425.1359
- **Average R^2 Score**: 0.9966

Location B:

- **Average Mean Squared Error (MSE)**: 223.5140
- **Average R^2 Score**: 0.9982



Gradient Boosting proved to be the most effective model, particularly in scenarios with complex relationships and noise in the data.

Performance Summary: Gradient Boosting Regression emerged as the most reliable method for anomaly detection across sensors, particularly for Location B, where data complexity was higher.