Common Q&A for Poster Presentation Pitch

Q1: Why does setting quantile-based thresholds guarantee any desired false-positive rate?

A1: By fitting a Generalized Pareto Distribution (GPD) to exceedances over a baseline threshold, we obtain an explicit model of the nominal data's tail. Selecting the $(1-\alpha)$-quantile from this GPD ensures that only an α fraction of nominal values exceed the decision threshold, directly controlling the false-positive rate.

Q2: How do you choose the initial threshold (u) for exceedance collection?

A2: The threshold u should be high enough that exceedances capture the tail behavior but low enough to provide sufficient data for a stable GPD fit. In practice, heuristics like using the 90th or 95th empirical percentile on an initial batch (e.g., first 1,000 points) are common.

Q3: How frequently should the GPD model be updated online?

A3: Updates occur whenever a new exceedance is observed. Each exceedance is added to the fitting set, and the GPD parameters are refit (often via incremental MLE). This continuous update allows tracking of slow changes in the nominal tail distribution.

Common Q&A for Poster Presentation Pitch

Q4: How does DSPOT handle abrupt concept drift?

A4: DSPOT detrends observations using a moving average window. For abrupt shifts, the moving average rapidly adjusts, and subsequent exceedances reflect deviations from the new baseline, allowing the GPD fit to recalibrate to the new regime.

Q5: What is the computational overhead of SPOT/DSPOT?

A5: Each new exceedance triggers an MLE update for GPD parameters, which is O(k) in the number of exceedances. Since exceedances are rare by design, the average per-point cost remains low, making the method suitable for high-throughput streams.

Q6: Can this approach be extended to multivariate streams?

A6: Yes. One can model marginal tails via univariate GPDs and then apply multivariate tail dependence frameworks (e.g., copulas) to capture joint extremes, or use directional projections to reduce dimensionality.

Common Q&A for Poster Presentation Pitch

Q7: What other anomaly detectors are used in our comparison?

A7: We compare SPOT and DSPOT against several state-of-the-art streaming detectors: - ADWIN (Adaptive Windowing): detects change points by monitoring the mean in a variable-length window. - Isolation Forest: an ensemble of random trees isolating anomalies in subwindows. - RRCF (Robust Random Cut Forest): builds trees to measure the isolation of points in a stream. - HTM (Hierarchical Temporal Memory): a biologically inspired model for sequence prediction and anomaly scoring.

Q8: What is the step-by-step SPOT algorithm?

A8: 1. Initialization: collect first N points and set initial threshold u as their p-th percentile. 2. For each new data point x_t: a. If x_t <= u: do nothing. b. If x_t > u: record excess y = x_t - u, add to exceedance set, and re-estimate GPD parameters (ξ, β). Then compute new threshold t = u + (β/ξ)((1-α)^{-ξ} - 1)). 3. Slide u dynamically by keeping a fixed number of recent non-exceedance points if desired. 4. Raise an alarm whenever x_t > t.

Q9: Why use the Numenta Anomaly Benchmark (NAB) and what is it?

A9: The Numenta Anomaly Benchmark (NAB) is a widely adopted open-source dataset and evaluation framework for real-time anomaly detection in streaming data. It provides diverse real-world and artificial time-series, labeled anomalies, and a scoring methodology that rewards early and accurate detection. We use NAB to demonstrate our methods on community-accepted benchmarks and compare fairly with other detectors.