Time Series Anomaly Detection for IoT Sensors

AI/ML Engineer (Fresher) – Assignment Summary

1. Problem Understanding

In manufacturing environments, IoT sensors are widely used to monitor equipment health in real time. These sensors generate continuous time-series data, where abnormal patterns may indicate equipment faults, degradation, or the need for maintenance.

The main challenge in such systems is that anomalies are rare and usually unlabeled. Because of this, traditional supervised learning approaches are often not feasible.

The objective of this assignment is to design an end-to-end anomaly detection solution that can identify unusual sensor behavior using unsupervised learning techniques, while clearly explaining the reasoning behind each design decision.

2. Dataset and Data Preparation

Dataset Choice

For this assignment, synthetic time-series data was generated to simulate realistic IoT sensor behavior. This approach was chosen because it allows controlled injection of anomalies and easier validation in the absence of labeled real-world data.

Normal Sensor Behavior

- Smooth sinusoidal pattern
- Small random noise to represent sensor variation

Injected Anomalies

- To reflect real industrial scenarios, the following anomalies were introduced
- Sudden spikes representing sensor glitches
- Sudden drops representing power or communication failures
- Gradual drift representing equipment wear over time
- Sudden reset after drift representing maintenance or recalibration

Data Quality Issues

To simulate real-world conditions:

- Missing values were intentionally introduced
- Missing values were handled using forward-fill, with backward-fill applied for leading missing values

Exploratory data analysis was performed using time-series plots to visually understand normal patterns and abnormal behavior.

3. Feature Engineering

Raw sensor values alone are often insufficient to capture contextual anomalies. To improve detection performance, additional features were engineered:

- Rolling mean to capture local trends
- Rolling standard deviation to capture local variability
- Difference between consecutive readings to detect abrupt changes
- Deviation from rolling mean to highlight abnormal deviations

These features help models understand how the sensor behaves relative to its recent history. All features were normalized using standard scaling to ensure consistent model behavior.

4. Anomaly Detection Approaches

Two different unsupervised approaches were implemented and compared.

4.1 Isolation Forest

Isolation Forest is an unsupervised anomaly detection algorithm that works by isolating rare observations through random partitioning.

Reason for selection:

- Does not require labeled data
- Efficient and relatively easy to interpret
- Suitable for detecting rare point anomalies

Observations:

- Successfully detected sudden spikes and drops
- Performed well on abrupt, point-based anomalies
- Had limited ability to detect gradual drift, since drift points are not rare individually

4.2 Autoencoder (Deep Learning Approach)

A lightweight dense autoencoder was implemented using TensorFlow. The model was designed to be CPU-friendly and suitable for a standard laptop environment.

Working principle:

- The autoencoder learns to reconstruct normal sensor behavior
- Anomalies result in higher reconstruction error
- A threshold based on the 95th percentile of reconstruction error was used to flag anomalies

Observations:

- Better at detecting gradual drift and contextual anomalies
- Successfully captured regime changes such as sensor reset
- Produced some false positives due to higher sensitivity

5. Model Evaluation and Validation

Since the dataset does not contain ground-truth labels, traditional supervised metrics such as accuracy, precision, and recall were not directly applicable.

Evaluation Strategy

Models were evaluated using:

- Visual inspection of anomaly detection plots
- Alignment with injected anomaly regions
- Comparison and overlap between different detection methods
- Domain-based reasoning

Model Comparison

- Isolation Forest performed well for sharp, isolated anomalies
- Autoencoder was more effective for detecting gradual drift and extended anomalous regions
- A trade-off was observed between sensitivity and false positives

This comparison highlights the importance of using multiple approaches for anomaly detection in real-world systems.

6. Key Findings and Business Insights

- Early detection of sensor drift can help identify equipment degradation before failure
- Sudden spikes and drops may indicate sensor faults or communication issues
- Combining statistical and deep learning approaches provides better coverage of different anomaly types
- Unsupervised models are practical for industrial environments where labeled anomalies are unavailable

From a business perspective, such a system can support predictive maintenance, reduce downtime, and improve operational reliability.

7. Limitations

- No labeled anomaly data for quantitative evaluation
- Synthetic data may not fully capture all real-world complexities
- Threshold selection for reconstruction error is heuristic
- Autoencoder sensitivity can lead to false positives

8. Future Improvements

- Apply the solution to real industrial IoT datasets
- Introduce labeled anomalies for quantitative evaluation
- Explore sequence-based models such as LSTM autoencoders
- Implement adaptive or dynamic anomaly thresholds
- Deploy the model as a real-time monitoring service

9. Conclusion

This project demonstrates a complete, end-to-end approach to time-series anomaly detection for IoT sensor data. It highlights the importance of data understanding, feature engineering, model selection, and careful interpretation of results in the absence of labeled data.

The assignment provided valuable hands-on experience in building practical anomaly detection systems suitable for real-world industrial use cases.