

Smart Factory Anomaly Detection

AI-Powered Industrial Sensor Analysis Pipeline

Assignment Report & Results

By Ng Yee Chian

Project Overview

Objective

Build and evaluate a comprehensive anomaly detection pipeline for Smart Factory sensor data to identify equipment failures before they occur

Methodology

Leverage AI-powered tools for rapid development, from dataset generation to model deployment and documentation

Deliverables

- Generated dataset
- Isolation Forest model
- Anomaly detection agent
- Complete documentation

AI Development Toolkit



AI Assistant (ChatGPT)

Strategic guidance for workflow design, prompt engineering, and technical research. Provided insights into anomaly detection algorithms and Isolation Forest methodology.



Code Generator (ChatGPT)

Automated creation of Python scripts for dataset generation, preprocessing pipelines, model training, and evaluation frameworks. Significantly reduced development time.



Gamma Platform

Professional presentation generation with sophisticated layouts and visual storytelling capabilities for technical reporting and stakeholder communication.

Example Use Case of AI tools (Prompt Engineering)

User: I want you to be my assistant to the assistant, give me prompts that helps a LLM assistant to complete task that I specify.

.
.
.

ChatGPT: If you want, next I can produce ready-to-send prompts for an LLM assistant to: (a) generate the dataset script, (b) implement the anomaly detector (rule-based + simple ML baseline), (c) create README and report sections, and (d) prepare demo screenshots and CLI outputs.

User: Yes, give me prompt (a)

ChatGPT:

You are a helpful Python developer. Produce a single, well-documented, runnable Python script named `generate_smartfactory_dataset.py` that creates a synthetic CSV dataset for the Smart Factory Alert Agent (Task 3). Follow these exact requirements:

1) CSV format & header

- Header columns (in this order): `timestamp,temp,pressure,vibration,label`
- Timestamp format: `YYYY-MM-DD HH:MM:SS` (24h)`
- Save the CSV default path: `./data/smartfactory_dataset.csv``

.
.
.

Example Use case of AI tools (Code Generating, Debugging)

User: (source code)

Give me prompt to ask you to write the whole code again according above, with modifying according to below 1) Give average precision and recall 2) Save metrics to a file

ChatGPT:

Prompt:

Rewrite the entire `train_isolation_forest.py` script with the following specifications:

Keep all existing functionality (loading data, preprocessing, Isolation Forest training, grid search, threshold selection, saving model).

Modify the evaluation section to additionally compute average precision and average recall across both classes.

Save all evaluation metrics (accuracy, precision/recall/F1 per class, average precision/recall, ROC-AUC, anomaly fraction) to a JSON file in the output directory.

Ensure logging still prints all metrics to console in a clear tabular format.

Maintain compatibility with command-line arguments (`--train`, `--test`, `--scaler`, `--output_model`, `--grid_search`, `--out_dir`, `--report`).

Example Use case of AI tools (Presentation Generation)

User: Create content for gammas to generate. Presentation must include below: introduction to the AI tools used: 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. a complete report of the assignment results and all relevant screenshots or source code on Github

ChatGPT:

Slide 1 – Title

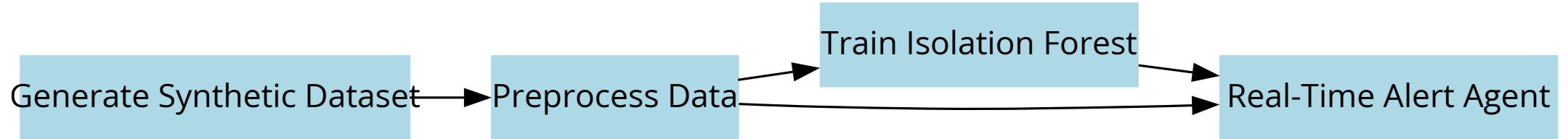
Smart Factory Anomaly Detection with AI-Powered Tools

Assignment Report & Results

By [Your Name]

·
·
·

Pipeline Architecture




Synthetic Dataset Generation




Dataset size: 500 rows of sensor data



Data Frequency: 1-minute intervals



Anomaly percentage: Approximately 10% of the data contains anomalies



Features: Temperature, pressure, and vibration measurements

Data Preprocessing Pipeline



Missing Value Strategy

Utilize the **forward fill (ffill)** method to efficiently handle and impute any missing sensor readings, ensuring data continuity.



Feature Scaling

Apply **standard scaling** to normalize the temperature, pressure, and vibration measurements, bringing them to a common scale.



Data Split

Perform an **80/20 train-test split** of the dataset to facilitate robust model training and validation.

Model Configuration

Algorithm: Isolation Forest

Unsupervised learning approach using `sklearn.ensemble.IsolationForest` for anomaly detection.

Hyperparameters are tuned with grid search to maximize f1-score.

Key Features: Temperature, pressure, vibration measurements from industrial equipment sensors.

Parameter	Value
n_estimators	200
max_samples	auto
contamination	0.05
random_state	42
n_jobs	-1

Performance Metrics

0.98

Overall Accuracy

0.94

Average Precision

0.94

Average Recall

0.99

ROC_AUC

0.9

Abnormal Precision

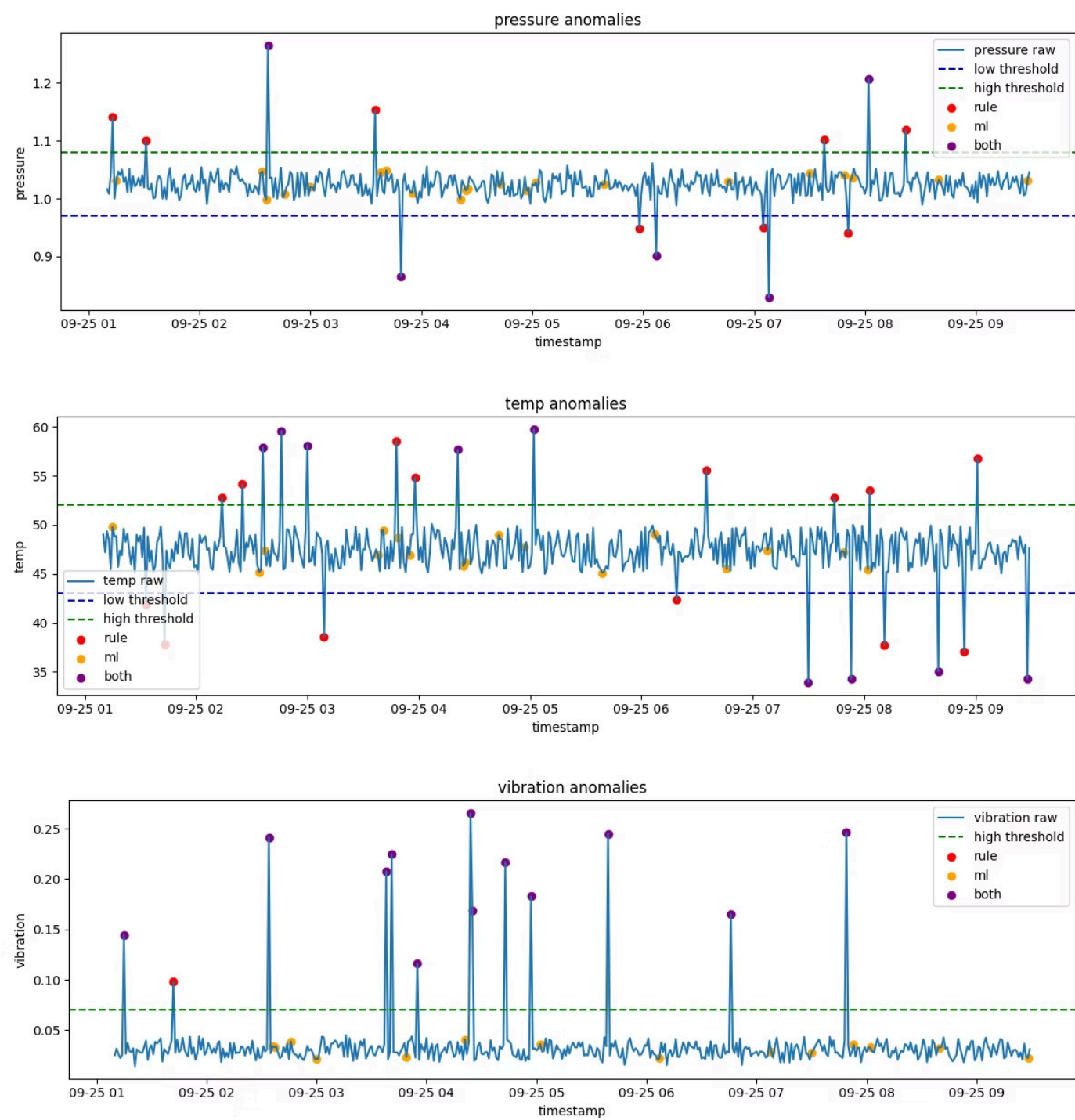
High confidence in anomaly
detection

0.9

Abnormal Recall

Getting all

Agent Detection



Future Work

- Time-series models to better detection of gradual drifts and contextual anomalies
- Compare with more models.
- Build ensemble detector by aggregating votes.
- Explainability of anomalies using SHAP/LIME

Repository Structure

Source Code

Complete Python implementation including dataset generation, preprocessing pipelines, model training, and anomaly detection agent

Data

Generated datasets with realistic sensor patterns, anomaly labels, and validation sets for comprehensive testing

Model Outputs

Trained models, performance metrics, evaluation reports, and visualization assets for stakeholder review

Documentation

Comprehensive README, workflow diagrams, technical specifications, and deployment guidance

GitHub Repository: [[GitHub - Nameyc99/Factory_Agent](#)]