Enhancements to RoLA: Optimizing Real-Time Anomaly Detection in Time-Series Data

Samuel Moses Orokpo

March 20, 2025

1 Introduction

This report documents the enhancements and modifications made to the RoLA (Real-time Online Lightweight Anomaly Detection) system to improve its accuracy, efficiency, and reliability in detecting anomalies in multivariate time-series data.

1.1 Objectives

- Merge multiple sensor datasets into a single structured dataset
- Preprocess the dataset to handle missing values, normalize data, and generate ground truth labels
- Optimize the RoLA framework to improve precision, recall, and inference efficiency

2 Merging Multiple Sensor Datasets

2.1 Source Datasets

The final dataset was created by merging the following individual sensor datasets:

- C3_Temperature_10-28_10-30.csv
- C3_Turbidity_10-28_10-30.csv
- Flow_Flow_10-28_10-30.csv
- Flow_Temperature_10-28_10-30.csv
- Optode_Concentration_10-28_10-30.csv
- Optode_Saturation_10-28_10-30.csv
- Optode_Temperature_10-28_10-30.csv
- SEB45_Conductivity_10-28_10-30.csv
- SEB45_Salinity_10-28_10-30.csv

2.2 Merging Strategy

These datasets were merged using their common timestamp field:

1. Convert all timestamp columns to a standard datetime format
2. Merge all datasets on the timestamp field using an inner join
3. Ensure uniformity in column names and alignment of records

3 Preprocessing Steps

3.1 Handling Missing Data

- Forward-fill (ffill) and backward-fill (bfill) were applied to propagate missing values.
- Linear interpolation was used for estimating missing values in numerical columns.

3.2 Feature Selection & Normalization

- All numerical sensor readings were kept for anomaly detection.
- Min-Max normalization was applied to scale all sensor readings between 0 and 1.

3.3 Generating Ground Truth Anomaly Labels

Since the dataset lacked labeled anomalies, a new column 'Anomaly_Label' was generated using statistical outlier detection:

1. Compute the Z-score for each numerical feature.
2. Mark a data point as an anomaly (1) if any feature exceeds 3 standard deviations.
3. Otherwise, mark the point as normal (0).

4 Key Modifications to RoLA

4.1 1. Increased Lookback Window for LSTM

Change: Adjusted '"LOOKBACK"' from 5 to 10 to allow the model to better capture long-term patterns.
Impact:

- Improved ability to detect complex anomalies over time.
- Helped reduce false positives caused by short-term fluctuations.

4.2 2. Stricter Dynamic Thresholding for Anomalies

Change: Adjusted the dynamic thresholding multiplier from 3 $\rightarrow$ 3.5:

```
threshold = mean + 3.5 * std
```

Impact:

- Reduced false positives from normal fluctuations.
- Improved precision in anomaly detection.

4.3 3. Simplified Correlation-Based Anomaly Confirmation

Change: Applied a lighter correlation analysis to reduce unnecessary complexity:

```
if abs(corr) > CONFIG["CORREL_THRESHOLD"]:
    confirmed_anomalies[i] = confirmed_anomalies[i] or anomalies[j]
```

Impact:

- Ensured anomalies are only flagged when correlated variables also show deviations.
- Reduced false positives without significantly affecting recall.

5 Performance Evaluation

Final Evaluation Results (n=4600 full dataset):

```
TP: 71, FP: 168, FN: 38
Precision: 0.297, Recall: 0.651, F1-score: 0.408
Avg Inference Time: 0.619s, Std Dev: 0.043s
```

Impact:

- Balanced recall (0.651) and precision (0.297).
- Reduced FP compared to earlier versions.
- Maintained inference time at 0.6s, making it real-time applicable.

6 Conclusion & Future Work

The enhancements made to RoLA have resulted in better anomaly detection performance by reducing false positives while maintaining real-time processing capabilities.

6.1 Next Steps for Further Optimization

- Hybrid Model (LSTM + Isolation Forest) for further FP reduction.
- Fine-tuning threshold dynamically based on rolling mean trends.
- Testing on a larger dataset to improve robustness.