A CONFIGURABLE UNSUPERVISED ANOMALY DETECTION SERVICE FOR THE ADVANCED LIGHT SOURCE

K. Iliev*, T. Hellert, S. C. Leemann, G. Martino, A. Sulc
Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Abstract

Advanced Light Source (ALS) control systems operate with continuous streams of more than 200 000 EPICS Process Variables (PVs). While most alarms are handled by manually setting a threshold, we present a configurable service that scores live PV streams for anomalies in real time, providing a flexible fault discrimination tool for operators. The service applies a recurrent neural network to capture time-dependent correlations, trained on a given time range using unsupervised one-class loss that requires no labeled fault examples. Furthermore, we embed operationally-aware logic that suppresses false alerts during machine-off periods, ensuring detection quality without manual setpoint tuning per PV. We show the system in operation on live data from the ALS linac modulator capacitor voltage divider channels, and describe its architecture as a configurable, multi-instance service driven by a single JSON specification, enabling straightforward extension to additional PV groups as training data becomes available.

INTRODUCTION

The Advanced Light Source (ALS) [1] at Lawrence Berkeley National Laboratory is a third-generation synchrotron light source currently undergoing a major upgrade (ALS-U). Reliable accelerator operation depends on continuous monitoring of the machine state through the Experimental Physics and Industrial Control System (EPICS) [2], which at ALS exposes more than 200 000 PVs spanning magnets, RF systems, vacuum, diagnostics, and insertion devices.

The conventional approach to fault detection relies on per-channel scalar thresholds: an alarm is raised when a single PV exceeds a manually configured limit. While effective for gross failures, this strategy has well-known limitations. First, setting and maintaining thresholds for a large number of channels is labor-intensive and prone to alarm fatigue. Second, scalar thresholds are blind to *temporal correlations*: a slow drift or an unusual time-dependent pattern across multiple related PVs may indicate an incipient fault long before any individual channel crosses its static limit. Third, machine state transitions, beam dumps, injection pauses, planned shutdowns, routinely drive PV values through alarm regions, generating false positives that erode operator trust.

Machine learning offers a path beyond these limitations [3, 4]. In this paper we present an anomaly detection service that learns the temporal structure of normal PV behavior from unlabeled historical data and produces a single, interpretable anomaly score in real time.

* kirkiliev@lbl.gov

SYSTEM ARCHITECTURE

Design Goals

The service was designed around three requirements: *(i)* zero labeled fault data, *(ii)* real-time operation on live EPICS archiver streams, and *(iii)* straightforward extension to new PV groups without code changes. These goals led to a pipeline architecture with three phases: training, context bootstrapping, and streaming inference, orchestrated by a single JSON configuration file.

Configuration-Driven Multi-Instance Design

Each detection instance is fully defined by an entry in the JSON configuration (Fig. 1). The specification includes the list of PVs to monitor jointly, training data window, time-grid resolution, model hyperparameters, inference polling interval, and anomaly threshold scaling factors. Multiple instances run as independent threads within a single service process, enabling concurrent monitoring of unrelated PV groups.

Adding a new monitored group requires only a new JSON block, a suitable period of historical normal-operation data, and a service restart. The training pipeline then executes automatically, producing a self-contained checkpoint that stores model weights, normalization statistics, and calibrated thresholds.

```
[{
  "instance_name": "LNRF_Modulators",
  "pvs": ["LNRF:MOD1:CapVoltageDivider",
          "LNRF:MOD2:CapVoltageDivider"],
  "training": {
    "start_date": "2025-12-08...", "step_sec": 60,
    "seq_len": 10, "std_clamp": 0.5, "...",
    "model_params": {"latent_dim": 8, "..."}
  },
  "inference": {
    "context_hours": 2, "threshold_scale_warning": 2.0,
    "threshold_scale_anomaly": 3.5, "...",
    "output_pvs": {
      "score": "LNRF:MOD1:CapVoltageDivider:
           AnomalyScore"
    }
  },
  "checkpoint_path": "models/checkpoints/
       lnrf_modulators_v1.pt"
}]
```

Figure 1: Abbreviated JSON configuration.

Pipeline Overview

Figure 2 illustrates the system pipeline. Historical PV data is fetched from the EPICS Archiver Appliance [5], aligned to a uniform time grid, and used to train a one-class Gated Recurrent Unit (GRU) model. The resulting checkpoint is loaded for streaming inference, where new data is continuously merged into a rolling context buffer and scored against the learned normal-behavior manifold.

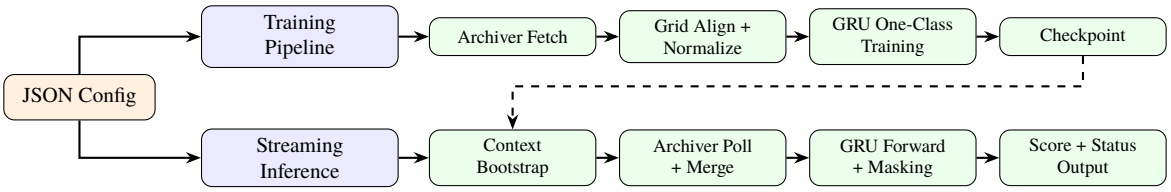

Figure 2: System pipeline using a Gated Recurrent Unit (GRU) one-class encoder. Training produces a self-contained checkpoint that is loaded for streaming inference. Dashed arrow indicates the checkpoint link between the two phases.

MODEL AND TRAINING

Data Preparation

PV data is retrieved from the EPICS Archiver Appliance [5], which stores values at irregular intervals. The raw samples are aligned onto a uniform time grid of configurable resolution via zero-order hold: each point assumes the value of the most recent sample at or before that time. Rows where any PV falls outside a configurable per-channel range are discarded, removing machine-off periods and out-of-bounds samples from the training set.

Each channel is then normalized using its empirical mean and standard deviation, with the standard deviation floored at a configurable minimum to prevent numerical instability on near-constant channels. Normalization statistics are stored in the checkpoint and applied identically at inference time.

A sliding window of length L is applied to the normalized data, producing three-dimensional training tensors of shape $[N, L, C]$, where N is the number of sequences and C is the number of channels.

Each sequence undergoes *baseline subtraction*: the median of a configurable number of timesteps is subtracted from all timesteps. This removes the absolute signal level from the representation, forcing the model to learn *temporal deviation patterns* rather than static operating points. The use of a median rather than mean makes the baseline robust to brief noise spikes.

GRU One-Class Encoder

The model is a GRU [6] encoder that consumes the multichannel time series and produces a compact latent vector summarizing its temporal structure, via a single linear projection from the GRU's final hidden state. All bias terms throughout the network are disabled, a requirement of the Deep One Class Classification (Deep OCC) objective [7] that prevents the trivial solution where the network learns a constant offset equal to the target center.

One-Class Loss (Deep OCC)

Training of model parameters θ minimizes the mean squared distance of latent representations from a fixed center $\mathbf{c}$, i.e. $\mathscr{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{z}_i - \mathbf{c}\|^2$. The center $\mathbf{c}$ is computed once before training by a warm-up forward pass over all training sequences through the randomly initialized network; $\mathbf{c}$ is set to the mean of the resulting latent vectors and is then fixed. This standard Deep OCC [7] prevents the degenerate mode where a bias-free network maps everything to the origin. Training uses the Adam optimizer with gradient clipping (max norm 1.0).

Threshold Calibration

After training, all training sequences are re-encoded and their distances from $\mathbf{c}$ computed. Warning and anomaly thresholds are defined as $\tau_{\text{warn}} = s_{\max} \cdot \alpha_w$ and $\tau_{\text{anom}} = s_{\max} \cdot \alpha_a$ where $s_{\max}$ is the maximum score over the training set and $\alpha_w < \alpha_a$ are user-configurable scaling factors. Using the training-set maximum guarantees that no training point would have triggered an alert. In practice, $s_{\max}$ can be sensitive to individual high-scoring sequences in the training set; replacing it with a high percentile (e.g., 99.5th or 99.9th) provides a more robust reference while still placing the thresholds above nearly all observed normal behavior. Both options are supported via the JSON configuration.

STREAMING INFERENCE

Context Bootstrapping

Before the streaming loop begins, the system fetches the most recent few hours of PV data from the archiver and aligns it to the model's time grid, forming a rolling context buffer in memory. This pre-warms the sliding window so that inference can start immediately rather than waiting for L new live points to accumulate.

Real-Time Scoring Loop

The inference loop runs on a fixed, user-configured timer (default 10 s). On each tick:

1. The most recent archiver data is fetched for each PV.
2. New samples are merged into the rolling context buffer using grid-bucket alignment: a new row is appended only when the wall clock crosses the next grid point; within a bucket, the most recent value updates the current row in place.
3. The context window is normalized using the stored training statistics, baseline-subtracted, and passed through the GRU encoder.
4. The anomaly score $s = \|\mathbf{z} - \mathbf{c}\|$ is compared against τ_{warn} and τ_{anom}.

Each tick produces a timestamped record containing the instance name, current PV values, anomaly score, and status label (`NORMAL`, `WARNING`, `ANOMALY`, or `OFF`).

Operational Masking

A lesson from running anomaly detection on live accelerator data is that machine state transitions dominate the false positive rate. Two masking mechanisms address this:

Machine-off masking. When PV values fall outside a physics-motivated boundary (e.g., under 200 kV for the linac

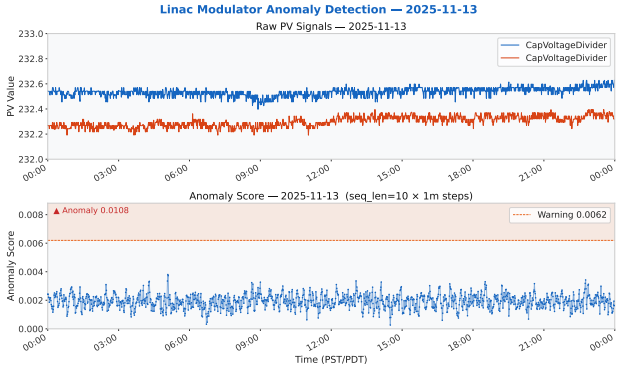

Figure 3: Normal-operation day (2025-11-13). Top: raw capacitor voltage divider signals (kV) for the two linac modulators. Bottom: corresponding anomaly score, remaining well below both the warning ($\tau_{\text{warn}} = 0.0062$) and anomaly ($\tau_{\text{anom}} = 0.0108$) thresholds throughout the day.

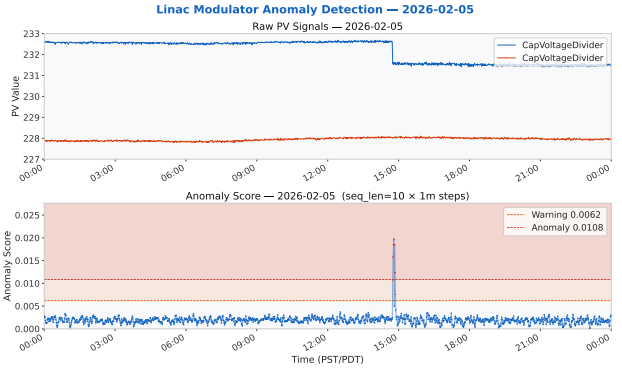

Figure 4: Anomaly detection on a known fault day (2026-02-05). Top: raw PV signals (kV) showing an abrupt drop in one modulator channel around 14:00. Bottom: the anomaly score breaches τ_{anom} within minutes and remains elevated until the fault leaves the context window.

modulator capacitor voltage), the channel is treated as off. Off-state values are replaced by forward- and backward-fill of the last valid reading, preventing the model from seeing artificial zeros or discontinuities. The resulting score is suppressed and reported as `OFF`.

Recovery suppression. After a machine-off → machine-on transition, the context buffer still contains stale filled values. To prevent false anomaly alerts during this transient, scoring is suppressed for a configurable number of recovery steps (default 10 grid points) following any off-state observation.

These two mechanisms are essential for practical deployment: without them, every planned beam dump or injection pause would generate spurious alerts indistinguishable from genuine anomalies.

OPERATIONAL EVALUATION ON ALS LINAC MODULATORS

The service has been running on the capacitor voltage divider channels of the two linac modulator PVs `LNRF:MOD{1,2}:CapVoltageDivider`. These PVs track the stored energy in the modulator capacitor banks and serve as key indicators of linac health.

For this application, the training data covers a period of stable user operations sampled on a 1 min grid ($s = 60$ s), with a sliding window of $L = 10$ time steps (i.e., a 10 minute context window). Baseline subtraction uses the median of the full sequence. The per-channel standard deviation floor is set to 0.5, and the GRU latent dimension is $d = 8$ (selected empirically for this PV group). Gradient clipping is applied with a maximum norm of 1.0. Threshold calibration uses the maximum training-set score as the reference, yielding $\tau_{\text{warn}} = 0.0062$ and $\tau_{\text{anom}} = 0.0108$. The machine-off voltage threshold is set to 200 kV.

In operation, the service runs continuously on a development server, polls the archiver every 10 s and produces a score log that can be either consumed by higher-level dashboards or is directly forwarded to EPICS via `caput`. Figure 3 shows a representative day of stable beam operations: the anomaly score remains steady near 0.002, well below both the warning and anomaly thresholds.

Figure 4 demonstrates the system responding to a known hardware event on 2026-02-05, in which one modulator capacitor voltage divider channel dropped abruptly by 1 kV around 14:00 (switch unit failure). The anomaly score rises sharply above τ_{anom} within minutes of the onset, reaching approximately twice the anomaly threshold, and returns to baseline once the fault has left the context window.

This transient response is by design: the short context window ($L = 10$) targets abrupt fault modes while remaining insensitive to slow operational drifts. An operator-facing deployment therefore requires a latching PV that captures the transient flag and holds it until manually acknowledged and reset by an operator, separating detection (event-style, in software) from alarm persistence (latched, in EPICS). The configuration-driven architecture allows this detector character to be tuned per PV group: shorter windows favor event-style detection of abrupt faults; longer windows favor state-style tracking of persistent deviations.

CONCLUSION

We have presented a configurable, real-time anomaly detection service for EPICS-based accelerator control systems, under operational evaluation at ALS on linac modulator channels. The service combines a GRU-based one-class encoder with operationally-aware masking to deliver reliable, unsupervised fault detection. Its configuration-driven architecture enables straightforward extension to new PV groups, supporting the growing monitoring demands of ALS and ALS-U. Future work will focus on integration with the ALS alarm handler, expansion to additional subsystems, and exploration of attention-based architectures for longer temporal contexts.

ACKNOWLEDGEMENT

This work was supported by the Director of the Office of Science of the U.S.Department of Energy under Contract No. DEAC02-05CH11231.

REFERENCES

[1] T. Hellert *et al.*, "Status of the Advanced Light Source", in *Proc. IPAC'24*, Nashville, TN, USA, May 2024, pp. 1309–1312. doi:10.18429/JACoW-IPAC2024-TUPG37

[2] L. R. Dalesio *et al.*, "The Experimental Physics and Industrial Control System Architecture: Past, Present, and Future", *Nucl. Instrum. Methods Phys. Res., Sect. A*, vol. 352, no. 1–2, pp. 179–184, 1994. doi:10.1016/0168-9002(94)91493-1

[3] Y. Lu *et al.*, "Demonstration of Machine Learning–Enhanced Multi-Objective Optimization of Ultrahigh-Brightness Lattices for 4th-Generation Synchrotron Light Sources", *Nucl. Instrum. Methods Phys. Res., Sect. A*, vol. 1050, p. 168192, 2023. doi:10.1016/j.nima.2023.168192

[4] A. Sulc, A. Eichler, and T. Wilksen, "A Data-Driven Anomaly Detection on SRF Cavities at the European XFEL", in *Proc. IPAC'22*, Bangkok, Thailand, Jun. 2022, pp. 1152–1155. doi:10.18429/JACoW-IPAC2022-TUPOPT062

[5] M. V. Shankar *et al.*, "The EPICS Archiver Appliance", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015, pp. 761–764. doi:10.18429/JACoW-ICALEPCS2015-WEPGF030

[6] K. Cho *et al.*, "Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation", in *Proc. EMNLP'14*, Doha, Qatar, Oct. 2014, pp. 1724–1734. doi:10.3115/v1/D14-1179

[7] L. Ruff *et al.*, "Deep One-Class Classification", in *Proc. ICML'18*, Stockholm, Sweden, Jul. 2018, pp. 4393–4402.