Anomaly Detection Report

By: Chetan Sai Borra

1. Data Preprocessing:

The dataset consisted of daily measurements for various KPIs collected from multiple telecom network sectors.

These KPIs included metrics such as RSRP, SINR, DL/UL Throughput, RTT, CPU Utilization, Active Users, and Packet Loss, etc.

To ensure consistency, all date entries were standardized using pandas.to_datetime.

The dataset contained **no missing values**.

Each sector's data was then organized as a time series by sorting records chronologically.

To ensure sufficient data for meaningful analysis, only sectors with at least 10 valid entries for a given KPI were included in the anomaly detection process.

Domain-Informed Outlier Removal:

Domain-informed bounds were applied to each KPI to eliminate physically implausible or operationally invalid values.

These thresholds were based on realistic ranges derived from network engineering standards and empirical observations in production telecom environments.

Outlier filtering followed a two-step strategy:

1. Lower Bound: Defined by physical or logical constraints (e.g., no negative throughput, call drop rates ≥ 0).
2. Upper Bound: Defined using quantile-based clipping (e.g., 97th/98th percentile) to remove sporadic, non-representative spikes caused by measurement artifacts, congestion bursts, or logging glitches.

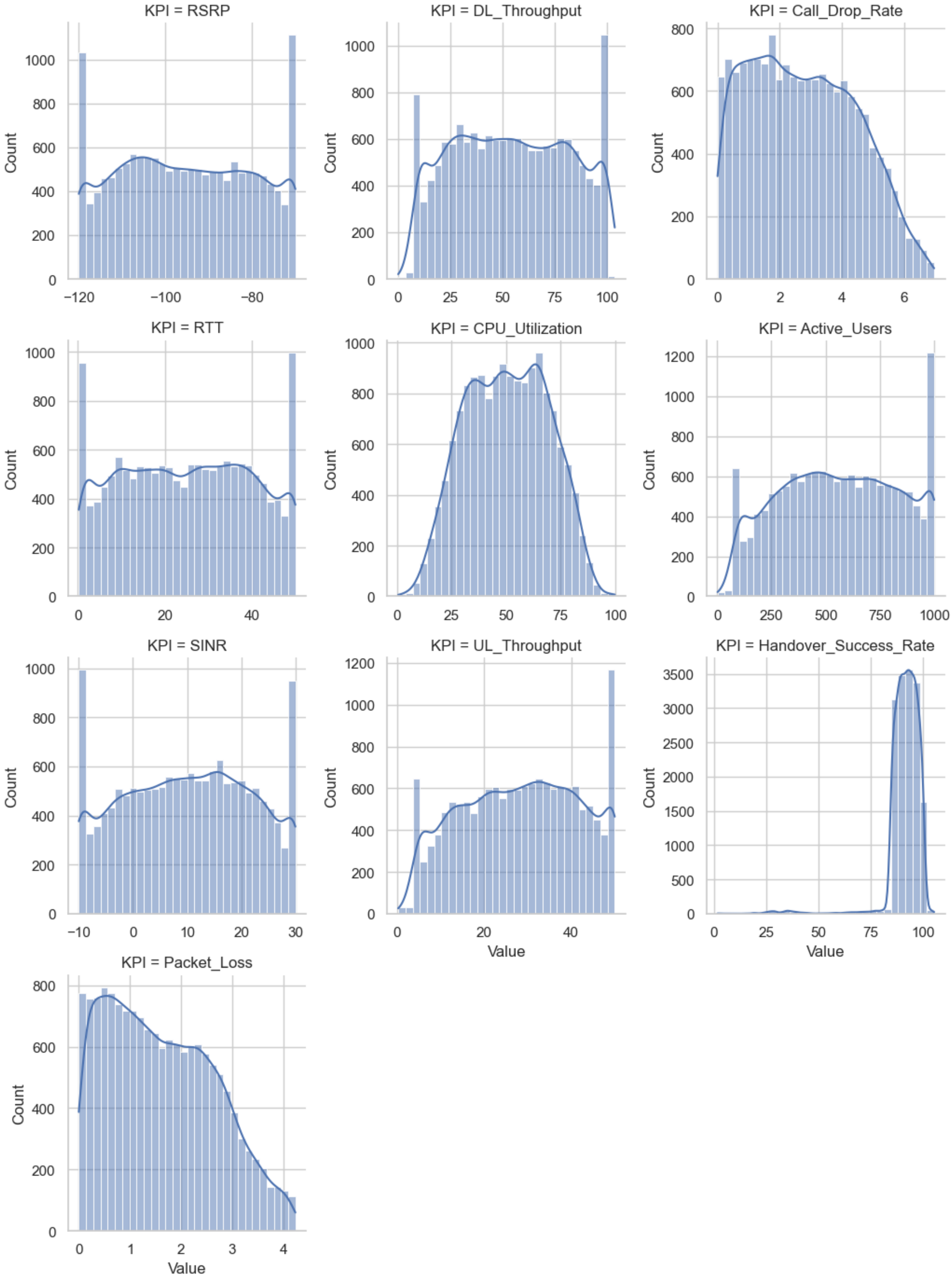
The table below outlines the KPI-specific bounds applied:

KPI	**Lower Bound**	**Upper Bound Strategy**	**Justification**
RSRP	-120 dBm	97th percentile	RSRP typically ranges from -120 to -60 dBm; higher values are likely noise.
DL_Throughput	0 Mbps	98th percentile	Negative throughput is invalid; upper cap removes rare measurement spikes.

KPI	**Lower Bound**	**Upper Bound Strategy**	**Justification**
Call_Drop_Rate	0%	98th percentile	Drop rates below 0 are invalid; upper clipping removes localized issues.
RTT	0 ms	97th percentile	Negative latency is impossible; top-end clipping accounts for congestion.
CPU_Utilization	0%	100% (hard limit)	Valid range is 0–100%; anything outside is erroneous.
Active_Users	0 users	98th percentile	User count must be ≥0; upper tail trimmed to remove rare logging errors.
SINR	-10 dB	97th percentile	SINR below -10 dB is atypical; high values often reflect measurement errors.
UL_Throughput	0 Mbps	98th percentile	Same logic as DL throughput.
Handover_Success_Rate	0%	98th percentile	Ideally 0–100%; clipping removes misleading near-100% artifacts.
Packet_Loss	0%	97th percentile	Negative loss is invalid; upper cap addresses burst errors or outliers.

By anchoring the data within domain-validated bounds, the models were able to focus on meaningful deviations rather than noise.

Distribution of KPI values across the dataset. Each subplot shows the distribution of a specific KPI plotted using a histogram with KDE (Kernel Density Estimation).


KPI = RSRP
KPI = DL_Throughput
KPI = Call_Drop_Rate
KPI = RTT
KPI = CPU_Utilization
KPI = Active_Users
KPI = SINR
KPI = UL_Throughput
KPI = Handover_Success_Rate
KPI = Packet_Loss
Count
Value
−120
−100
−80
0
25
50
75
100
2
4
6
20
40
250
500
750
1000
−10
10
30
1
3
200
400
600
800
1200
500
1000
1500
2000
2500
3000
3500

2. Model Selection and Rationale:

To effectively detect anomalies in the multivariate telecom KPI time series data, I employed a combination of complementary unsupervised methods. Models was selected based on its suitability for time-series anomaly detection and its performance in comparative evaluations in the literature.

DWT-MLEAD (Discrete Wavelet Transform + MLEAD):
Wavelet Decomposition:
The input time series is decomposed using Discrete Wavelet Transform (DWT) into a set of coefficients representing different frequency bands (details and approximation).

This multiresolution view helps detect changes in trend, sudden spikes, or repetitive patterns across various time scales.

MLEAD Detection:
After decomposition, the MLEAD algorithm identifies outliers in the wavelet domain by estimating the likelihood of observed changes. It assigns anomaly scores based on statistical deviations in the reconstructed signal.

- Telecom KPIs often exhibit non-stationary behaviour, which DWT captures effectively.
- It identifies shape-based anomalies—not just outliers in magnitude.
- Robust to local spikes, noise, and non-Gaussian distributions.
- Particularly effective for KPIs like RSRP, SINR, and UL/DL Throughput, where subtle trend changes may indicate early signs of network degradation.

Isolation Forest:

- Isolation Forest is a tree-based algorithm that isolates observations by randomly selecting a feature and a split value.
- Normal points require more splits to isolate, whereas anomalies get isolated faster.
- The average path length of a point across trees becomes its anomaly score (shorter path = more anomalous).

Method 2: Ensemble-Based Anomaly Detection (Voting Mechanism):

To improve anomaly detection precision and reduce false positives, I implemented an ensemble approach by combining two unsupervised methods: DWT-MLEAD and Isolation Forest. This voting-based method considers a data point as anomalous only if both models agree.

Approach:

For each KPI and each sector:

- Apply DWT-MLEAD to detect temporal anomalies in the KPI time series.

- Apply Isolation Forest to capture point anomalies based on statistical deviation.

- Identify common anomaly indices that are flagged by both models.

- Retain only those anomalies as ensemble-detected anomalies.

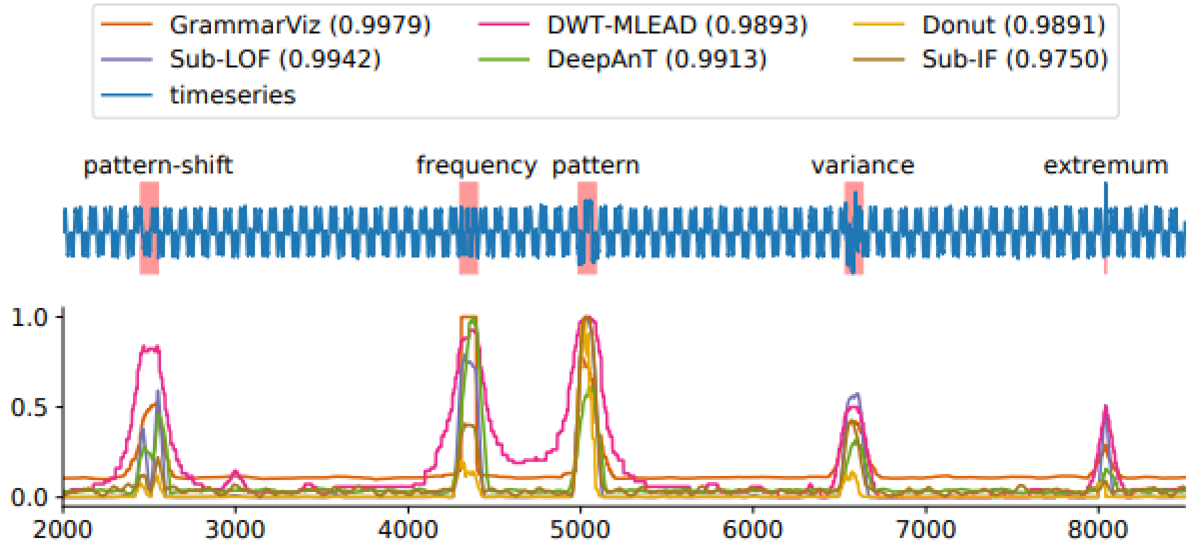


Fig: 1 This figure visualizes the output of multiple anomaly detection algorithms applied to a synthetic time series exhibiting various anomaly types: pattern-shift, frequency change, pattern repetition, variance spike, and extremum.

Reference:
Schmidl, S., Wenig, P., & Papenbrock, T. (2022). *Anomaly Detection in Time Series: A Comprehensive Evaluation*.
Hasso Plattner Institute, University of Potsdam; Philipps University of Marburg.

3. Hyperparameter Tuning:

Effective anomaly detection depends not only on model selection but also on carefully chosen hyperparameters. I adopted a combination of empirical tuning, domain knowledge, and visual validation to finalize hyperparameters for each model. Below is a detailed breakdown of the tuning process:

Isolation Forest:

Final Parameters:

- n_estimators = 100
- contamination = 0.03

Tuning Strategy:

- I began by testing values of n_estimators ranging from 50 to 300. A setting of 100 trees provided a good balance between model stability and computational efficiency.
- The contamination parameter, which defines the expected proportion of anomalies in the dataset, was tuned based on domain expectation and anomaly density observed in exploratory data analysis. A setting of 0.03 (3%) reflected the typical upper-bound of daily anomaly frequency across sectors.

DWT-MLEAD:

Final Parameters:

- wavelet = 'db4'
- decomposition_level = 3
- threshold_ratio = 2

Tuning Strategy:

- I evaluated multiple wavelet families (db2, db4, sym5, coif1) and found that Daubechies 4 (db4) provided the best trade-off between temporal locality and frequency sensitivity.
- The decomposition level was tuned based on the length of time series. A level of 3 was optimal for the 60-day KPI sequences, preserving enough resolution to detect mid-frequency trends.
- Thresholding in the wavelet domain was applied using a z-score-based logic. A threshold ratio of 2 (i.e., values exceeding 2 standard deviations) was empirically chosen to balance sensitivity and false positives. This threshold was validated visually on signal reconstructions to confirm accurate anomaly localization.

4. Performance Evaluation

Evaluating the quality of unsupervised anomaly detection is inherently challenging due to the absence of labelled ground truth.

Instead, I employed a combination of **visual validation**, **anomaly rate heuristics** to evaluate model effectiveness and calibration.

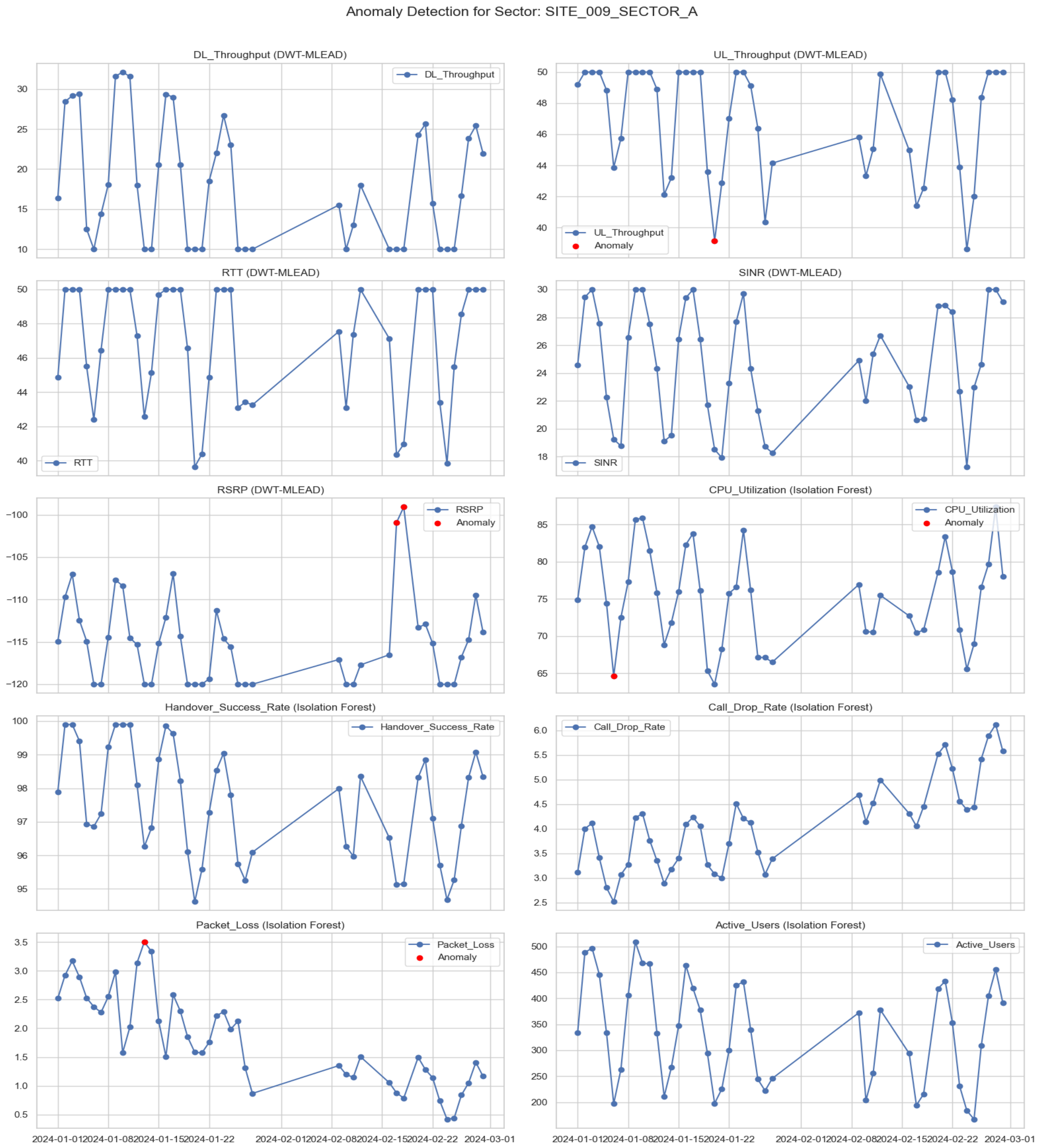
Evaluation Methods:

1. Visual Inspection

I manually reviewed line plots of time series overlaid with model-identified anomalies for a representative sample of KPIs and sectors. This qualitative approach allowed us to:

- Confirm whether anomalies aligned with expected behaviours (e.g., sudden drops in SINR, spikes in Packet Loss).
- Detect overfitting or excessive sensitivity by observing whether anomalies clustered around normal operational patterns.
- Adjust model thresholds or ensemble voting rules accordingly.

Anomalies for SITE_009_SECTOR_A:



2. Anomaly Rate Heuristic

I used expected anomaly frequency as a guiding metric. Based on domain knowledge and literature, a reasonable anomaly rate in telecom operational datasets is typically: 0.5% to 5% of all observations

After running the models, I computed the anomaly rate for each KPI-sector time series. Models or thresholds were considered well-calibrated if the anomaly rate stayed within this range.

- Rates <0.5% suggested the model may be too conservative, missing subtle but valid anomalies.
- Rates >5% indicated potential overfitting or noise amplification.

Thank You!