

DF-PLNet: Budgeted Periodic Filtering for Industrial Control System Process-Record Diagnostics

Yipeng Zhang

North China University of Technology

Jiahui Chen

chen_jiahui@mail.ncut.edu.cn

North China University of Technology

Yue Li

Shanghai Branch

Yun Shi

North China University of Technology

Article

Keywords: Industrial Control System, Budgeted Periodic Filtering, Process-Record Diagnostics, Time-Series Anomaly Detection, DF-PLNet, Residual-Stream Evaluation

Posted Date: October 16th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10327330/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

DF-PLNet: Budgeted Periodic Filtering for Industrial Control System Process-Record Diagnostics

Yipeng Zhang¹, Jiahui Chen^{1,*}, Yue Li², and Yun Shi¹

¹School of Computer Science and Technology, North China University of Technology, No. 5 Jinyuan Road, Shijingshan District, Beijing 100144, China

²China Electronic Port Data Center, Shanghai Branch, 35th Floor, No. 211 Century Avenue, Pudong New Area, Shanghai 200135, China

*Corresponding author: chenjiahui@mail.ncut.edu.cn

¹E-mail addresses: zhangyipeng@ncut.edu.cn (Y.Z.); chenjiahui@mail.ncut.edu.cn (J.C.); liyue@customs.gov.cn (Y.L.); 2024322040106@mail.ncut.edu.cn (Y.S.)

ABSTRACT

Industrial control system (ICS) anomaly detection protects critical infrastructure, but studies on shared datasets often rely on released process records rather than raw telemetry or packet payloads. We study this setting as offline residual-stream diagnostics on SWaT and HAI splits, converting each timestamped row into a single scaled deviation value paired with its time interval. The task is challenging because normal operation is periodic, anomaly labels are imbalanced, periodic pruning must remain label-blind, and test labels must not guide threshold selection. Dual-Feature Periodic Learning Network (DF-PLNet) combines label-blind Budgeted Periodic Filtering, serialized value/interval inputs, and a U-shaped encoder-decoder for record-position scoring under a train-only threshold protocol. Across three seeds (mean $\pm$ s.d.), DF-PLNet with a 10 % pruning budget obtains retained-stream area-under-curve (AUC) and F1 values of $0.9505 \pm 0.0150/0.8966 \pm 0.0122$ on SWaT and $0.6486 \pm 0.0029/0.3112 \pm 0.0091$ on HAI. A 0–20 % budget sweep shows that 5 % pruning offers a more conservative trade-off on SWaT, whereas 20 % pruning is harmful on HAI. Coverage-aware F1 decreases from 0.8866 to 0.8694/0.8429/0.7623 on SWaT and from 0.3118 to 0.2967/0.2906/0.1122 on HAI at 5 %, 10 %, and 20 %, showing that retained-stream gains require all-record coverage.

Keywords: Industrial Control System; Budgeted Periodic Filtering; Process-Record Diagnostics; Time-Series Anomaly Detection; DF-PLNet; Residual-Stream Evaluation

1 Introduction

Industrial control systems (ICS) support critical infrastructure, including water treatment, power generation, and petrochemical refining. Their convergence with enterprise IT networks enables remote monitoring and data-driven optimization, but also expands the attack surface of once-isolated environments [1]. Incidents such as Stuxnet and TRITON show that ICS compromises can have severe operational and physical consequences [2, 3]. In many post-incident or shared-dataset studies, however, the available artifact is a chronological process-record file rather than a complete packet trace or raw plant capture. For a water-treatment plant, a monitor may receive only timestamped sensor and actuator rows with numeric process variables and binary labels. Because much of such a stream reflects routine periodic operation, pruning periodic background before scoring is a natural way to focus analyst attention. The central question is whether conservative periodic filtering can reduce downstream scoring and review without making the filtering stage label-dependent.

Existing detection methods remain limited in this setting along three dimensions. *First*, payload-dependent or protocol-specific methods [4, 5] are difficult to transfer to released process-record artifacts that no longer expose packet payloads or protocol fields. *Second*, general network-level metadata detectors, exemplified by Kitsune [6], motivate lightweight monitoring but do not directly model the process-cycle periodicity that dominates water-treatment and plant-control traces. *Third*, recent protocol-light and deep traffic detectors [7, 8] can learn useful representations, but comparing them in our setting requires careful handling of class imbalance and train-only threshold selection, together with clarity about whether evaluation covers all records or only a filtered residual stream.

These limitations define our target setting: offline residual-stream diagnostics on chronological process records. We use label-blind periodic preprocessing with an explicit drop budget and a train-threshold protocol in which test labels are used only for final evaluation.

This setting introduces three coupled challenges. First, periodic filtering must be conservative because unconstrained filtering can remove too many records and create an artificial residual stream. Second, the model must extract discriminative signal from low-dimensional value/interval sequences, because the released records compress multivariate plant behavior into one scalar per timestamp. Third, thresholding must be separated from test evaluation so that reported operating points do not

25 rely on held-out labels.

26 DF-PLNet addresses these challenges with a label-blind Budgeted Periodic Filtering Module (PFM) and a residual-stream
27 detector. PFM matches periodic structure from timestamps, record-level scalar values, and optional symbols, then prunes only
28 a bounded subset of periodic candidates. Value/interval serialization and a U-shaped encoder–decoder score retained record
29 positions with local and longer-range context. The operating threshold is selected from training scores and labels only. We
30 evaluate the pipeline with explicit train/test process-record files, repeated runs over three random seeds, label-aligned metrics,
31 PFM budget sweeps, and coverage-aware audits.

32 The main contributions of this paper are as follows.

- 33 1. **Process-record diagnostic setting** (Sections 3 and 4). We formulate released ICS process-record monitoring as offline
34 diagnosis on timestamped scalar deviation records, with retained-stream and all-record coverage accounting.
- 35 2. **Label-blind periodic-filtering procedure** (Sections 4.3, 4.4, and 5.8). We instantiate a Budgeted PFM stage that uses
36 timestamps, values, and optional symbols without labels, then masks periodic candidates within fixed 5 %, 10 %, and
37 20 % drop budgets.
- 38 3. **Residual-stream detector and train-threshold protocol** (Sections 4.2, 4.5, and 4.6). We implement value/interval
39 serialization and a U-shaped encoder–decoder backbone that scores retained residual streams under a train-only threshold
40 protocol. This component provides the record-level scorer used throughout the diagnostic pipeline.
- 41 4. **Protocol-compatible evaluation and coverage audit** (Section 5). We evaluate convolutional neural network (CNN),
42 DF-PLNet without PFM, DF-PLNet, and Transformer baselines under matched train/test files, epoch budget, seed
43 set, and threshold selection. The 0–20 % budget sweep shows that retained-stream gains must be read together with
44 coverage-aware F1.

45 The remainder of this paper reviews related work, defines the diagnostic setting, presents DF-PLNet, reports the SWaT and
46 HAI process-record experiments, and discusses limitations.

47 2 Related Work

48 2.1 ICS protocol security and vulnerability analysis

49 ICS security is closely tied to weaknesses in industrial communication protocols. Protocols such as Modbus, DNP3, and
50 PROFINET prioritize low-latency operation over authentication and encryption; Modbus/TCP, for example, exposes function
51 codes and register addresses in plaintext [9]. Security-enhanced protocol suites can add authentication and integrity layers [1],
52 but released datasets and external monitors may still lack payloads, protocol fields, or raw capture metadata.

53 2.2 Intrusion detection for industrial control systems

54 Industrial-network anomaly detection must handle cyber, operational, and service anomalies while balancing placement,
55 latency, and generalization [10]. Protocol-aware detectors parse function codes, register addresses, or state machines [5, 11],
56 but depend on parseable protocol fields. Invariant-based methods mine operational rules but often assume access to process
57 variables or system semantics beyond compact released artifacts [12]. Learning-based intrusion detection systems (IDSs),
58 including Kitsune [6] and packet-payload deep detectors [4], motivate comparison, but their native inputs differ from the
59 reduced process-record stream studied here.

60 2.3 Process-record and metadata-based detection

61 Content-limited detectors use packet lengths, inter-arrival times, flow statistics, behavioral sequences, and process-variable
62 summaries [13]. These studies show that low-dimensional observable features can support anomaly detection. However, the
63 periodic background in ICS records, including polling, acquisition, and control cycles, is rarely handled under an explicit drop
64 budget. DF-PLNet addresses this gap with label-blind Budgeted PFM before value/interval detection.

65 2.4 Deep sequence modeling for anomaly detection

66 Sequential deep learning is widely used for traffic and process anomaly detection. CNNs capture local temporal patterns
67 efficiently [14], LSTMs model longer dependencies [15], and lightweight or cross-domain ICS detectors combine neural
68 models with dimensionality reduction or joint cyber-physical representations [16, 17]. Encoder–decoder architectures, typified
69 by U-Net [18] and U-Time [19], preserve per-position alignment while aggregating wider context [20]. DF-PLNet uses this
70 structure after bounded periodic filtering to expose the trade-off between workload reduction and residual-stream fidelity.

71 3 Background

72 3.1 ICS architecture

73 An ICS combines field devices, control units, monitoring systems, and management integration to regulate industrial processes.
74 Here, the monitor observes chronological process-record streams for offline residual-stream analysis without changing plant
75 operation; the assumed passive observation point in a layered ICS network is illustrated in Supplementary Fig. S1 online.

76 3.2 Process anomaly detection

77 Anomalous records deviate from the normal-operation baseline because of malicious activity, operational error, or hardware
78 faults. ICS detection pipelines span acquisition, preprocessing, feature extraction, modeling, anomaly determination, and alarm
79 generation [21]. Two properties shape the task: industrial systems produce predictable periodic patterns, and attacks may
80 affect only a small fraction of records. This paper targets offline residual-stream diagnosis; online prevention and root-cause
81 attribution remain deployment-oriented extensions.

82 3.3 Detection objective

83 This study treats released ICS process-record monitoring as offline diagnosis on chronological metadata. After train/test splitting
84 and optional budgeted filtering, each retained record contributes one interval position and one value position (Section 4.2). The
85 value position inherits the record label and is the evaluation target; interval positions provide context but are not counted as
86 separate anomalous events. The detector maps each retained sign-separated segment to record-position anomaly scores under
87 three constraints:

- 88 1. **Released-record scope:** the detector uses timestamp, scalar record value, and label files prepared from released SWaT
89 and HAI process records, not raw plant captures.
- 90 2. **Budgeted residual analysis:** the detector should expose how a fixed periodic drop budget changes record-position
91 metrics and evaluated workload.
- 92 3. **Train-only thresholding:** the binary operating threshold is selected on training scores and training labels, then frozen for
93 final test evaluation.

94 3.4 Threat and evaluation model

95 We adopt an offline plant-monitor model: the detector receives processed chronological records, does not modify plant operation,
96 and does not inspect packet payloads. The adversary can cause abnormal sensor or actuator behavior, while the detector
97 observes only timestamps and scalar process-deviation values. The defender has no raw-source reconstruction materials and
98 passes no labels to the periodic front end. In-scope attacks are labeled SWaT and HAI process anomalies; attacks that preserve
99 both scalar deviation and interval context require separate robustness evaluation [22].

100 4 DF-PLNet Framework

101 4.1 Problem definition

102 Given a chronologically ordered sequence of released process records $\mathcal{R} = \{r_1, r_2, \dots, r_n\}$, each record is represented by its
103 timestamp t_i , a scalar record-level value v_i , and a binary training or evaluation label $y_i \in \{0, 1\}$. For SWaT and HAI, v_i is the
104 mean absolute z-score across numeric process variables, computed with statistics from normal training records and scaled to
105 the loader range described in Section 5. Raw packets, protocol fields, and full raw-source reconstruction are not used. The
106 objective is to learn a detector that maps observable value/interval sequences to record-position anomaly decisions on a held-out
107 future partition. Labels are used for supervised training of the neural detector and for final evaluation only; the periodic front
108 end never receives labels.

109 In the water-treatment example, r_i is one released sensor or actuator record after preprocessing. The timestamp determines
110 the interval token, the multivariate row becomes the scalar value token, and PFM decides whether the record belongs to a
111 bounded periodic candidate set before the neural scorer assigns an anomaly score.

112 The proposed pipeline is the Dual-Feature Periodic Learning Network (DF-PLNet). The first stage is a label-blind front
113 end that prunes periodic-matching records under a maximum drop budget. The second stage is a neural record-position
114 detector trained and evaluated on the retained residual sequence. The name reflects the dual value/interval representation, the
115 periodic-filtering front end, and the learned residual-stream scoring backbone. Figure 1 shows the pipeline.

116 4.2 Value and interval sequence representation

117 For each retained process record, the scalar value is clipped at the loader cap and normalized as

$$V'_i = \frac{\min(v_i, 1500)}{1500}. \quad (1)$$

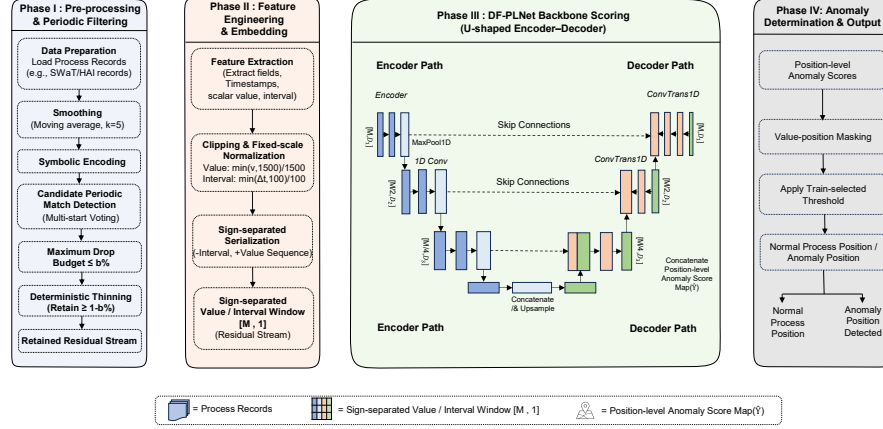


Figure 1. Overview of the DF-PLNet pipeline.

The inter-record interval is computed from adjacent retained timestamps. We first set the raw interval as

$$\Delta t_1 = 0, \quad \Delta t_i = t_i - t_{i-1}, \quad i \geq 2. \quad (2)$$

It is then clipped at $T_{\max} = 100$ and normalized as

$$T'_i = \frac{\min(\max(\Delta t_i, 0), T_{\max})}{T_{\max}}. \quad (3)$$

Each retained process record contributes two serialized positions: a negative interval position and a positive value position. The model input is

$$\mathbf{z} = [-T'_1, V'_1, -T'_2, V'_2, \dots, -T'_m, V'_m], \quad (4)$$

where m is the number of retained records in the partition. This sign convention distinguishes timing context from process-deviation magnitude while keeping a single input channel.

Non-overlapping windows of `segment_len=400` positions are formed for training and inference. Value positions inherit record labels and serve as alert targets; interval positions provide context but are excluded from metrics by a label-aligned mask. Equations (1), (2), (3), and (4) define this retained-record serialization.

4.3 Budgeted Periodic Filtering Module

ICS records often contain periodic polling, control, and reporting patterns [23, 24]. Budgeted PFM treats periodicity as an observable regularity rather than a ground-truth class. It receives timestamps, scalar record values, and optional symbols, but no labels, and is applied independently to each already-collected partition.

Figure 2 summarizes the workflow. Record counts are aggregated in fixed time windows, smoothed, quantized into symbols, and passed to a multi-start period estimator. Candidate periods are scored by template consistency for each starting offset. Votes across offsets select a dominant period when one is reliable. If no reliable period is found, the module falls back to the original stream. The same fallback is used when optional symbol confirmation fails.

The PFM score is defined on fixed aggregation windows. Let c_q denote the record count in window q , let $\tilde{c}_q$ be its smoothed value, and let s_q be the quantized count symbol. For a candidate period p and starting offset a , the residue-template symbol is

$$\tau_{a,p,r} = \text{mode}\{s_q : q \equiv a + r \pmod{p}\}.$$

The count-consistency score is

$$S_c(a, p) = \frac{1}{|\mathcal{Q}_{a,p}|} \sum_{q \in \mathcal{Q}_{a,p}} \mathbb{I}[s_q = \tau_{a,p,(q-a) \bmod p}],$$

where $\mathcal{Q}_{a,p}$ contains the windows covered by the candidate template. When value-derived symbols are available, the same template score is computed over those symbols as $S_v(a, p)$. A period is accepted if it passes the hard count threshold or if it passes the soft count threshold with value-symbol confirmation. Otherwise, PFM keeps the original stream.

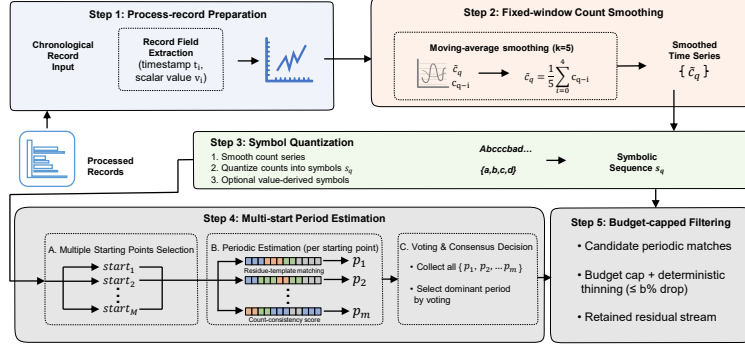


Figure 2. Workflow of the Budgeted Periodic Filtering Module.

The maximum drop budget is the main safety mechanism. For an accepted period, define the label-blind candidate set over retained record indices as

$$\mathcal{C} = \{i : q(i) \text{ matches the accepted periodic template}\}.$$

Given drop budget b and m retained records before pruning, the maximum number of dropped records is $K_b = \lfloor bm \rfloor$. If $K_b = 0$, no candidate is dropped. If $K_b > 0$ and $|\mathcal{C}| \leq K_b$, all candidates are dropped. If $0 < K_b < |\mathcal{C}|$, write $\mathcal{C} = \{c_1 < c_2 < \dots < c_{|\mathcal{C}|}\}$ and select an evenly spaced deterministic subset

$$\mathcal{D}_b = \left\{ c_{\lceil (j-\frac{1}{2})|\mathcal{C}|/K_b \rceil} : j = 1, \dots, K_b \right\},$$

with duplicate selections resolved by the next unused candidate. The final keep mask is the complement of $\mathcal{D}_b$. The main setting uses a 10 % budget, and the sensitivity study covers 0 %, 5 %, 10 %, and 20 %. If the retained set is too small for segmentation, the implementation expands it deterministically without labels.

4.4 PFM reproducibility details

PFM is executed after the explicit process-record split and independently inside each already-collected partition. This batch operation may use global periodic structure within the partition, which is allowed only under the offline diagnostic setting defined in Section 3; label files are subset to the retained records only after the keep mask has been produced.

The final PFM-enabled runs use a window size of 5, a hard periodicity threshold of 0.95, a soft periodicity threshold of 0.40, period search over 2–60 windows, moving-average smoothing with $k = 5$, optional value-derived symbols, and periodic-candidate dropping. The main maximum drop ratio is fixed at 0.10, with a corresponding minimum keep ratio of 0.90. The sensitivity settings use maximum drop ratios of 0.05 and 0.20, with minimum keep ratios of 0.95 and 0.80.

PFM then aggregates, smooths, quantizes, searches periods with a multi-start estimator [25], marks template-matching candidate drops, and caps them with the deterministic budget. With `segment_len=400`, each complete evaluation window corresponds to 200 process records; the unfiltered complete-window test sets contain 54,600 (SWaT) and 230,200 (HAI) evaluated records.

4.5 DF-PLNet backbone for residual-stream detection

The DF-PLNet backbone is the residual-stream detector. It consumes the sign-separated value/interval windows from Section 4.2 and emits one score per position. The model follows a U-shaped encoder–decoder design in the U-Net family [18, 19]. The encoder uses convolutional blocks and max pooling to aggregate local and longer-range context. The decoder restores position-level resolution with transposed convolutions and skip connections. A final 1×1 convolution produces logits, which are converted to anomaly scores by a sigmoid function.

Figure 3 illustrates the layout. The network does not reconstruct protocol fields; it learns patterns in value and interval sequences without using payload contents.

The supervised objective is defined in Equation (5). It combines focal loss [26], class-weighted binary cross-entropy, and Dice loss [27]:

$$\mathcal{L} = w_f L_{\text{Focal}} + w_b L_{\text{BCE}}^{(\text{weighted})} + w_d L_{\text{Dice}}. \quad (5)$$

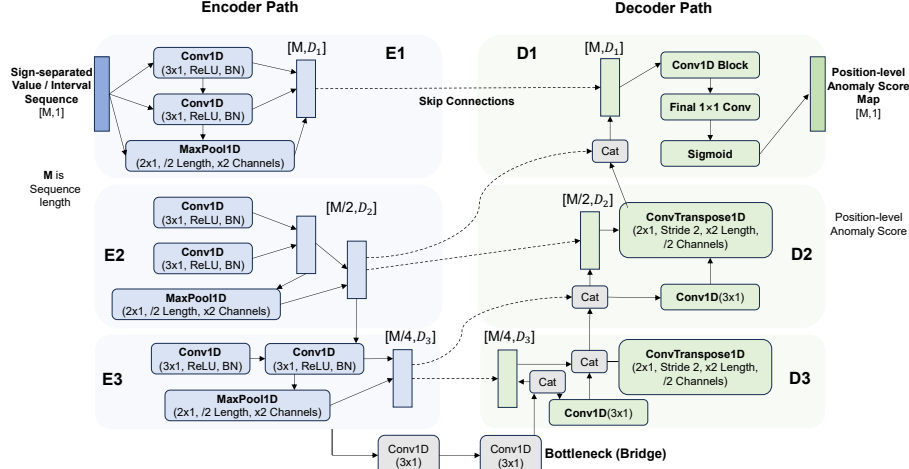


Figure 3. Architecture of the DF-PLNet encoder-decoder backbone.

The top-level loss-mixture weights are fixed in all reported experiments:

$$w_f = 1.0, \quad w_b = 1.0, \quad w_d = 1.0.$$

Class imbalance is handled inside the weighted BCE term at the batch level. For a batch with N_1 anomalous positions, N_0 benign positions, and $N = N_0 + N_1$, the BCE class weights are

$$\alpha_1 = \frac{N}{2 \max(N_1, 1)}, \quad \alpha_0 = \frac{N}{2 \max(N_0, 1)}.$$

Thus,

$$L_{\text{BCE}}^{(\text{weighted})} = -\frac{1}{N} \sum_{j=1}^N [\alpha_1 y_j \log(\hat{y}_j) + \alpha_0 (1 - y_j) \log(1 - \hat{y}_j)],$$

with probabilities clipped to avoid taking the logarithm of zero. Parameters are updated with Adam using a learning rate of 0.001. The network uses labels only after label-blind PFM and sequence construction have produced the training windows.

4.6 Train-threshold decision protocol

All methods follow the same train-threshold protocol. After a fixed 15-epoch run, the final checkpoint scores the training partition, and the binary operating threshold is selected by maximizing F1 on label-aligned value positions. The threshold is then frozen and applied once to the held-out test partition. Test labels are not used for threshold selection, checkpoint selection, early stopping, hyperparameter tuning, or model selection.

5 Experiments and Evaluation

This section evaluates DF-PLNet on supervised SWaT and HAI process-record splits, then reports main results, a representation sanity check, and PFM budget sensitivity.

5.1 Experimental setting

The experiments were run with our PyTorch implementation; configurations, seeds, and metric outputs were archived for each run. Each main row uses seeds 777, 778, and 779, and all $\pm$ values in this paper denote mean $\pm$ standard deviation (s.d.) over these three seeds. The fixed schedule uses 15 epochs, batch size 32, Adam with learning rate 0.001, weight decay 10^{-5} , gradient clipping at 1.0, and `segment_len=400`. The epoch budget is fixed in advance because validation-based early stopping would either consume labeled attack records or reintroduce a tuning path outside the train-only threshold protocol. The scheduler monitors training loss only.

Inference time is measured during the final test pass and reported as evaluated process-record positions per second. Because runtime depends on device, retained-stream size, and seed-level batching, speed is a diagnostic measurement rather than a hardware-independent efficiency claim.

5.2 Data construction and reproducibility

The experiments use two released process-record inputs. A preparation script converts each multivariate row to a scalar mean absolute z-score, estimated with normal training statistics, clipped at the 99th percentile of normal training values, scaled to [0, 1500], and normalized by the loader as in Equation (1). This yields one record-level value stream paired with timestamps.

SWaT (Secure Water Treatment) [28] instantiates the water-treatment running example introduced in Section 1. HAI (a hardware-in-the-loop augmented ICS dataset) [29] adds a second released process-record source with different imbalance and process-variable coverage.

SWaT process-record split. SWaT uses 51 numeric columns and a normal-training 99th percentile of 1.14218422. The prepared training file contains 1,387,097 records, including 27,310 attacks; the held-out test file contains 54,622 records, including 27,311 attacks and 27,311 non-attacks.

HAI process-record split. HAI uses 86 numeric columns and a normal-training 99th percentile of 0.920241462. The prepared training file contains `hai-train1-hai-train4` normal records plus labeled `hai-test1`, giving 950,400 training records with 2,981 attacks; labeled `hai-test2` gives 230,400 test records with 8,403 attacks.

Table 1 summarizes the prepared train/test splits.

We audited three potential leakage paths and confirmed that each is closed: scalar normalization uses normal training statistics only; PFM is applied independently after the train/test split and receives no labels; and thresholds are selected from training scores and labels only before being frozen for the held-out test file.

The prepared process-record protocol differs from common official-benchmark usage in its scalar input stream, chronological split definition, train-only threshold selection, and record-position evaluation set (Supplementary Table S1).

5.3 Model and baseline configuration

The input representation follows the sign-separated sequence construction in Section 4.2. Each retained process record contributes one interval position and one value position. Metrics are computed only on label-aligned value positions by the implementation’s position mask.

The two-dataset comparison includes locally implemented CNN, Transformer [30], DF-PLNet without PFM, and DF-PLNet variants. The CNN baseline probes local temporal feature extraction, the Transformer probes attention-based sequence modeling, DF-PLNet without PFM (abbreviated w/o PFM in tables) isolates the backbone, and the full DF-PLNet quantifies the added effect of Budgeted PFM.

Payload-dependent, protocol-field, native-feature, and unsupervised reconstruction pipelines require different inputs or objectives, so the matched comparison focuses on models that operate on the shared supervised value/interval stream. The detailed input-scope comparison is provided in Supplementary Table S2.

All baselines use the same prepared train/test files, `segment_len`, epoch budget, seed set, and train-only threshold protocol within each dataset.

5.4 Evaluation metrics

We report the area under the receiver operating characteristic curve (AUC), F1, precision, recall, the false positive rate (FPR), coverage-aware F1, and test speed on retained process-record positions. AUC summarizes ranking quality, F1 exposes the precision-recall trade-off, FPR measures alarm pressure, and speed measures evaluated record positions per second. PFM-enabled retained-stream metrics evaluate only records retained after periodic pruning and complete-window tail exclusion, so they must be interpreted under the stated drop budget.

For audit-enabled reruns, the implementation also records the precision-recall AUC (PR-AUC), retained and dropped complete-window counts, and coverage-aware confusion counts. The coverage-aware view treats a dropped attack record as a missed detection and a dropped benign record as a non-alarm benign record.

5.5 Main diagnostic results

Tables 2 and 3 report the supervised comparison under the same train-threshold protocol on both datasets. CNN, Transformer, and DF-PLNet without PFM use the unfiltered complete-window evaluation set. The main DF-PLNet row uses the pre-specified 10 % PFM budget, while 5 % and 20 % are reported as sensitivity settings in Section 5.8.

PFM changes the retained evaluation set, so Tables 2 and 3 report coverage-aware F1 (Cov. F1) next to retained-stream F1. Cov. F1 treats records removed by PFM as missed detections and equals retained-stream F1 for unfiltered rows; Table 4 gives the corresponding coverage audit.

5.6 Result analysis

On SWaT, DF-PLNet with 10 % PFM improves F1 over DF-PLNet without PFM (0.8966 vs. 0.8866), with higher recall but lower AUC and higher FPR. Coverage-aware F1 drops to 0.8429 because 2,868 attack positions are removed before scoring. The audit also shows that the SWaT drops are close to label-proportional (2,868 attack vs. 2,732 benign positions on a balanced

test split), so the 10 % budget does not enrich the retained stream with benign records; the retained-stream gain should be read as a scoring-focus effect whose coverage cost is quantified by the audit.

On HAI, 10 % PFM remains close to the no-PFM row in F1 and improves recall, but decreases precision and coverage-aware F1. Overall, DF-PLNet is competitive in this residual-stream setting when periodic pruning improves the retained stream without excessive coverage loss.

The contrast between SWaT and HAI is important for interpreting the method. SWaT has a balanced held-out test split and a clearer retained-stream benefit at mild pruning levels. HAI is more imbalanced, so each dropped attack position has a larger effect on coverage-aware F1. The practical operating point should therefore be selected from both retained-stream performance and the audit view, rather than from retained-stream F1 alone.

5.7 Representation sanity check

We checked the scalar process-deviation representation against raw FIT101, a single flow-sensor channel, under the same SWaT no-PFM protocol. The FIT101 run gives similar recall but much higher FPR, so its F1 and Matthews correlation coefficient (MCC) are lower; detailed values are provided in Supplementary Table S3.

5.8 PFM budget sensitivity

The key ablation compares DF-PLNet without PFM with fixed PFM budgets of 0 %, 5 %, 10 %, and 20 %. Detailed values are provided in Supplementary Table S4.

Figure 4 summarizes retained-stream F1, recall, coverage-aware F1, and attack retention across the four PFM budgets.

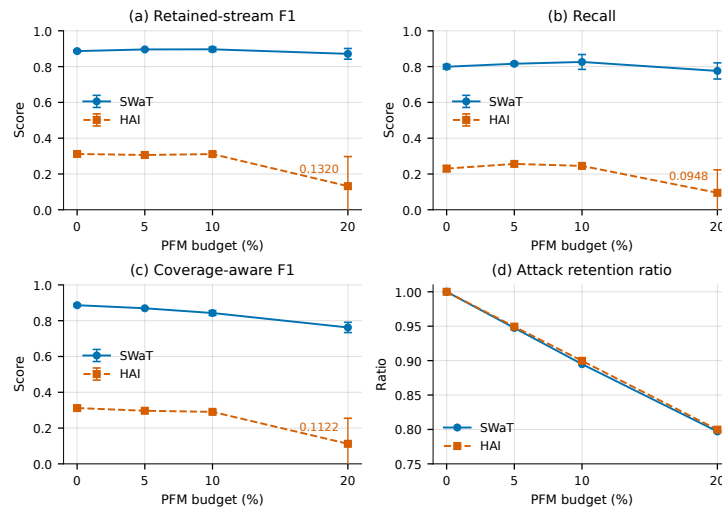


Figure 4. PFM budget sensitivity on SWaT and HAI.

On SWaT, 5 % pruning gives the highest AUC and nearly matches the 10 % F1 gain while retaining more records. On HAI, 5 % pruning gives the highest recall but lower F1, whereas 20 % pruning sharply reduces both. The budget study supports conservative pruning rather than treating PFM as automatically beneficial.

Dedicated audit runs clarify what the pruning budget removes. Table 4 reports retained and dropped complete-window positions together with coverage-aware F1.

On SWaT, 5 % pruning raises retained-stream F1 from 0.8866 to 0.8959 while coverage F1 decreases to 0.8694 because 1,434 attack positions are removed. HAI shows a stricter trade-off: at 10 % pruning, retained-stream F1 remains close to the no-PFM row, but coverage F1 decreases from 0.3118 to 0.2906; at 20 %, coverage F1 drops to 0.1122 on HAI and 0.7623 on SWaT.

6 Conclusion and Outlook

6.1 Summary

We have presented DF-PLNet, an offline residual-stream diagnostic framework for released ICS process-record analysis. Label-blind PFM prunes periodic-matching records within a conservative budget, and the backbone scores the retained sign-separated value/interval sequence. Under explicit SWaT and HAI splits, SWaT benefits from mild-to-moderate PFM budgets in retained-stream F1 and recall, whereas HAI requires more conservative budget selection because coverage-aware metrics degrade faster. The method should therefore be interpreted as an offline diagnostic aid rather than a streaming protection mechanism.

The main implication is procedural rather than universal: periodic pruning can be useful when it is label-blind, budgeted, and reported with all-record coverage. Without the coverage audit, retained-stream metrics can overstate the benefit of removing periodic records before scoring. DF-PLNet is therefore best viewed as a reproducible protocol for studying residual-stream diagnostics under controlled pruning budgets.

6.2 Limitations

Several limitations shape the interpretation of these results. Budgeted PFM does not improve all metrics simultaneously: SWaT gains retained-stream F1 and recall at 5–10 % pruning, while HAI is more budget-sensitive, so retained-stream and coverage-aware metrics must be considered together. The mean absolute z-score representation compresses multivariate process rows into one scalar; the SWaT single-feature sanity check supports this design choice but motivates multivariate encoders. The evidence comes from two released process-record datasets under protocol-compatible local splits, and additional public ICS datasets would strengthen transfer claims. Finally, the current PFM is offline; causal streaming PFM remains deployment-oriented future work. Future work should therefore replace fixed PFM budgets with validation-based selection, broaden representation studies, and evaluate causal streaming PFM with retained/drop masks.

Data availability

The raw SWaT and HAI datasets are controlled by their respective providers and are not redistributed with this article. The derived process-record datasets, preprocessing details, and experimental configurations are described in the manuscript; derived records and experimental artifacts are available from the corresponding author upon reasonable request, subject to the original dataset access conditions and institutional data-management requirements.

References

- [1] Pliatsios, D., Sarigiannidis, P., Lagkas, T. & Sarigiannidis, A. G. A survey on SCADA systems: Secure protocols, incidents, threats, and tactics. *IEEE Commun. Surv. & Tutor.* **22**, 1942–1976, DOI: [10.1109/COMST.2020.2987688](https://doi.org/10.1109/COMST.2020.2987688) (2020).
- [2] Karnouskos, S. Stuxnet worm impact on industrial cyber-physical system security. In *Proceedings of the 37th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 4490–4494, DOI: [10.1109/IECON.2011.6120048](https://doi.org/10.1109/IECON.2011.6120048) (2011).
- [3] Johnson, B. *et al.* Attackers deploy new ICS attack framework “TRITON” and cause operational disruption to critical infrastructure. <https://cloud.google.com/blog/topics/threat-intelligence/attackers-deploy-new-ics-attack-framework-triton/> (2017). Mandiant / Google Cloud Threat Intelligence blog, accessed 2026-07-01.
- [4] Liu, J. *et al.* Deep anomaly detection in packet payload. *Neurocomputing* **485**, 205–218, DOI: [10.1016/j.neucom.2021.01.146](https://doi.org/10.1016/j.neucom.2021.01.146) (2022).
- [5] Caselli, M., Zamboni, E. & Kargl, F. Sequence-aware intrusion detection in industrial control systems. In *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security (CPSS '15)*, 13–24, DOI: [10.1145/2732198.2732200](https://doi.org/10.1145/2732198.2732200) (ACM, New York, 2015).
- [6] Mirsky, Y., Doitshman, T., Elovici, Y. & Shabtai, A. Kitsune: An ensemble of autoencoders for online network intrusion detection. In *Proceedings of the Network and Distributed System Security Symposium (NDSS 2018)*, 1–15, DOI: [10.14722/ndss.2018.23204](https://doi.org/10.14722/ndss.2018.23204) (Internet Society, 2018).
- [7] Hao, W. *et al.* Unsupervised real-time communication traffic anomaly detection for multi-dimensional industrial networks. *IEEE Trans. Ind. Cyber-Phys. Syst.* **3**, 228–240, DOI: [10.1109/TICPS.2024.3524185](https://doi.org/10.1109/TICPS.2024.3524185) (2025).
- [8] Schuster, F. & König, H. No need for details: Effective anomaly detection for process control traffic in absence of protocol and attack knowledge. In *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2024)*, 278–297, DOI: [10.1145/3678890.3678932](https://doi.org/10.1145/3678890.3678932) (ACM, 2024).
- [9] Stouffer, K. *et al.* Guide to operational technology (OT) security. Special Publication 800-82 Rev. 3, National Institute of Standards and Technology (2023). DOI: [10.6028/NIST.SP.800-82r3](https://doi.org/10.6028/NIST.SP.800-82r3).
- [10] Kuchar, K. & Fajdiak, R. Anomaly detection in industrial networks: Current state, classification, and key challenges. *IEEE Sens. J.* **25**, 5031–5043, DOI: [10.1109/JSEN.2024.3512857](https://doi.org/10.1109/JSEN.2024.3512857) (2025).
- [11] Nyasore, O. N., Zavorsky, P., Swar, B., Naiyeju, R. & Dabra, S. Deep packet inspection in industrial automation control system to mitigate attacks exploiting Modbus/TCP vulnerabilities. In *2020 IEEE 6th International Conference on Big Data Security on Cloud (BigDataSecurity)*, 241–245, DOI: [10.1109/BigDataSecurity-HPSC-IDS49724.2020.00051](https://doi.org/10.1109/BigDataSecurity-HPSC-IDS49724.2020.00051) (2020).
- [12] Feng, C., Palleti, V. R., Mathur, A. & Chana, D. A systematic framework to generate invariants for anomaly detection in industrial control systems. In *Proceedings of the Network and Distributed System Security Symposium (NDSS 2019)*, 1–15, DOI: [10.14722/ndss.2019.23265](https://doi.org/10.14722/ndss.2019.23265) (Internet Society, 2019).
- [13] Fu, C., Li, Q., Shen, M. & Xu, K. Detecting tunneled flooding traffic via deep semantic analysis of packet length patterns. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS '24)*, 3659–3673, DOI: [10.1145/3658644.3670353](https://doi.org/10.1145/3658644.3670353) (ACM, 2024).
- [14] Wang, W., Zhu, M., Wang, J., Zeng, X. & Yang, Z. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 43–48, DOI: [10.1109/ISI.2017.8004872](https://doi.org/10.1109/ISI.2017.8004872) (2017).
- [15] Goh, J., Adepu, S., Tan, M. & Lee, Z. S. Anomaly detection in cyber physical systems using recurrent neural networks. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, 140–145, DOI: [10.1109/HASE.2017.36](https://doi.org/10.1109/HASE.2017.36) (2017).

- [16] Kravchik, M. & Shabtai, A. Efficient cyber attack detection in industrial control systems using lightweight neural networks and PCA. *IEEE Trans. Dependable Secure Comput.* **19**, 2179–2197, DOI: [10.1109/TDSC.2021.3050101](https://doi.org/10.1109/TDSC.2021.3050101) (2022).
- [17] Zhan, D. *et al.* Anomaly detection in industrial control systems based on cross-domain representation learning. *IEEE Trans. Dependable Secure Comput.* **22**, 2505–2518, DOI: [10.1109/TDSC.2024.3520155](https://doi.org/10.1109/TDSC.2024.3520155) (2025).
- [18] Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 234–241, DOI: [10.1007/978-3-319-24574-4_28](https://doi.org/10.1007/978-3-319-24574-4_28) (Springer, 2015).
- [19] Perslev, M., Jensen, M. H., Darkner, S., Jennum, P. J. & Igel, C. U-Time: A fully convolutional network for time series segmentation applied to sleep staging. In *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 4415–4426 (2019).
- [20] Audibert, J., Michiardi, P., Guyard, F., Marti, S. & Zuluaga, M. A. USAD: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 3395–3404, DOI: [10.1145/3394486.3403392](https://doi.org/10.1145/3394486.3403392) (ACM, 2020).
- [21] Chou, D. & Jiang, M. A survey on data-driven network intrusion detection. *ACM Comput. Surv.* **54**, 182:1–182:36, DOI: [10.1145/3472753](https://doi.org/10.1145/3472753) (2021).
- [22] Han, D. *et al.* Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors. *IEEE J. Sel. Areas Commun.* **39**, 2632–2647, DOI: [10.1109/JSAC.2021.3087242](https://doi.org/10.1109/JSAC.2021.3087242) (2021).
- [23] Barbosa, R. R. R., Sadre, R. & Pras, A. Towards periodicity based anomaly detection in SCADA networks. In *17th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–4, DOI: [10.1109/ETFA.2012.6489745](https://doi.org/10.1109/ETFA.2012.6489745) (2012).
- [24] Lin, X. *et al.* A real-time anomaly detection method for industrial control systems based on long-short period deterministic finite automaton. *IEEE Internet Things J.* **12**, 14599–14621, DOI: [10.1109/JIOT.2025.3526599](https://doi.org/10.1109/JIOT.2025.3526599) (2025).
- [25] Wen, Q., Yang, L. & Sun, L. Robust dominant periodicity detection for time series with missing data. In *ICASSP 2023 – 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5, DOI: [10.1109/ICASSP49357.2023.10095241](https://doi.org/10.1109/ICASSP49357.2023.10095241) (2023).
- [26] Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2980–2988, DOI: [10.1109/ICCV.2017.324](https://doi.org/10.1109/ICCV.2017.324) (2017).
- [27] Milletari, F., Navab, N. & Ahmadi, S.-A. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, 565–571, DOI: [10.1109/3DV.2016.79](https://doi.org/10.1109/3DV.2016.79) (2016).
- [28] Goh, J., Adepu, S., Junejo, K. N. & Mathur, A. A dataset to support research in the design of secure water treatment systems. In *Critical Information Infrastructures Security (CRITIS 2016)*, vol. 10242 of *Lecture Notes in Computer Science*, 88–99, DOI: [10.1007/978-3-319-71368-7_8](https://doi.org/10.1007/978-3-319-71368-7_8) (Springer, 2017).
- [29] Shin, H.-K., Lee, W., Yun, J.-H. & Kim, H. HAI 1.0: HIL-based augmented ICS security dataset. In *13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20)* (USENIX Association, 2020).
- [30] Vaswani, A. *et al.* Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 5998–6008 (2017).

Figure legends

Figure 1. Overview of the DF-PLNet pipeline. The figure summarizes how DF-PLNet converts released ICS process records into anomaly decisions. Phase I shows label-blind budgeted periodic filtering applied to the chronological record stream. Phase II shows serialization of each retained record into interval and value positions. Phase III shows position-level scoring with the encoder–decoder backbone. Phase IV shows threshold-based conversion of scores into final record-position decisions.

Figure 2. Workflow of the Budgeted Periodic Filtering Module. The figure shows the main stages of budgeted periodic filtering before residual-stream diagnosis. Step 1 shows aggregation of record counts in fixed windows. Step 2 shows smoothing of the aggregated sequence. Step 3 shows quantization into symbols. Step 4 shows multi-start period estimation and dominant-period selection. Step 5 shows bounded pruning of periodic candidates under the preset drop budget.

Figure 3. Architecture of the DF-PLNet encoder–decoder backbone. The figure shows the U-shaped network used for residual-stream anomaly scoring. The encoder path illustrates successive convolution and pooling blocks that expand temporal context. The bridge shows the bottleneck representation at the coarsest scale. The decoder path shows transposed convolutions and skip connections that restore position-level resolution. The output layer shows the final per-position anomaly scores.

Figure 4. PFM budget sensitivity on SWaT and HAI. The figure compares retained-stream and coverage-aware trends under different pruning budgets. Panel (a) shows retained-stream F1 across the 0 %, 5 %, 10 %, and 20 % budgets. Panel (b) shows recall across the same budgets. Panel (c) shows coverage-aware F1, highlighting the cost of dropped attack records. Panel (d) shows attack retention ratio, indicating how much anomalous coverage remains after pruning. Error bars indicate the standard deviation across random seeds.

Funding statement

This work was supported by the Beijing Natural Science Foundation under Grant Nos. 4264102 (Youth Program) and L233035; the Youth Research Special Project of North China University of Technology (NCUT) under Project No. 110051360025XN077-16; the R&D Program of Beijing Municipal Education Commission under Project No. 110052972508-11; and the Scientific Research Foundation of NCUT.

Author contributions statement

Y.Z. contributed methodology, software, validation, and original drafting. J.C. contributed conceptualization, investigation, data curation, and review and editing. Y.L. contributed resources, investigation, and review and editing. Y.S. contributed supervision, methodology, and review and editing. All authors reviewed the manuscript.

Additional information

Competing interests: All authors declare no competing interests.

Correspondence: Correspondence and requests for materials should be addressed to J.C. (chenjiahui@mail.ncut.edu.cn).

AI use statement: An AI language model assisted with language editing, formatting checks, and LaTeX organization. The authors reviewed and verified all scientific content, analyses, citations, and conclusions.

Tables

Table 1. Prepared SWaT and HAI process-record splits.

Dataset	Split	Total records	Attack records	Attack rate	Complete-window eval. records
SWaT	Train	1,387,097	27,310	0.0197	1,387,000
SWaT	Test	54,622	27,311	0.5000	54,600
HAI	Train	950,400	2,981	0.0031	950,200
HAI	Test	230,400	8,403	0.0365	230,200

Table 2. SWaT retained process-record metrics under the matched train-threshold protocol. Values are mean $\pm$ s.d. over seeds 777, 778, and 779; speed is reported in evaluated record positions per second.

Method	AUC	F1	Cov. F1	Prec.	Rec.	FPR	Speed
CNN	0.964 $\pm$ 0.003	0.881 $\pm$ 0.011	0.881 $\pm$ 0.011	0.999 $\pm$ 0.000	0.787 $\pm$ 0.017	0.001 $\pm$ 0.000	238k
Transformer	0.976 $\pm$ 0.003	0.887 $\pm$ 0.004	0.887 $\pm$ 0.004	1.000 $\pm$ 0.000	0.797 $\pm$ 0.006	0.000 $\pm$ 0.000	42k
DF-PLNet w/o PFM	0.967 $\pm$ 0.001	0.887 $\pm$ 0.007	0.887 $\pm$ 0.007	0.996 $\pm$ 0.003	0.799 $\pm$ 0.012	0.004 $\pm$ 0.002	97k
DF-PLNet	0.951 $\pm$ 0.015	0.897 $\pm$ 0.012	0.843 $\pm$ 0.013	0.983 $\pm$ 0.028	0.826 $\pm$ 0.042	0.016 $\pm$ 0.025	67k

Table 3. HAI retained process-record metrics under the matched train-threshold protocol. Values are mean $\pm$ s.d. over seeds 777, 778, and 779; speed is reported in evaluated record positions per second.

Method	AUC	F1	Cov. F1	Prec.	Rec.	FPR	Speed
CNN	0.650 $\pm$ 0.003	0.291 $\pm$ 0.037	0.291 $\pm$ 0.037	0.430 $\pm$ 0.179	0.241 $\pm$ 0.037	0.017 $\pm$ 0.017	297k
Transformer	0.645 $\pm$ 0.013	0.143 $\pm$ 0.111	0.143 $\pm$ 0.111	0.250 $\pm$ 0.009	0.134 $\pm$ 0.152	0.016 $\pm$ 0.018	29k
DF-PLNet w/o PFM	0.647 $\pm$ 0.000	0.312 $\pm$ 0.011	0.312 $\pm$ 0.011	0.488 $\pm$ 0.030	0.230 $\pm$ 0.017	0.009 $\pm$ 0.002	80k
DF-PLNet	0.649 $\pm$ 0.003	0.311 $\pm$ 0.009	0.291 $\pm$ 0.008	0.428 $\pm$ 0.042	0.245 $\pm$ 0.005	0.013 $\pm$ 0.002	80k

Table 4. PFM coverage audit on SWaT and HAI. F1 values are mean $\pm$ s.d. over seeds 777, 778, and 779; retained and dropped counts are deterministic per budget.

Dataset	Budget	Ret. eval.	Drop atk.	Drop ben.	Atk. ret.	Ret. F1	Cov. F1
SWaT	0 %	54,600	0	0	1.0000	0.8866 $\pm$ 0.0067	0.8866 $\pm$ 0.0067
SWaT	5 %	51,800	1,434	1,366	0.9475	0.8959 $\pm$ 0.0037	0.8694 $\pm$ 0.0037
SWaT	10 %	49,000	2,868	2,732	0.8949	0.8966 $\pm$ 0.0122	0.8429 $\pm$ 0.0133
SWaT	20 %	43,600	5,537	5,463	0.7971	0.8713 $\pm$ 0.0300	0.7623 $\pm$ 0.0286
HAI	0 %	230,200	0	0	1.0000	0.3118 $\pm$ 0.0106	0.3118 $\pm$ 0.0106
HAI	5 %	218,800	425	10,975	0.9494	0.3061 $\pm$ 0.0054	0.2967 $\pm$ 0.0050
HAI	10 %	207,200	843	22,157	0.8997	0.3112 $\pm$ 0.0091	0.2906 $\pm$ 0.0079
HAI	20 %	184,200	1,683	44,317	0.7997	0.1320 $\pm$ 0.1653	0.1122 $\pm$ 0.1426

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [supplementaryinformation.pdf](#)