Project Proposal:
AI/ML IoT Sensor Data Validation Pipeline For Upstream Digitalisation

Muhammad Nur Arif Zanuri

23rd May 2025

CONTENT

INTRODUCTION

Problem Statement:

- Irregularities in newly installed IoT-sensor data may lead to inaccurate production reports. This could result in incorrect tax declarations and expose thee company to potential regulatory fines.

Objective:

- To develop a AI/ML validation pipeline verifying the integrity of new IoT-sensors with the following requirements:
  1. **Accuracy**: Detect deviations/anomalies in sensor behaviour based on historical patterns.
  2. **Timeliness**: Trigger alerts as instant as irregularities detected.
  3. **Actionability**: Classify sensors according to risk level to guide proactive maintenance.

Scope:the company

ss near-real time and historical data from legacy and newly installed sensors.

- Apply anomaly detection and early degradation tracking using ML models.
- Visualize outputs and status per sensor for operational decision-making
- Pipeline need to be modular and can easily scaled across multiple sites.

DATA UNDERSTANDING AND EXPLORATORY DATA ANALYSIS STRATEGY

Data Gathering:

- Identify and gather validated old IoT-sensor data with unique id for each sensor.*
- Identify and gather data from new IoT-sensor with unique id for each sensor.*

Feature Profiling:

- Identify key raw features available from sensors:
  - *In our demo, we will use the following features: Temperature, Vibration, Wind Speed, Humidity*
- Review data types, value ranges, and missing/null values
- Visualize distribution of data using histograms and boxplots to understand trends and variances.

Time Series Behaviour Analysis

- Plot feature trends over time for every sensor and identify irregular patterns such as sudden spikes or drops, gradual increase or drift, plateauing or flatlines

Infer-feature Relationship

- Study the relationships between features using scatterplots and correlation metrices. This is to help in identifying which features can be use to explain or predict the behaviour of others.

* Data used in our demo is a mock data with various degradation scenarios applied to new sensors.

DATA UNDERSTANDING AND EXPLORATORY DATA ANALYSIS STRATEGY (CONT'D)

Identify Feature Trends Over Time:

- The feature trends needs to be computed to:
  - Identify change patterns for any hidden degradation
  - Identify early the potential gradual declines prior crossing the failure thresholds.
  - Enables better learning of machine learning models with additions of behaviour-rich data.
- The features to be computed are:
  - Rolling Mean (5 min) – To captures the local trend within 5mins window.
  - Rolling Standard Deviation – Detect stability or volatility in readings within 5 mins window.
  - Delta (1st Difference) – Measures rate of change to identify early indicators of drift.

* Data used in our demo is a mock data with various degradation scenarios applied to new sensors.

DATA PREPARATION

1. Replace null/missing values with median imputation.
2. Normalize the data with MinMaxScaler
3. Add trend aware features.
4. Perform feature selection by:
    1. Drop features with low variance
    2. Combine highly correlated features to avoid redundancy
    3. Identify features known relate to degradation with input from SME
    4. Use ML model based feature selection such as feature importance scores (XGBoost), Recursive Feature Elimination (RFE), and SHAP values

ML MODEL STRATEGY

The following ML Model are suggested to be use to validate the accuracy of IoT sensor data:

1) **XGBoost Residual Models (Trend-Based Model):**
   - Train models will be created for each features and residual between predicted and actual values will be calculated
   - Enable early inconsistencies detection by highlighting gradual drift.

2) **Autoencoder Neural Networks (Trend-Based Model):**
   - Learned and rebuild a behaviour of normal sensor and flags reading with high reconstruction error.
   - Enable detection of non-obvious sensor faults.

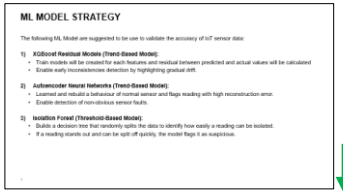
3) **Isolation Forest (Threshold-Based Model):**
   - Builds a decision tree that randomly splits the data to identify how easily a reading can be isolated.
   - If a reading stands out and can be split off quickly, the model flags it as suspicious.

SENSOR ALERT STRATEGY

The decision to determine sensor status based on application of all 3 ML models in the previous slides are as below:

Sensor Status	Trigger Condition
Faulty	• Flagged by Isolation Forest OR • Both XGBoost and Autoencoder show anomaly
Warning	• Either XGBoost or Autoencoder detects early sign of degradation
Normal	• No anomalies detected by any model

HIGH LEVEL ARCHITECTURE DIAGRAM


ML MODEL STRATEGY

The following ML Model are suggested to be use to validate the accuracy of IoT sensor data:

1) XGBoost Residual Models (Trend-Based Model):
- Train models will be created for each features and residual between predicted and actual values will be calculated
- Enable early inconsistencies detection by highlighting gradual drift.

2) Autoencoder Neural Networks (Trend-Based Model):
- Learned and rebuild a behaviour of normal sensor and flags reading with high reconstruction error.
- Enable detection of non-obvious sensor faults.

3) Isolation Forest (Threshold-Based Model):
- Builds a decision tree that randomly splits the data to identify how easily a reading can be isolated.
- If a reading stands out and can be split off quickly, the model flags it as suspicious.

Model will be re-trained with new sensor data after every 5 mins

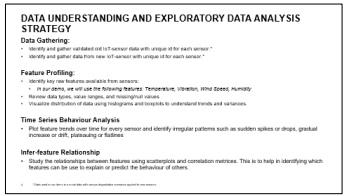
Old Sensor Database

Stream every 30 seconds

New Sensor

Sensor_1:
Working normally
Sensor 2:
Gradual Degradation With Time
Sensor 3:
Obvious faulty sensor

Exploratory Data Analysis and Data Preparation

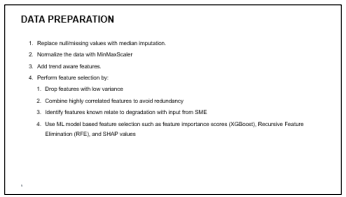

DATA UNDERSTANDING AND EXPLORATORY DATA ANALYSIS STRATEGY

Data Gathering:
- Identify and gather validated old IoT-sensor data with unique id for each sensor.*
- Identify and gather data from new IoT-sensor with unique id for each sensor.*

Feature Profiling:
- Identify key raw features available from sensors:
  - *In our demo, we will use the following features: Temperature, Vibration, Wind Speed, Humidity*
- Review data types, value ranges, and missing/null values.
- Visualize distribution of data using histograms and boxplots to understand trends and variances.

Time Series Behaviour Analysis
- Plot feature trends over time for every sensor and identify irregular patterns such as sudden spikes or drops, gradual increase or drift, plateauing or flatlines.

Infer-feature Relationship
- Study the relationships between features using scatterplots and correlation metrics. This is to help in identifying which features can be use to explain or predict the behaviour of others.

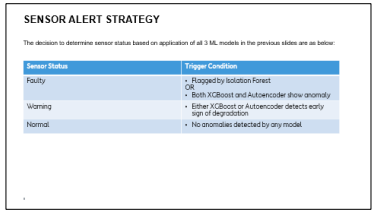

DATA PREPARATION

1. Replace null/missing values with median imputation.
2. Normalize the data with MinMaxScaler
3. Add trend aware features.
4. Perform feature selection by:
   1. Drop features with low variance
   2. Combine highly correlated features to avoid redundancy
   3. Identify features known relate to degradation with input from SME
   4. Use ML model based feature selection such as feature importance scores (XGBoost), Recursive Feature Elimination (RFE), and SHAP values

Model Training

1) XGBoost
2) Autoencoder Neural Networks
3) Isolation Forest

Alert Predictions


SENSOR ALERT STRATEGY

The decision to determine sensor status based on application of all 3 ML models in the previous slides are as below:

Sensor Status	Trigger Condition
Faulty	• Flagged by Isolation Forest OR • Both XGBoost and Autoencoder show anomaly
Warning	• Either XGBoost or Autoencoder detects early sign of degradation
Normal	• No anomalies detected by any model.

New Sensor Stream for 5 minutes

yes

no

Power BI

Alert To Site Engineer To Check the Sensor

Data Ingestion Layers

Data Transformation Layers

ML Processing Layer

Visualization & Decision Layers

CONCLUSION

- The proposed validation system are designed to process a IoT real time sensor data via multi model ML ensemble where 2 of those are trend based model (XGBoost and Autoencoder Neural Network) and 1 threshold based model which is Isolation Forest.
- The demonstration with mock-up data shows the utilization of the proposed validation pipeline enables identification of critical faults to the sensor and provide warning as the sensor showing signs of degradation.
- The solution is easily scalable to ingest sensor data from multiple sites and able to prevent data quality issues that could impact the company legally and financially.

• Proposed Enhancements

- Integrate the validation system to the company main operation alerting platforms enabling real-time visibility and faster response from site engineers.
- Expands the scope of the system with predictive modelling on the sensors expected end of life (EOL) estimation via recurrent neural network (RNN) technique such as Long Short Term Memory to forecast remaining sensor lifespan.
- Introduce buffering middleware layer such as Apache Kafka to decouple data ingestion from processing. This avoid overloading transformation and ML servers while ensuring new sensor data is still captured in the event of temporary server failures.