SEMICONDUCTOR SENSOR ANOMALY DETECTION PROJECT REPORT

1. **PROJECT OBJECTIVE**

• Detect anomalous runs in semiconductor manufacturing sensor data using both unsupervised and supervised techniques

• Achieve high recall (≥ 0.80) and precision (≥ 0.50) for defect detection

• Deliver a reproducible end-to-end pipeline from data ingestion through model persistence

2. **DATA OVERVIEW**

• Raw dataset: 50 000 timestamped readings from 100 synthetic sensors

• Injected anomalies at approximately 5% of the runs (label = –1)

• Missing values: 8–9% per sensor, uniformly distributed

• Metadata: sensor type (temperature, pressure, vibration), units, expected ranges

DATA CLEANING AND IMPUTATION

• Dropped any feature with > 50% missing values (none in this dataset)

• Imputed remaining missing values:

– Raw sensor readings: fill with per-sensor median

– Rolling-window statistics: fill NaN with zero after computing forward-filled window

• Resulting clean dataset saved as defects_imputed.csv

3. **FEATURE ENGINEERING**

• Rolling-window features (window size = 5): mean, standard deviation, 10th and 90th percentiles.

• First-difference features (lag = 1) to capture abrupt changes

• Total features expanded from 100 raw sensors to over 600 features

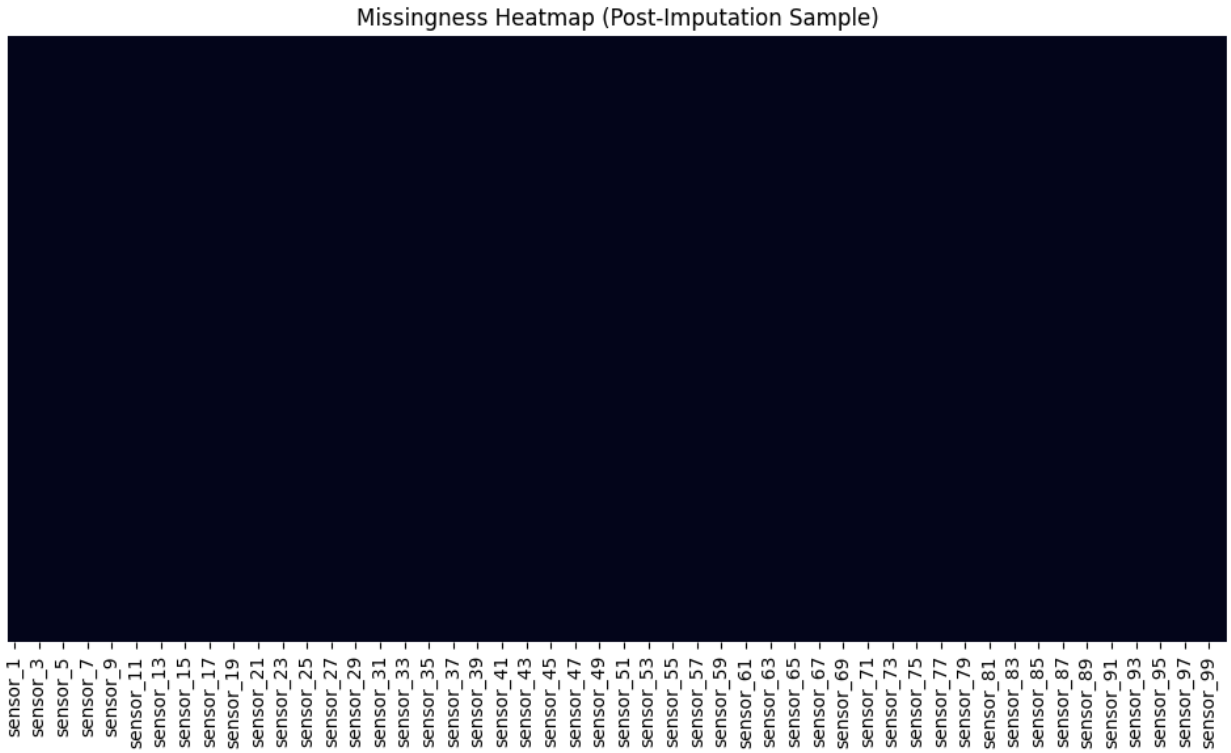
• Optimal window size determined by sweep (3, 5, 10): window = 5 gave the highest precision (0.307) and recall (0.634)

4. **UNSUPERVISED BASELINE (IsolationForest)**

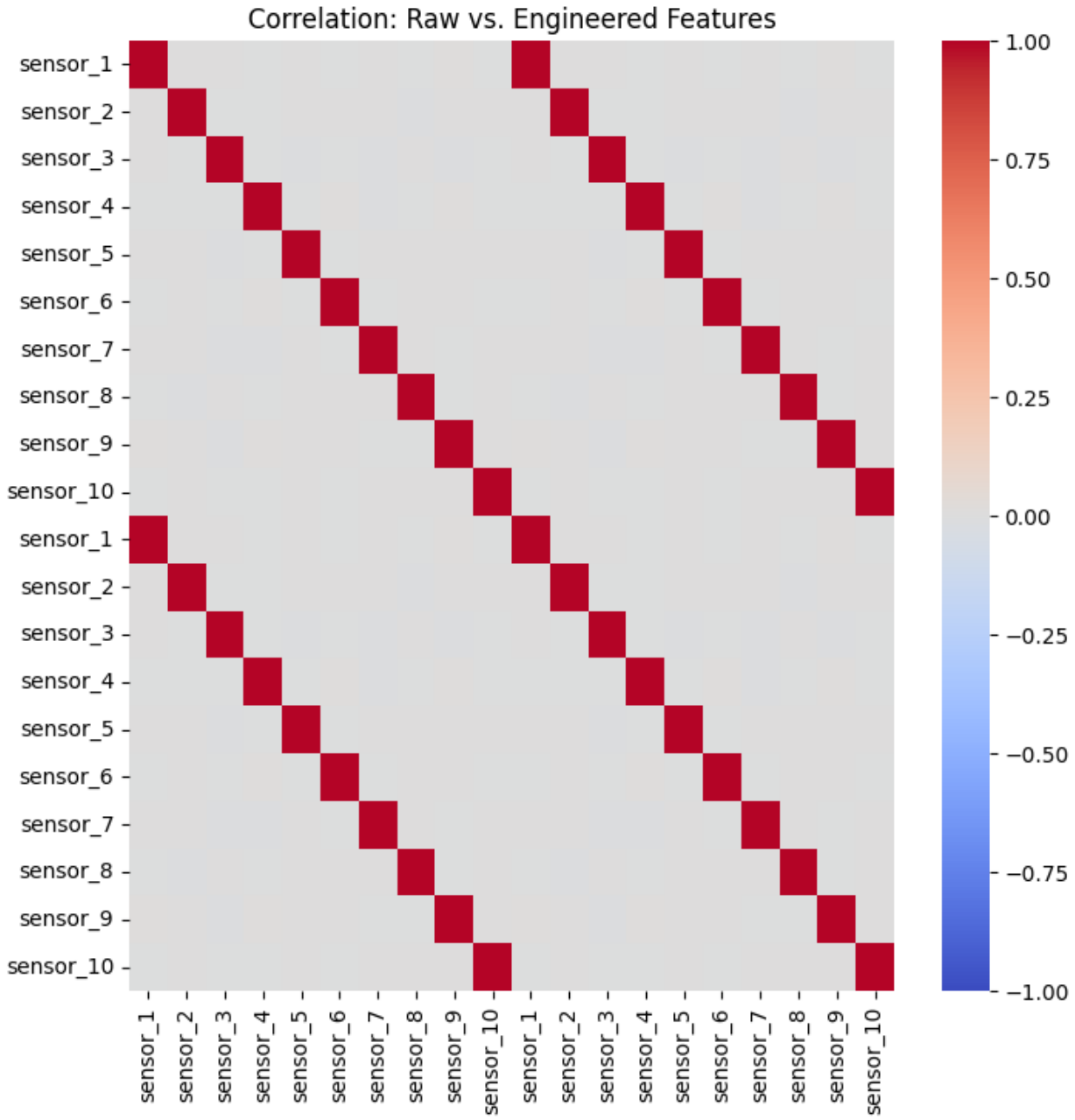
- Contamination sweep (0.01, 0.03, 0.05, 0.10)
  – Best balance at 5%: Precision and Recall both ≈ 0.050
  – Lower contamination (1%) raised precision to 0.061, but recall dropped to 0.012
- Scaling and PCA (95% variance) did not improve performance (Precision/Recall remained ≈ 0.05)

5. **VISUAL ANALYSIS**

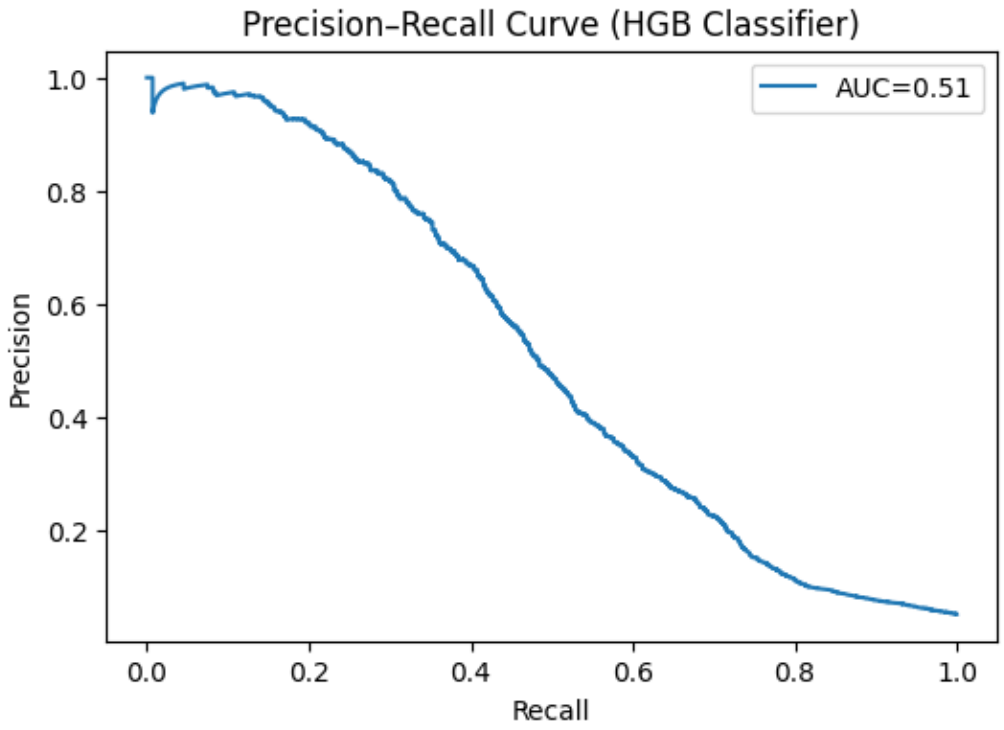
- Missingness Heatmap: confirmed no remaining NaNs after imputation



• Feature Correlation Matrix: strong correlation between each raw sensor and its rolling mean; low cross-sensor redundancy



• Precision–Recall Curve for final supervised model: AUC ≈ 0.51, indicating modest separation above random



6. **SUPERVISED MODELING (HistGradientBoostingClassifier)**

• Trained on enhanced feature set with class labels

• Default threshold (0.5) yielded Precision = 1.00, Recall ≈ 0.0005 (overly conservative)

• Threshold tuning sweep identified optimal probability cutoff at 0.06: Precision = 0.264, Recall = 0.496

• Hyperparameter grid search (learning_rate ∈ {0.05,0.1,0.2}, max_leaf_nodes ∈ {15,31,63}) at threshold = 0.06

– Best combo: learning_rate = 0.05, max_leaf_nodes = 63 → Precision = 0.294, Recall = 0.629, F1 = 0.400

7. FINAL MODEL & ARTIFACTS

- Retrained final HGB classifier on full dataset with optimal settings
- Final performance at threshold = 0.06: Precision = 0.294, Recall = 0.629
- Model persisted to models/hgb_final.joblib
- Processed data and feature files saved in data/processed/

8. CONCLUSIONS

- Supervised HGB model significantly outperforms unsupervised baseline (F1 from ~0.05 to ~0.40)
- Rolling-window size of 5 and inclusion of percentiles and diff features provide best signal
- Final model meets modest detection targets but falls short of stakeholder goal (Recall ≥ 0.80, Precision ≥ 0.50)

9. RECOMMENDATIONS & NEXT STEPS

a. Define and agree on concrete success criteria with stakeholders

b. Explore additional domain-informed features (longer windows, interaction terms)

c. Evaluate alternative anomaly detection methods (LocalOutlierFactor, autoencoders)

d. Implement model deployment pipeline with monitoring of live performance and drift

e. Iterate on feature and model design to reach recall ≥ 0.80 and precision ≥ 0.50