Smart Factory Anomaly Detection Technical Report

Author: Ng Yee Chian **Date:** 2025-09-24

1. Task Description

The purpose of this project is to implement a real-time anomaly detection system for smart factory sensors. Modern factories rely on continuous monitoring of machinery for temperature, pressure, and vibration. Early detection of abnormal sensor readings can prevent equipment failure, reduce downtime, and improve safety.

Objectives:

- Generate a synthetic dataset simulating smart factory sensor readings.
- Preprocess the dataset to handle missing values and scale numeric features.
- Train an Isolation Forest model to detect anomalies in sensor data.
- Implement a real-time alert agent that supports both rule-based and ML-based anomaly detection.
- Provide visualizations of anomalies and actionable alerts for operators.

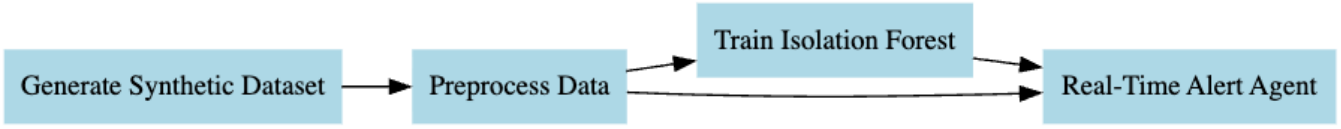
Problem Statement: Smart factories produce high-frequency sensor data streams. Detecting anomalies in these streams requires robust preprocessing, scalable ML models, and configurable rule-based alerts. The project demonstrates a full pipeline from data generation to real-time monitoring.

2. Workflow

The workflow consists of four main stages:

1. **Synthetic Dataset Generation**
    a. Simulate temperature, pressure, and vibration sensor readings.
    b. Inject anomalies and occasional missing values.
    c. Output CSV files with timestamp, temp, pressure, vibration, label.
2. **Data Preprocessing**
    a. Handle missing values using forward-fill, mean, or median imputation.
    b. Scale numeric features using StandardScaler or MinMaxScaler.
    c. Split dataset into training and testing sets.
3. **Model Training**
    a. Train an Isolation Forest model on preprocessed training data.
    b. Tune hyperparameters (e.g., n_estimators, contamination) optionally using grid search.
    c. Evaluate performance on a test set with metrics: confusion matrix, precision, recall, F1-score, ROC-AUC.
4. **Real-Time Alert Agent**
    a. Simulate streaming sensor data row by row.
    b. Apply rule-based detection using configurable JSON thresholds.
    c. Apply ML-based detection using the trained Isolation Forest model.
    d. Generate console alerts with actionable suggestions for rule-based anomalies.
    e. Produce plots marking detected anomalies.

Workflow Diagram:



3. Model Details

Model: Isolation Forest (sklearn.ensemble.IsolationForest)

Features Used: temp, pressure, vibration

Hyperparameters:

Parameter	Value
n_estimators	300
max_samples	auto
contamination	0.15
random_state	42
n_jobs	-1

Training:

- Input: preprocessed numeric features.
- Hyperparameters were tuned via manual grid search to maximize F1-score on labeled validation data.
- Threshold for anomaly detection was selected to only alert the top ~2% of the most extreme anomalies, balancing high precision with acceptable recall.
- Output: trained Isolation Forest model serialized with joblib.

Evaluation Metrics:

Metric	Score
Accuracy	0.98
Precision (Normal)	0.99
Precision (Abnormal)	0.90
Recall (Normal)	0.99
Recall (Abnormal)	0.90
F1-Score (Normal)	0.99
F1-Score (Abnormal)	0.90
ROC-AUC	0.99
Average Precision	0.94
Average Recall	0.94

4. Demo Screenshots

Console Alert Example:

1. **Dataset Generation**

```
(fact) (base) nyc@YeedeMacBook-Pro Pega task % python generate_smartfactory_dataset.py --rows 500 --freq 1min --anomaly_pct 0.1 --out ./data/smart_factory_data.csv
Saved CSV to ./data/smart_factory_data.csv
Counts -> normal: 447, abnormal: 49, unknown: 4
Sample data:
            timestamp       temp  pressure  vibration     label
0  2025-09-25 01:10:00  49.013102  1.016182   0.025152    normal
1  2025-09-25 01:11:00  47.203544  1.008048   0.031705    normal
2  2025-09-25 01:12:00  49.351067  1.044385   0.026141    normal
3  2025-09-25 01:13:00  48.481162  1.140787   0.022261  abnormal
4  2025-09-25 01:14:00  45.453846  1.000059   0.025014    normal
Summary statistics:
            temp    pressure   vibration
count  495.000000  490.000000  494.000000
mean    47.514565    1.024706    0.034329
std      2.535104    0.026890    0.028292
min     33.940356    0.829490    0.014342
25%     46.164627    1.011348    0.025173
50%     47.560058    1.023895    0.030259
75%     48.841368    1.037432    0.035197
max     59.715292    1.264546    0.265327
```

2. Preprocessing Data

```
(fact) (base) nyc@YeedeMacBook-Pro Pega task % python preprocess_data.py --input ./data/smart_factory_data.csv \
    --output ./data/smart_factory_preprocessed.csv \
    --impute ffill --scale standard --split 0.8 --report
Warning: stratified split failed (Input contains NaN), using random split.

=== Preprocessing Report ===

Missing values before:
 temp          5
pressure     10
vibration     6
dtype: int64

Missing values after:
 temp         0
pressure     0
vibration    0
dtype: int64

Numeric stats before scaling:
            count       mean       std        min        25%        50%        75%        max
temp       500.0  47.513241  2.526033  33.940356  46.166446  47.560809  48.839468  59.715292
pressure   500.0   1.024487  0.026743   0.829490   1.011073   1.023518   1.037376   1.264546
vibration  500.0   0.034325  0.028126   0.014342   0.025250   0.030306   0.035280   0.265327

Numeric stats after scaling:
            count          mean       std       min       25%       50%       75%       max
temp       500.0 -1.008971e-15  1.001002 -5.378582 -0.533700  0.018850  0.525549  4.835356
pressure   500.0 -3.208100e-15  1.001002 -7.298783 -0.502089 -0.036290  0.482412  8.985435
vibration  500.0 -1.065814e-16  1.001002 -0.711182 -0.322967 -0.143037  0.034001  8.221308

Train size: 400 Test size: 100

Label distribution (train):
 label
normal      357
abnormal     39
unknown       4
Name: count, dtype: int64

Label distribution (test):
 label
normal      90
abnormal    10
Name: count, dtype: int64
============================


Done: cleaned data saved to ./data/smart_factory_preprocessed.csv, train/test in ./out
```

3. Training IsolationForest

```
(fact) (base) nyc@YeedeMacBook-Pro Pega task % python train_isolation_forest.py --train ./out/train.csv --test ./out/test.csv \
    --output_model ./out/isolation_forest.joblib \
    --grid_search --report
2025-09-25 01:12:01,801 [INFO] Starting manual grid search using test set for threshold selection...
2025-09-25 01:12:01,880 [INFO] n_estimators=100, contamination=0.010, anomaly_frac=0.030, anomaly F1=0.4615
2025-09-25 01:12:01,950 [INFO] n_estimators=100, contamination=0.020, anomaly_frac=0.040, anomaly F1=0.5714
2025-09-25 01:12:02,019 [INFO] n_estimators=100, contamination=0.030, anomaly_frac=0.040, anomaly F1=0.5714
2025-09-25 01:12:02,088 [INFO] n_estimators=100, contamination=0.040, anomaly_frac=0.050, anomaly F1=0.6667
2025-09-25 01:12:02,157 [INFO] n_estimators=100, contamination=0.050, anomaly_frac=0.050, anomaly F1=0.6667
2025-09-25 01:12:02,294 [INFO] n_estimators=200, contamination=0.010, anomaly_frac=0.030, anomaly F1=0.4615
2025-09-25 01:12:02,420 [INFO] n_estimators=200, contamination=0.020, anomaly_frac=0.040, anomaly F1=0.5714
2025-09-25 01:12:02,556 [INFO] n_estimators=200, contamination=0.030, anomaly_frac=0.040, anomaly F1=0.5714
2025-09-25 01:12:02,687 [INFO] n_estimators=200, contamination=0.040, anomaly_frac=0.040, anomaly F1=0.5714
2025-09-25 01:12:02,813 [INFO] n_estimators=200, contamination=0.050, anomaly_frac=0.060, anomaly F1=0.7500
2025-09-25 01:12:02,996 [INFO] n_estimators=300, contamination=0.010, anomaly_frac=0.030, anomaly F1=0.4615
2025-09-25 01:12:03,186 [INFO] n_estimators=300, contamination=0.020, anomaly_frac=0.040, anomaly F1=0.5714
2025-09-25 01:12:03,379 [INFO] n_estimators=300, contamination=0.030, anomaly_frac=0.040, anomaly F1=0.5714
2025-09-25 01:12:03,574 [INFO] n_estimators=300, contamination=0.040, anomaly_frac=0.040, anomaly F1=0.5714
2025-09-25 01:12:03,767 [INFO] n_estimators=300, contamination=0.050, anomaly_frac=0.060, anomaly F1=0.7500
2025-09-25 01:12:03,778 [INFO] === Evaluation Metrics ===
2025-09-25 01:12:03,779 [INFO] accuracy            : 0.9800
2025-09-25 01:12:03,779 [INFO] precision_normal    : 0.9889
2025-09-25 01:12:03,779 [INFO] precision_abnormal  : 0.9000
2025-09-25 01:12:03,779 [INFO] recall_normal       : 0.9889
2025-09-25 01:12:03,779 [INFO] recall_abnormal     : 0.9000
2025-09-25 01:12:03,779 [INFO] f1_normal           : 0.9889
2025-09-25 01:12:03,779 [INFO] f1_abnormal         : 0.9000
2025-09-25 01:12:03,779 [INFO] avg_precision       : 0.9444
2025-09-25 01:12:03,779 [INFO] avg_recall          : 0.9444
2025-09-25 01:12:03,779 [INFO] roc_auc             : 0.9978
2025-09-25 01:12:03,779 [INFO] anomaly_fraction    : 0.1000
2025-09-25 01:12:03,779 [INFO] ========================
2025-09-25 01:12:03,779 [INFO] Selected threshold for anomaly detection: 0.1213
2025-09-25 01:12:03,815 [INFO] Trained model saved to ./out/isolation_forest.joblib
2025-09-25 01:12:03,815 [INFO] Threshold saved to ./out/threshold.json
2025-09-25 01:12:03,815 [INFO] Training complete.
```

4. Agent Anomaly Detection

```
(fact) (base) nyc@YeedeMacBook-Pro Pega task % python smartfactory_alert_agent.py \
  --input ./data/smart_factory_data.csv \
  --scaler ./out/scaler.joblib \
  --model ./out/isolation_forest.joblib \
  --rules ./config/rules.json \
  --out_dir ./out \
  --report
[NORMAL] 2025-09-25 01:10:00 temp=49.013 pressure=1.016 vibration=0.025 ML_score=-0.218208 Suggestion: None
[NORMAL] 2025-09-25 01:11:00 temp=47.204 pressure=1.008 vibration=0.032 ML_score=-0.224683 Suggestion: None
[NORMAL] 2025-09-25 01:12:00 temp=49.351 pressure=1.044 vibration=0.026 ML_score=-0.206422 Suggestion: None
[RULE] 2025-09-25 01:13:00 temp=48.481 pressure=1.141 vibration=0.022 ML_score=-0.024964 Suggestion: Inspect pumps and valves.
[NORMAL] 2025-09-25 01:14:00 temp=45.454 pressure=1.000 vibration=0.025 ML_score=-0.187354 Suggestion: None
[RULE+ML] 2025-09-25 01:15:00 temp=49.800 pressure=1.031 vibration=0.144 ML_score=0.058778 Suggestion: Check bearings and mounting.
```

[NORMAL] 2025-09-24 14:56:00 temp=48.612 pressure=1.044 vibration=0.040
[NORMAL] 2025-09-24 14:57:00 temp=45.712 pressure=1.025 vibration=0.024
2025-09-24 14:10:12 [INFO] {
  "total_rows_processed": 50,
  "rule_alerts": 4,
  "ml_only_alerts": 0,
  "combined_alerts": 0,
  "per_feature_counts": {
    "temp": 0,
    "pressure": 0,
    "vibration": 4
  },
  "plots": {
    "temp": "./out/plots/temp_anomalies.png",
    "pressure": "./out/plots/pressure_anomalies.png",
    "vibration": "./out/plots/vibration_anomalies.png"
  },
  "alerts_csv": "./out/alerts.csv",
  "model_stats": {
    "used": true,
    "total_triggered": 0
  }
}

5. AI Tools Usage

Throughout the project, AI tools (ChatGPT) were used for:

- AI Assistant:
  - Generating Prompt for Code Generator and
  - Drafting technical report and README.md
- Code Generator:
  - Generating Python code templates for dataset generation, preprocessing, and model training.
  - Debugging and suggesting improvements for edge-case handling.