

An Empirical Evaluation of COUTA Time-Series Anomaly Detection under Different Evaluation Protocols

Feiyang Teng*

School of Cyber Engineering, Xidian University, Xi'an, Shaanxi 710126, China

**Corresponding author: Feiyang Teng.*

Abstract

Experimental conclusions in time-series anomaly detection depend not only on the model but also on threshold selection and evaluation protocols. Taking COUTA as the object, without changing the model structure, this research trains and tests on SMAP, MSL, and PSM separately, compares original point-wise evaluation, Point Adjustment (PA), and Train-Q99 alarm results in which the threshold is determined only by the 99th percentile of valid training scores, and also conducts sensitivity experiments on PSM at 5 synthetic training contamination levels. Every group of experiments employs five model random seeds and reports the mean and population standard deviation. Results of this paper show that PA-F1 on SMAP and MSL is 0.7547 and 0.8910, respectively, whereas Train-Q99 F1 is only 0.0130 and 0.0933; the Train-Q99 F1 on PSM is 0.4568, but the normal-point false positive rate reaches 0.1844. From 0% to 10% additional contamination, PSM's Oracle Raw F1 decreases from 0.5476 to 0.4515, while PA-F1 increases from 0.8849 to 0.9470. This research establishes a traceable evaluation result set across multiple datasets and seeds, quantifies the gap between offline optimal evaluation, and fixed-threshold alarming, and reveals the directional divergence between point-wise evaluation and PA evaluation under training contamination. The results suggest that COUTA's performance and contamination robustness are supposed to be interpreted jointly with raw scores, threshold information sources, and alarm burden. Differences in valid evaluation positions among protocols and preprocessing conditions are clarified in the main text.

Keywords

time-series anomaly detection, COUTA, empirical evaluation, point adjustment, threshold calibration, training contamination

1. Introduction

Multivariate time series can be used to model the dynamic changes in the state of systems and act as a critical basis of data on which server monitoring, space shuttle telemetry, and industrial fault alerts are based. The time-series anomaly detection detects the outlier values or successive periods that are not typical in the usual way, and it forms the ground of identifying any operational threats. Deep learning studies have established technical directions in forecasting, reconstruction, representation learning, and one-class classification, and in recent years have gone on to evolve self-supervised learning and pretrained models [1, 2]. Nonetheless, the issue that the detection outcomes cannot be compared consistently has not been addressed by the constant revisions of the network structures. The review based on a process reveals that the data

organization, score generation, method choice, and evaluation measures influence the ultimate performance; and being a leader in a single indicator is not enough to address all these aspects [3].

Evaluation reliability has thus been a subject of research as critical as algorithm design. Data integrity, metric reliability and the unified experimental pipeline feature are integrated into the benchmark construction process of TSB-AD; TAB additionally organizes unified comparisons based on multivariate and univariate data; mTSBench in 2026 looks at the multivariate anomaly detectors and unsupervised model selectors concurrently, and reports that the performance of the methods depends on the dataset, and that there is a gap between the outcomes of the model selection and the Oracle reference [4-6]. The experiments reveal that model performance on the new data must be experimentally verifiable and must differentiate detection capabilities related to data applicability, threshold calibration, and evaluation rules.

The paper chooses the calibrated one-class classification algorithm COUTA as the research subject. Training data may be unsupervised, which can have unlabeled anomalies, and such contaminated examples can hinder normal-pattern estimation. COUTA reduces the influence of possible contamination by calibration through an uncertainty model and constructs auxiliary anomalies via native anomaly calibration, which gives learning signals to separate normal and anomalous behavior [7]. Because the ability to tolerate contamination is a critical design goal, it is appropriate to use COUTA both to test the feasibility of using a fixed configuration on various data sets and to investigate whether the evaluation procedures lead to different conclusions regarding the strength of contamination resistance.

COUTA paper The original paper uses the Point Adjustment (PA) method as the major assessment measure and substitutes the scores of each ground-truth anomaly interval with the highest score of the interval and chooses the threshold that produces the highest F1 using test labels [7]. PA extends the detection credit that is earned through the local high-score hits, whereas the best-threshold search engine adds test-label information, and the two alter the correct-detection criterion and the alarm-threshold-selection condition, respectively. Kim et al. showed that PA has a potential risk of performance overestimation, which was later explored in more recent works on Balanced Point Adjustment based on the concept of fairness requirements [8, 9]. Thus, it is necessary to conduct an independent analysis of whether a high PA-F1 means that enough anomaly points are covered and if the offline best F1 can be transformed into useful alerts at a threshold without test labels.

The discrepancies in the evaluation cannot be resolved by choosing only one of the so-called “correct” metrics. Metric taxonomy studies differentiate between goals like point-wise coverage, event hits, and time tolerance; PATE and VUS enhance the assessment through temporal proximity and interval information, and more recent formal studies indicate trade-offs between metric properties [10-13]. At the same time, work on online confidence-sequence thresholds and multi-system calibrated thresholds demonstrates that deriving alarm results on continuous anomaly scores is also a separate modeling problem [14, 15]. It is hence important to consider the combined examination of score quality, threshold effectiveness, and interval adjustment when studying models with anti-contamination designs, instead of focusing on robustness based on a single offline protocol.

The current paper, according to the discussed issues, trains and tests COUTA on SMAP, MSL, and PSM independently, maintains the primary model structure, and compares Raw ranking measures, Oracle Raw F1 using thresholds determined by test labels, PA-modified measures, and Train-Q99 alarm findings. The Train-Q99 defines the threshold based on the 99th percentile of the valid training anomaly scores, without considering the test labels when choosing the threshold, and gives point-wise precision, recall, event recall, and normal-point false positive rate. Also, the paper considers five synthetic contamination degrees in the PSM training data, preserves the original test data and labels, and investigates if Raw and PA provide the same verdicts regarding the change in performance. Every set of experiments is done with five model random seeds, and the results are saved per run, and mean and standard deviations are computed.

The main contributions of this paper are as follows.

(1) The multi-seed evaluation outcome of COUTA on three public datasets is created, which explicitly connects the data representation, model configuration, score recording, and statistical conventions, measures the differences in detection performance at each point of measurement between two datasets with a fixed configuration, and offers a trackable reference to be used in future method research.

(2) The gap between offline optimal evaluation and fixed threshold alarming without test labels is clear-cut. By concurrently analyzing Raw, PA, and Train-Q99, event recall and false positive rate, inconsistencies in event hits, anomaly-duration coverage, and alarm burden, operational characteristics which a single F1 cannot possibly show are supplemented.

(3) The directional divergence of the evaluation conclusions in the case of PSM training contamination is uncovered (at no additional contamination and at 10% synthetic contamination, Oracle Raw F1 reduces by 9.21 percent (0.5476/0.4515) and PA-F1 increases by 6.21 percent (0.8849/0.9470)). This finding indicates that using only the PA evaluation does not necessarily indicate a loss of point-wise performance, which gives direct experimental proof of multi-protocol reporting of contamination robustness.

The present paper is an empirical paper based on an already existing model wherein every single dataset was separately trained and tested. The experimental set-up and limitations of the study explain uniformly the applicability of the conditions of the conclusions and the disparity in proper assessment points between protocols.

2. Related Work

2.1 Anomaly Representation Learning and Normal-Pattern Modeling

The fundamental distinction between time-series anomaly detection approaches is based on what normal patterns are defined and how deviations can be transformed into anomaly scores. Deep learning surveys systematically structure the associations between various network architectures, learning contexts, and types of anomalies; self-supervised learning surveys also separate self-prediction, contrastive learning, and their combinations, highlighting the design of training tasks when there are no anomaly labels [1, 2]. This research direction demonstrates that the choice of a method entails both model capacity as well as normal pattern representation and synthetic anomaly prior and assumption about time dependence.

DCdetector, in self-supervised representation learning, is a learning model that acquires discriminative representations through multi-scale dual attention and a contrastive objective; TriAD simultaneously considers temporal, frequency, and residual views and is sensitive to the difference in the duration and morphology of anomalies [16, 17]. TFMAE relies on temporal and frequency masked reconstruction and contrastive learning to mitigate anomalous training bias effects and distribution shifts; Dual-TF matches anomaly-score granularity in time and frequency domains by nested sliding windows to allow more precise representation of interval boundaries [18, 19]. These techniques increase the number of modeling techniques of anomaly representation and it is also suggested that the relationship between window granularity and point-wise evaluation has to be made clear.

The research conducted in 2024-2025 has placed even more emphasis on the concept of variable associations and continuous-time structure. SARAD assesses relational changes due to anomalies that occur in the space of variable-associations; CATCH acquires correct channel associations in frequency patches; ScatterAD simultaneously describes the temporal and topological representations in order to depict the scattering properties of samples [20-22]. TSINR employs implicit neural representations to describe temporal continuity, and KAN-AD acquires local normal patterns using Kolmogorov-Arnold networks and the associated function representations [23, 24]. The stated works reflect various structural presumptions, and their library outcomes cannot be compared directly in the horizontal way without paying heed to the initial data edition and assessment laws.

Pretrained models and generative models are other technical avenues that can be used in anomaly detection. ImDiffusion is a combination of the missing value imputation and diffusion models that leverage temporal and inter-variable correlations to create an anomaly detection pipeline; AnomalyLLM distills the detection data based on knowledge distillation between a large language model teacher and a small student network; MOMENT constructs open pretrained models of various kinds of time-series tasks [25-27]. The present research will consider these as the background of the research and will focus on the assessment of the performance of the fixed COUTA model rather than developing a unified baseline table by using their initially reported values.

2.2 Benchmark Construction and Multi-Dataset Empirical Studies

Comparisons based on reliable empirical evidence need to control data processing, hyperparameters, score alignment, and statistical summaries. Mejri et al. evaluate a wide range of unsupervised time-series anomaly detection methods and report results both with and without PA and assess the stability of repeated runs; Sarfraz et al. note the effect of benchmark practices and evaluation options on the credibility of performance claims [28, 29]. So, multiple runs and multi-metric reporting form the basis of the empirical study, and the research value is determined by their ability to respond to a specific detection question.

The recent benchmark studies have extended the list of evaluated objects and analysis dimensions. TSB-AD implements method comparison beginning with data quality and metric reliability; TAB supports the analysis of various models and data types via a common pipeline; mTSBench additionally assesses multivariate detectors along with unsupervised model selection [4-6]. The works offer wide horizontal evidence and also imply that the behavior of a particular model should be understood regarding the representation of its data and selection conditions. The research design used in this paper is complementary: fixing the model body, linking point-wise results, threshold-alarm results, and training contamination results on three datasets, and concentrating on discussing the meaning of performance of the same detection pipeline under various evaluation circumstances.

2.3 Point Adjustment and Temporal-Structure Evaluation

The point-wise measures assess the accuracy of the prediction at every time point, although the anomaly events tend to last more than a few time units, and the business goals might aim at first detection, continuous monitoring, or the detection period. The metric taxonomy of Sorbo and Ruocco demonstrates that these objectives can be related to various assessment preferences [10]. PA assumes an extension of one hit into a ground truth anomaly interval to the whole interval, which could be considered as a particular kind of event-discovery assumption, and the loss associated with missing a detection during the event is mitigated. Kim et al. note that such a mechanism could increase the estimated performance of the detection and even alter the ranking of methods; the interval-maximum score substitution used in COUTA is a particular version of this correction [7, 8].

In terms of how temporal structure is treated during an evaluation, the Affiliation metric measures the quality of detection based on the local distance between predicted sets and ground-truth events, PATE takes into account early and late detection by way of proximity-buffer weighting, and VUS is a summary measure of performance at threshold and interval-buffer configurations [11, 12, 30]. The Balance Between True Positives and False Positives in the Adjustment Process Balanced Point Adjustment also examines the balance between true positives and false positives in the adjustment process, whereas Wagner et al. formally examine the characteristics of 37 measurements and demonstrate that a particular measurement option cannot be applied without condition [9, 13]. Altogether, these works are consistent in supporting reporting point-wise, event-related, and time-related findings independently, which does not imply that each study should necessarily calculate all the metrics.

The paper is based on the initial COUTA PA implementation, which serves as a control, maintaining score assessment without changing intervals, and adds interpretation to the evaluation based on event recall and false positives at a constant level. The study concentrates on how current evaluation principles influence the performance assessments of COUTA, in particular if Raw and PA offer contradicting results of training contamination effects.

2.4 Threshold Calibration and Practical Alarm Objectives

The quality of ranking anomaly scores and the effectiveness of alarms once a threshold is chosen are not the same issues. The Oracle F1 uses the test labels to find the optimal threshold and can be used to describe the ex post optimal operating point of the provided scores; practical alarming requires determining an operating threshold when the future labels are unknown. Sun et al. apply confidence sequences to examine online adaptive thresholds, and M²AD integrates global scores with statistically calibrated thresholds to address heterogeneity in multi-sensor and multi-system environments [14, 15]. The calibration procedure is regarded as a significant element of the detection system by both types of studies.

The training-score Q99 is used in this paper to create a repeatable fixed-threshold reference and is reported with Oracle results. The given design may indicate the difference between a situation where “scores are distinguishable” and “the current threshold is usable.” The Q99 does not have either independent calibration set or adaptive updating and does not provide the statistical assurance like the other methods mentioned above; its utility is that it explains the source of information behind the threshold and explains various facets of the same operating point by means of point-wise recall, event hits, and false alarms load.

2.5 Training Contamination Robustness and Research Positioning

Training anomalies may be unlabeled contamination which has to be resisted or artificial creation of auxiliary learning signals, which are two concepts that ought not to be mixed up. NCAD assists in learning the boundary between normal and anomalous through introducing synthetic anomalies, whereas COUTA integrates native anomaly calibration with uncertainty modeling calibration to manage insufficient anomaly information as well as possible contamination in one-class training [7, 31]. TSAD-C is based on a spatio-temporal graph conditional diffusion model, combining data purification and variable-dependency modeling, as an alternative solution that considers contamination simultaneously at data and representation levels [32].

The change in data scenarios is also not the same as cross-domain adaptation. DACAD learns cross-domain representations by using source-domain labels and target-domain unlabeled data, based on contrastive learning and anomaly injection, and specifically constructs an information transfer flow between the source domain and the target domain [33]. The current paper is trained on one dataset at a time and examines the applicability of a fixed method to various data, not the performance of the transfer-learning. This difference maintains the consistency of the experimental object and the supportable conclusions.

The current literature has already uncovered PA bias, threshold difficulty, and training contamination threat; this paper also applies these problems to the same COUTA evaluation pipeline. The research measures gaps in protocols by using three-dataset, multi-seed results and demonstrates the direction of the divergence between Raw and PA by the PSM contamination experiment, which serves as empirical evidence of performance interpretation and reporting of its robustness. In contrast to general horizontal algorithm rankings, the present paper addresses the question of what evaluation conditions mask or uncover performance changes of this detection pipeline.

3. Research Object and Evaluation Methods

The present paper uses the full COUTA model as the research subject and performs the empirical analysis of multi-dataset performance, the differences in evaluation protocols, and the sensitivity to training contamination. The model structure and learning objectives do not change during the course of the research, but the study clearly defines three steps, i.e. generation of anomaly scores, setting of the threshold, and calculation of metrics, and compares the outcomes and limits of interpretations under various evaluation conditions. This chapter presents the model, evaluation protocols and evaluation pipeline; particular data and experimental setups are provided in Chapter 4. Figure 1 depicts the overall evaluation pipeline.

3.1 Couta Model and Anomaly Scores

Let the multivariate time series be $X=(\mathbf{x}_1, \dots, \mathbf{x}_T)$, where $\mathbf{x}_t \in \mathbb{R}^d$. For window length L , the input subsequence corresponding to time t is

$$W_t=(\mathbf{x}_{t-L+1}, \dots, \mathbf{x}_t), \quad t \geq L. \quad (1)$$

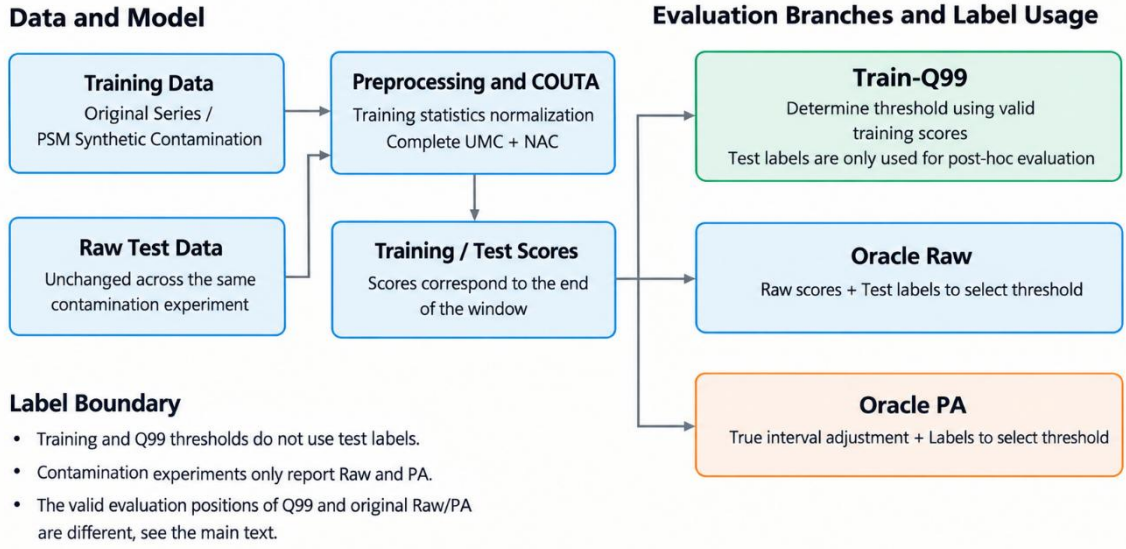
COUTA uses a temporal convolutional network to extract window representations and applies one-class learning to identify a small area of normal data within the representation space. To reduce the effects of possible contaminated samples on normality learning, uncertainty modeling calibration (UMC) is employed; native anomaly calibration (NAC) creates supplementary samples by perturbing training windows, offering learning signals that differentiate between normality and anomaly. This paper allows both UMC and NAC and has no module addition/removal or structure enhancement [7].

Based on the implementation of the experiment, the two sets of window representations are represented by the symbols $\mathbf{z}_t$ and $\mathbf{z}'_t$, the one-class center is represented by the symbol $\mathbf{c}$, and the anomaly score is

$$s_t = \|z_t - c\|_2^2 + \|z'_t - c\|_2^2. \quad (2)$$

The higher the score, the more extreme is the deviation of the window relative to the normal area that was trained by the model. The score of every window is relative to the endpoint time. Since there are no complete windows at the first $L-1$ time points, the program fills these locations with zeros; these scores are not valid inference outcomes of full windows, and the way they are processed is described in Section 3.2.

Figure 1: COUTA empirical evaluation pipeline. Training data are used for preprocessing parameter estimation and model training; model outputs enter raw-score evaluation, Point Adjustment evaluation, and Train-Q99 alarm evaluation, respectively. Test labels are used only in Oracle threshold selection, PA adjustment, and ex post metric computation, and do not participate in Train-Q99 threshold determination. Contamination experiments compare only Raw and PA results



3.2 Evaluation Protocols and Label-Use Boundaries

The paper identifies three evaluation environments: Oracle Raw, Point-adjusted, and Train-Q99. The first two are employed to analyze the offline performance, and the third to investigate a fixed alarm rule which does not rely on test labels to define thresholds. The three are not just substitutes for metric names but also vary in terms of the source of the threshold, the way the score is processed, and the time interval between evaluations.

(1) Oracle Raw. With test anomaly scores being held constant, the threshold that optimises F1 is chosen among candidate thresholds based on test labels:

$$F_{1,Raw}^* = \max_{\tau \in T} F_1(y, I[s \geq \tau]). \quad (3)$$

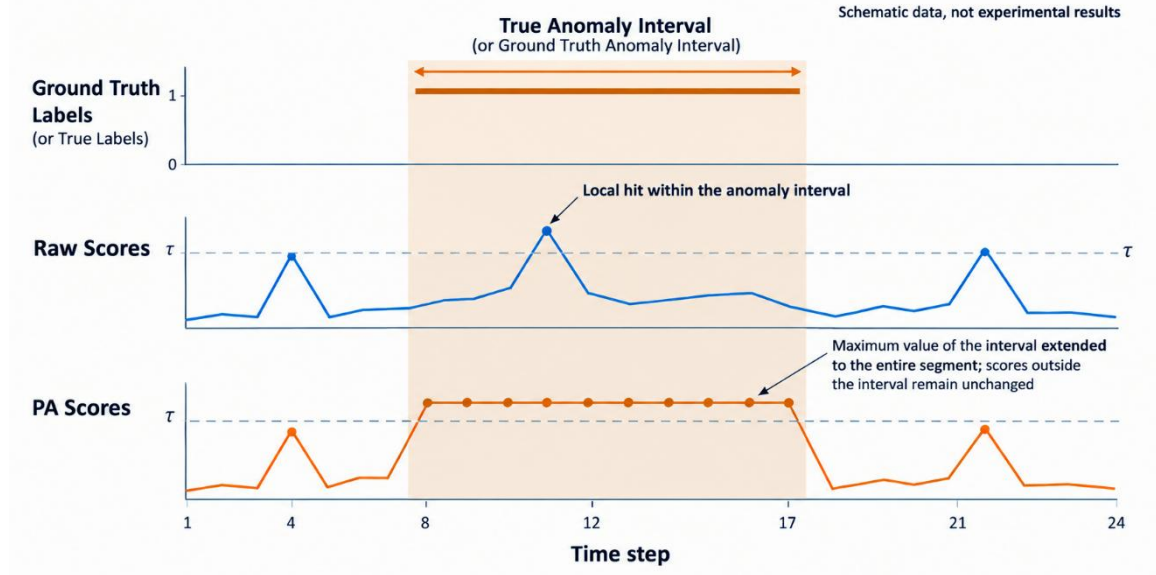
In this case, $\mathcal{T}$ is the candidate threshold that has been set based on test scores. The outcome is an ex-post optimal reference with regard to the score and the range of the candidate thresholds; it is neither an alarm performance achievable without the test labels, nor is it a hypothetical limit to the future data.

(2) Point adjusted. Assume that the series of anomaly intervals in the test labels is $A_1, \dots, A_K$. As in the original score adjustment procedure of COUTA, the anomaly scores of points in an interval are substituted by the largest value of that interval [7]:

$$s_t^{PA} = \begin{cases} \max_{u \in A_k} s_u, & t \in A_k, \\ s_t, & t \notin \bigcup_{k=1}^K A_k. \end{cases} \quad (4)$$

Next, AUROC, AUPR and optimal F1 are calculated based on the modified scores. In the case of a fixed threshold, this operation is the same as extending a single hit into an anomaly period to the whole anomaly period. Hence, PA outcomes should not be read directly as the capability of initial forecasts to cover the duration of anomalies; PA-F1 does not equal event-based F1 that assumes the same weight on every anomalous event. The described mechanism of score adjustment is presented in Figure 2.

Figure 2: Schematic of evaluation differences between raw scores and Point Adjustment. Within the same ground-truth anomaly interval, raw scores may exceed the threshold at only a few positions; PA unifies the interval scores to their maximum. The same threshold is used in the figure only to illustrate the adjustment mechanism and does not represent actual numerical values from the experiments in this paper



(3) Train-Q99. The 99th percentile of the valid training anomaly score set is calculated for the model that is obtained in each training run, and the threshold is decided as follows:

$$\tau_{Q99} = Q_{0.99}(S_{tr}), \quad \hat{y}_t = I[s_t > \tau_{Q99}]. \quad (5)$$

The threshold is calculated in isolation of training scores and test labels are only used to calculate alarm metrics after the fact. The training scores are acquired by re-inferring the training sequence with the trained model, and not by training loss; their calibration samples are also not an external calibration set that is independent of the learning data of the model.

The calibration and test evaluation of the Train-Q99 do not consider the first 99 padding scores. The current Oracle Raw and PA outcomes have kept the full test sequence that was applied in the previous assessment process. Thus, the parallel results of the three settings are part of a comparison of available protocols, and no numerical difference can be purely referred to as the choice of threshold or PA regulation. The present article does not view this comparison as an ablation experiment that completely controls the evaluation sample range.

3.3 Metric Definitions and Repeated-Experiment Summarization

Point-wise alarm evaluation uses Precision, Recall, and F1:

$$P = \frac{TP}{TP+FP}, \quad R = \frac{TP}{TP+FN}, \quad F_1 = \frac{2PR}{P+R}. \quad (6)$$

When the denominator is zero, the corresponding alarm metric is treated as zero. The existing Oracle evaluation procedure adds 10^{-5} when computing $P + R$ to avoid numerical problems; Q99 uses the standard binary-classification F1 implementation. This paper retains this implementation convention and does not interpret last-digit numerical differences as meaningful performance changes.

Ranking evaluation based on AUROC and Average Precision (AP) is used. In order to be in accordance with the previously published outcome tables, this paper preserves the name of AUPR, however, it is implemented as AP and not as a trapezoidal integration of the PR curve:

$$AP = \sum_i (R_i - R_{i-1}) P_i, \quad (7)$$

in which operating points are ordered by rising recall. Raw ranking metrics are calculated based on raw scores and PA ranking metrics are calculated based on scores that are corrected using Equation (4) these two should be clearly differentiated.

Q99 alarming additionally reports false positive rate and event recall:

$$FPR = \frac{FP}{FP+TN}, \quad R_{\text{event}} = \frac{1}{K} \sum_{k=1}^K I[\exists t \in A_k: \hat{y}_t = 1]. \quad (8)$$

The false positive rate is based on the number of normal test points (denominator); event recall is based on the number of consecutive anomaly intervals (denominator), where each interval is given equal weight. Latency of detection is the number of time steps between the beginning of a hit event and the appearance of the first alarm. Latency averaging does not consider missed events; thus, latency should be considered in conjunction with event recall.

Every set of experiments has a number of model random seeds of five that are used to calculate metrics on their own and subsequently report the mean and standard deviation:

$$\bar{m} = \frac{1}{5} \sum_{i=1}^5 m_i, \quad \sigma_m = \sqrt{\frac{1}{5} \sum_{i=1}^5 (m_i - \bar{m})^2}. \quad (9)$$

The standard deviation uses the population standard deviation definition as per the available summaries, meaning that the denominator is 5. It indicates the randomness during training with the chosen seeds, is not the confidence interval, and does not explain the uncertainty due to the sampling of data or the shift in contamination locations.

3.4 Evaluation Pipeline

Algorithm 1 presents the pipeline of multi-dataset score caching and protocol assessment, which consists of organizing model inference, threshold determination, label use and repeated-experiment summarization into traceable evaluation steps.

Algorithm 1: COUTA Multi-Dataset and Evaluation Protocol Evaluation

Input: Training sequences X_{tr} , test sequences X_{te} , and test labels y for each dataset; seed set $S = \{42, 43, 44, 45, 46\}$; window length $L=100$; quantile $q=0.99$.

Output: Scores, evaluation results for each run, and the mean and standard deviation of the five runs.

- 1: for each dataset D do
- 2: Estimate normalization parameters using the training set, and process the training and test sets
- 3: for each random seed $r \in S$ do
- 4: Set the model training random seed r
- 5: Construct training windows, and train the complete COUTA according to the established configuration
- 6: Infer the training sequence and test sequence separately to obtain anomaly scores
- 7: Save scores, test labels, seed, and the valid score starting point
- 8: Remove the first $L-1$ padding positions of the training scores to obtain Str
- 9: $\tau \leftarrow \text{Quantile}(S_{tr}, q)$ ▷ This step does not use test labels
- 10: Compute $\hat{y} \leftarrow I[s > \tau]$ on valid test positions
- 11: Compute the Q99 alarm metrics using the corresponding test labels
- 12: Compute Raw ranking metrics and Oracle F1 using the complete test scores and labels
- 13: Adjust scores based on the true anomaly intervals, compute PA metrics and Oracle PA-F1
- 14: **end for**
- 15: Summarize the mean and population standard deviation of each metric separately

```

16:   end for
17:   return score records, per-run metrics, and summary results

```

The contamination experiment uses the same model training and Raw/PA evaluation stages but fits a model under different contamination levels each time. The current contamination results have no Q99 evaluation, and thus the discussion of the stability of label-free thresholds in the presence of contamination will not be made using these results.

4. Experimental Design

The current research establishes three interconnected sets of experiments. The first group considers COUTA on SMAP, MSL, and PSM to test different performances by datasets; the second group compares Oracle Raw, PA, and Train-Q99 to compare the effect of evaluation rules and label-use on results interpretation; the third group alters the synthetic training contamination ratio on PSM to see if Raw and PA exhibit similar patterns. Earlier reproduction of data from the initial paper was employed to get acquainted with the implementation and test the experimental pipeline, which are not part of the core statistical comparison of the three groups of experiments in this section.

4.1 Datasets and Evaluation Units

The SMAP and MSL data are based on the work conducted by Hundman et al., on the spacecraft telemetry anomaly detection and the open data include telemetry data and associated command data, whereas the PSM data are based on the work conducted by Abdulaal et al. on server monitoring anomaly detection [34, 35]. The original data and implementations can be found at the projects they correspond to [36, 37]. The paper is based on the organized version that was published by THUML Time-Series-Library, maintains the same training set, test set, and test labels, and does not randomly re-partition the training and test data [38]. As shown in Table 1, the statistics of each dataset are displayed.

Table 1: Experimental dataset statistics

Dataset	Training points	Test points	Feature dimension	Test anomaly points	Test anomaly ratio
SMAP	135,183	427,617	25	54,696	0.1279
MSL	58,317	73,729	55	7,766	0.1053
PSM	132,481	87,841	25	24,381	0.2776

Note: The anomaly ratio is the number of test anomaly points divided by the total number of test points; the number of training points indicates the length of the input training file and is not equal to the number of windows actually involved in optimization.

The experiment considers every organized file of every dataset as a single multivariate sequence. Specifically, SMAP and MSL are not reconstituted as initial telemetry objects with distinct modeling followed by averaging at the object level. Thus, the findings in this paper are based on the entire assessment of the organized form and can not be compared directly with the literature findings that use various splits of the entities and aggregation methods.

4.2 Preprocessing and Model Configuration

After the real experimental code comes preprocessing. The Min-Max normalization is applied to features with non-zero values of a training set using the training-set minimum and maximum:

$$\tilde{x}_{t,j} = \frac{x_{t,j} - a_j}{b_j - a_j}. \quad (10)$$

The same set of training statistics is applied to transform test data, and the respective features are clipped to $[-4.5, 4.5]$. Constant features are not eliminated: non-zero constant features are rescaled by the constant, and zero constant features keep their initial values. Interpolation and backward filling are done in the time direction after scale processing on the training sequence and test sequence individually.

The above processing has been done offline preprocessing. Despite the fact that local normalization parameters are estimated based on the training set only, interpolation and backward filling can also incorporate information about later time points within the same sequence, and this paper does not audit all processing until

the release of the organized data. Thus, it is incorrect to extrapolate the notion of the no-test-label threshold of Train-Q99 to the statement that the whole pipeline meets strong online causal requirements.

The three datasets use a consistent model configuration: window length 100, training window stride 5, inference stride 1, 20 training epochs, batch size 64, Adam learning rate 10^{-4} . Both hidden dimension and representation dimension are 16, convolution kernel size is 2, dropout is 0, NAC loss weight is 0.1, and the auxiliary sample ratio parameter is 0.2. All experiments run on CPU, and early stopping is not enabled.

The implementation preserves the default inner window split: after shuffling training windows with a constant random seed of 42, about 75% are utilized in the optimization of parameters and about 25% in computing the validation loss. Since early stopping is not implemented, validation loss does not help in choosing the end training epoch. It is not considered as an independent time-segment validation because the windows can overlap and share observations.

The seeds of model training are 42-46 sequentially influencing random processes, including parameter initialization, ordering of batches, and auxiliary sample creation. The current paper does not independently optimize hyperparameters on various test sets.

4.3 Specific Settings of the Three Experiment Groups

Experiment with multi-dataset extension. COUTA is also trained on SMAP, MSL as well as PSM each of which is repeated five times and the results are reported in terms of AuROC, AuPR and optimal F1 of Raw and PA. The experiment is aimed at measuring the performance of a fixed configuration on various datasets and does not establish an algorithm superiority order using cross-dataset metric scores.

Comparison experiment of evaluation-protocols. The train-Q99 alarm scores are calculated using the training and test score caches achieved using the same configuration and corresponding seeds, and compared against the current values of Oracle Raw and PA. At Q99 stage, there is no model-structure change made, and test labels are not used to alter the quantile. Evaluation positions of various protocols are properly held and revealed separately as per Section 3.2.

Training contamination experiment. Using PSM as an object, synthetic perturbations are added to the original training time series at rates of 0%, 1%, 3%, 5%, and 10%. Contamination ratio is a fraction of the number of polluted time periods by the total length of the training sequence, not necessarily the ratio of polluted windows or the ratio of modified cells. 0% indicates no extra synthetic anomalies have been injected, but it cannot be used to prove that the initial training data are entirely free of anomalies.

Contamination generation includes three types of operations:

1. The first point perturbation is to create values based on some set of features that are around 6-10 times the standard deviation of the overall mean.

2. The contextual perturbation is constructing a change of about 4-6 times the average scale of the neighbourhood using the neighbourhood median and standard deviations; in the case of invalid local statistics, global statistics are used.

3. Collective perturbation: use scaling and shifting on adjacent intervals, typically taking 20-100 time units; the last interval can be reduced in case of an insufficient number of remaining points.

The time-point budget of any of the three perturbations is about one-third, and in each operation, a random number between 1-6 features is selected. The contamination markers are recorded as an indicator of an injection position and not as the training supervision labels. The test anomaly sample is not used in perturbation generation.

Every contamination level has a fixed generation seed of 42, and five model seeds are executed with this constant contamination realization. The nesting of positions in different contamination levels is not guaranteed and can not be interpreted as an increase in the contamination level at the same positions.

The original test data and test labels are kept intact in all contamination conditions, but the normalization parameters are estimated again on the polluted training sets. Hence, the test inputs that are transformed and given to the model can be different depending on contamination conditions. In this experiment, the sensitivity

of the detection pipeline (including preprocessing) to the training contamination is tested, not the model mechanism effect, after completely isolating the effects of normalization.

4.4 Result Summarization and Scope of Evidence

The results of all calculations use single-run metrics and report means and population standard deviations of five runs. No processing occurs to replace single-run metric averaging with averaging five predictions. All tables are consistent in keeping four decimal places.

Everything in tables, figures and the main text has four decimal points. The PSM no-additional-contamination PA-F1 baseline of 0.8848569 is computed based on full scores and it is always reported as 0.8849; the other metrics apply the stored summary results of the respective experiments.

Descriptive comparison is used in this paper, no replacement of significance testing with the standard deviation of five seeds has been made, and trends between low numbers of contamination levels are taken as general laws.

5. Experimental Results and Analysis

5.1 Multi-Dataset Extension Results

This set of experiments addresses: the differences between ranking performance and best F1 across datasets without changing COUTA and using a fixed main configuration, and does PA adjustment alter the way performance is presented. Five runs were conducted in Table 2.

Table 2: Raw and PA results of COUTA on three datasets

Dataset	Raw AUROC	Raw AUPR	Oracle Raw F1	PA AUROC	PA AUPR	Oracle PA F1
SMAP	0.4222 ± 0.0343	0.1088 ± 0.0052	0.2319 ± 0.0098	0.9238 ± 0.0083	0.7931 ± 0.0110	0.7547 ± 0.0223
MSL	0.5320 ± 0.0141	0.1562 ± 0.0039	0.2049 ± 0.0086	0.9896 ± 0.0051	0.9084 ± 0.0176	0.8910 ± 0.0032
PSM	0.7204 ± 0.0238	0.4600 ± 0.0441	0.5476 ± 0.0202	0.9726 ± 0.0096	0.9464 ± 0.0194	0.8849 ± 0.0411

Note: AUPR is implemented as AP. Both types of F1 use test labels to select thresholds. PA metrics use scores adjusted by ground-truth anomaly intervals. All results are mean ± population standard deviation.

After Raw evaluation, PSM has a higher AUROC, AUPR, and F1 than both of the other datasets. The Raw AUROC for SMAP is reported to be 0.4222 which suggests that raw scores do not have high performance in terms of overall point-wise ranking ability based on the present data presentation and setup; MSLs has an AUROC of 0.5320 and its discriminatory power in ranking is also constrained. These findings demonstrate that the fact that the model can be applied to various datasets does not necessarily entail that the same configuration has the same effects on all data.

The F1 of SMAP, MSL, and PSM increases by 0.5228, 0.6861, and 0.3373, respectively, after the application of PA. In the case of MSL in particular, both a low Raw F1 and a high PA-F1 can be observed at the same time, which is indicative of the distinction between local high-score hits and full pointwise coverage. Since PA already broadens the range that is deemed correct in an anomaly interval, this rise should not be seen as a gain in detection ability of the model through adjustment.

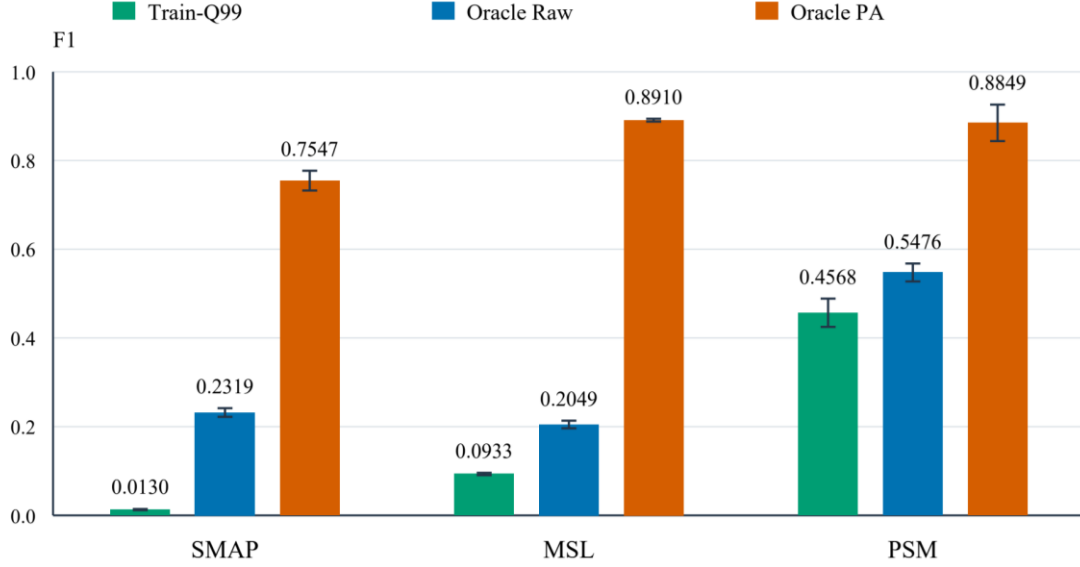
PSM and the remaining two datasets have varying anomaly rates, and so do AUPR and F1 depending on class composition. Hence, the high values of PSM do not mean that it produced superior scores in this scenario but rather cannot be employed as a tool to establish the superiority of the model based on server data. Five run results offer restricted visibility into randomness and are not synonymous with cross-scenario stability evidence.

5.2 Evaluation Protocols and Fixed-Threshold Alarm Performance

The next set of experiments also considers: do the results at offline optimal thresholds and PA have anything to do with alarm performance when thresholds are calculated without test labels. To achieve this, a training-score Q99 threshold is applied to calculate point-wise measures, event recall, and false positive rate, which are

presented together with Oracle outcomes. Figure 3 demonstrates the F1 comparison in various evaluation protocols, whereas the list of alarm measurements and evaluation outcomes of Oracle are represented in Table 3.

Figure 3: F1 comparison under different evaluation protocols. Each dataset shows Train-Q99 F1, Oracle Raw F1, and Oracle PA-F1. Error bars represent the population standard deviation of five runs. Q99 excludes the first 99 padding positions, while the original Raw and PA results retain the complete test sequence; this figure is a protocol-result comparison, not an ablation comparison with fully identical evaluation intervals



Note: Mean $\pm$ Population Standard Deviation; 5 model seeds; Q99 excludes the first 99 padding points.

Table 3: Fixed-threshold alarm and Oracle evaluation results

Dataset	Q99 Precision	Q99 Recall	Q99 F1	Oracle Raw F1	Oracle PA F1	Q99 Event Recall	Q99 False Positive Rate
SMAP	0.1156 $\pm$ 0.0113	0.0069 $\pm$ 0.0007	0.0130 $\pm$ 0.0013	0.2319 $\pm$ 0.0098	0.7547 $\pm$ 0.0223	0.5821 $\pm$ 0.0958	0.0077 $\pm$ 0.0002
MSL	0.3505 $\pm$ 0.0045	0.0538 $\pm$ 0.0019	0.0933 $\pm$ 0.0027	0.2049 $\pm$ 0.0086	0.8910 $\pm$ 0.0032	0.6889 $\pm$ 0.0208	0.0118 $\pm$ 0.0006
PSM	0.4772 $\pm$ 0.0185	0.4391 $\pm$ 0.0443	0.4568 $\pm$ 0.0319	0.5476 $\pm$ 0.0202	0.8849 $\pm$ 0.0411	0.7352 $\pm$ 0.0499	0.1844 $\pm$ 0.0098

Note: Event recall weights consecutive anomaly intervals equally; false positive rate uses normal test points as the denominator. The Q99 threshold does not use test labels for determination.

Q99 F1 is lower than Oracle Raw F1 and then lower than PA-F1 on all three datasets. SMAP and MSL have Q99 recalls of 0.0069 and 0.0538, respectively, meaning that with the present fixed threshold, the number of anomaly time points covered is very low; their event recalls are 0.5821 and 0.6889, respectively. It indicates that these abilities are not on the same level as hitting certain anomaly events and consistently covering the anomaly time intervals, and an increased event hit rate cannot be substituted by a point-wise recall.

The Q99 F1 of PSM is 0.4568, and event recall is 0.7352, but the normal-point false positive rate is 0.1844 (i.e., about 18.44 percent) of the normal test points cause an alarm to sound. It shows that greater anomaly coverage is associated with a clear false alarm load. SMAP and MSL have lower false positive rates, but the point-wise recall of these two is also smaller; a sensible threshold cannot be estimated merely by fewer false alarms.

Q99 is but one of the quantiles of the training score distribution, and it makes no assurance that the test normal-point false positive rate is at 1%1%. The correspondence can be influenced by the differences in training and test distributions, the amount of fit to training examples, and time dependency of anomaly scores; the current experiments cannot distinguish the role of each of these variables.

The mean detection latency of hit events is 77.6743+15.1371 time steps in the case of SMAP, 65.9545+14.3420 in the case of MSL and 14.5608 +2.6802 in the case of PSM. Since these findings do not

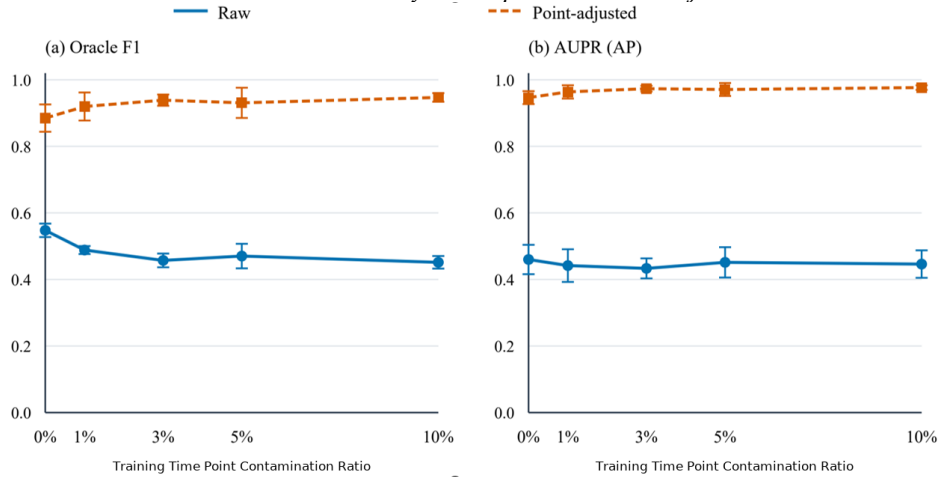
contain missed events and the sampling environments of various datasets are not the same, no inter-dataset real-time response performance comparison is conducted using them.

In general, these experiments demonstrate that the offline Oracle outcomes are not enough to directly answer the question of the effectiveness of alarms with a fixed label-free threshold. Simultaneously, poor Q99 performance means nothing more than the fact that it does not match the existing score distribution well, and it is not possible to conclude that any of the other label-free thresholds are ineffective.

5.3 Training Contamination Sensitivity

This set of experiments covers: if the initial test set stays intact, do Raw and PA assessment produce the same results regarding the performance shift after the training data has been exposed to varying levels of synthetic perturbation? The experiment only changes the synthetic contamination conditions and runs five model seeds for each of them; particular perturbations and preprocessing techniques are described in Section 4.3. The trend at each level of contamination is depicted in Figure 4 whereas the specific metric summary is presented in Table 4.

Figure 4: PSM training contamination sensitivity. (a) Oracle Raw F1 and Oracle PA-F1; (b) Raw AUPR and PA-AUPR. The horizontal axis is the ratio of synthetic perturbation time points to the training sequence. Error bars are the population standard deviation of five model seeds and do not include random variation in contamination positions. 0% indicates no additional synthetic perturbation is injected



Note: Error bars: Population standard deviation of 5 model seeds; there is only one fixed contamination realization for each contamination level.

Table 4: Detection results under different PSM training contamination levels

Contamination ratio	Raw AUPR	Oracle Raw F1	PA AUPR	Oracle PA F1
0%	0.4600 ± 0.0441	0.5476 ± 0.0202	0.9464 ± 0.0194	0.8849 ± 0.0411
1%	0.4417 ± 0.0491	0.4884 ± 0.0120	0.9636 ± 0.0200	0.9197 ± 0.0422
3%	0.4333 ± 0.0301	0.4572 ± 0.0206	0.9733 ± 0.0088	0.9388 ± 0.0170
5%	0.4514 ± 0.0456	0.4703 ± 0.0369	0.9708 ± 0.0195	0.9307 ± 0.0454
10%	0.4462 ± 0.0413	0.4515 ± 0.0189	0.9766 ± 0.0069	0.9470 ± 0.0130

Note: Each level uses the same original test data and labels, but normalization parameters are re-estimated according to its respective training set. Each contamination level has only one fixed contamination realization. 0% PA-F1 is uniformly reported as 0.8849 based on the full-precision baseline.

Raw F1 falls by an absolute amount of 0.0961 (17.6%) with a reduction in contamination of up to 10%. The modification is not necessarily monotonic: the outcome at 5% contamination is greater than that at 3% but Raw F1 with all non-zero contamination is less than the 0% baseline. Raw AUPR varies as well, and its endpoints are reduced to 0.4462, which cannot be described as steadily decreasing with increasing contamination ratio.

Conversely, PA-F1 rises by 0.8849 to 0.9470 and PA-AUPR rises by 0.9464 to 0.9766. PA has no strict monotonicity either, as it decreases at 5% when compared to 3%, however, the trend in the end points is

opposite to that of Raw F1. In case the decision is made according to PA-F1, then the decrease in raw scores at point-wise F1 can be overlooked.

These differences justify the statement that: in this PSM synthetic contamination, the evaluation protocol is able to alter the perception of the direction of performance change. This does not confirm that contamination in general will always enhance COUTA, or that it can serve as mechanistic evidence to state that UMC or NAC is ineffective. Since the normalization scale is not kept constant in this experiment under different contamination conditions, both the preprocessing and model learning processes might also be influencing the observed changes at the same time.

6. Discussion

6.1 Score Quality, Threshold Effectiveness, and Evaluation Adjustment Should be Distinguished

The results of anomaly detection include at least three stages, which are as follows: raw scores that could identify normal and anomalous, a specific threshold with the ability to create meaningful alarms, and assessment measures that determine the right detection. The Raw ranking metrics, Q99 alarm metrics and PA metrics in this paper represent various levels of the above-mentioned levels and cannot be substituted for each other.

The current metric studies indicate that various measurements have different levels of preference concerning the tolerance to time, spatial hits, and interval coverage [10]. More recent formal studies also indicate that the assessment metrics must be considered in light of explicit property constraints instead of considering a score as the only indication of success [13]. The findings presented in this paper are in line with this concept: it is possible to observe the situation where a more significant PA-F1 is accompanied by less point-wise F1 or fixed-threshold recall.

Consequently, the evaluation conclusions must show whether the ground-truth intervals were applied to adjust the scores that had been used, whether the threshold was chosen using the test labels, and how the validity of evaluation positions is established. These conditions do not exist, so it is hard to interpret the concept of “good detection performance” correctly.

In this layered view, the findings of this paper can be taken as having two particular implications. On the one hand, the high scores of SMAP and MSL in PA do not imply high point-wise recall in Train-Q99, which shows that the presence of high scores in events does not mean the same thing as the consistent coverage of anomaly duration. On the other hand, PSM has a relatively high Q99 recall, but at the same time an 18.44% false positive rate, which means that alarm capability cannot be defined solely by recall, but by recall and false alarm alike. As per the unified assessment principles of TAB and mTSBench, individual-model reports based on experience must also reveal separate elements.

6.2 Contamination Robustness Depends on Experimental Conditions

Robustness against contamination must be stated in reference to an explicit contamination distribution, a preprocessing step, and an evaluation goal. The paper notes that Raw F1 and PA-F1 have opposing endpoint change directions, but it does not directly measure the score shape of each anomaly interval before and after contamination.

This may be explained by the fact that the model can continue to generate local maximums in the anomaly range, but it can no longer determine the other positions in the range to be normal points; the interval-maximum operation of the PA enhances the importance of local peaks. Nevertheless, particular associations between within-interval peaks, coverage, normal-point scores, and normalization changes have not been broken down and checked, so this explanation is merely a hypothesis that fits with the results.

The COUTA and TSAD-C methods of training calibration or data purification are applied to contaminated data [7, 32]. This paper extends this research strand in the evaluation direction: it should be determined by results with no ground-truth interval adjustment that the designs of anti-contamination can preserve point-wise coverage. In the PSM experiment, Raw F1 at all non-zero contamination states is below the 0 percent baseline

and PA-F1 is above the baseline, which demonstrates that the choice of evaluation can directly influence the story of robustness. Mechanistic attribution has not been established yet by independent module and preprocessing control investigations.

The application of synthetic anomalies should also be differentiated. NCAD and AnomalyLLM apply auxiliary anomalies to mold learning signals, but the contamination considered in this paper is introduced into the training data as unlabeled perturbation [26, 31]. Contamination does not necessarily play a role in auxiliary supervision even if the two use the same form of perturbation. The PA increase in this paper cannot be seen as the idea of more contamination leading to higher training and must be evaluated along with Raw values.

6.3 Implications for Alarm Applications

The fixed-threshold outcomes imply that the offline best F1 cannot be used to characterize the real-life alarm burden. In the case of SMAP and MSL, low point-wise recall and low event hits are also problematic; in the case of PSM, the relatively large normal-point false positive rate is problematic. The metric trade-offs ought to be made at the same time based on the cost of missing detection, the cost of handling false alarms and the requirement of response-time in the application.

Online adaptive threshold studies consider threshold estimation to be a problem of separate design [14]. This paper Q99 is just a rough reference with no adaptive update mechanism or assurance against false alarms. Hence, a rational conclusion would be to test if the threshold rule does not fit application conditions and thus cannot be recommended or rejected by the current findings.

In terms of application, industrial research from Mueller's perspective focuses on the gap between factors like streaming processing, human-in-the-loop, and conditional anomalies and ordinary offline experiments [39]. The combination of reporting on the recall of events, the latency, and false positive rate in this paper addresses the issue of alarm interpretation in it, but these measures remain offline measures. The findings justify treating the threshold calibration as a distinct validation step: specify the data accessible to calibrate, report the recall and false alarms of the chosen operating point, and consider the control of false alarms on upcoming normal data as a feature to be tested instead of being an automatic output of training Q99. Multi-system calibration studies of M²AD's offer appropriate methodological background related to this direction [15].

7. Research Limitations and Future Research Directions

7.1 Research Limitations

To begin with, this paper considers only COUTA, and the comparison of the algorithm with other anomaly detectors is not unified. These conclusions should not be used to declare that COUTA is better or worse than the other techniques, nor should they be considered as universal characteristics inherent to all the deep anomaly detectors.

The second reason is that the multi-dataset experiments take different training and testing models, and no cross-domain transfer is used. SMAP and MSL have organized whole sequences which are not reconstructed to initial entity limits; cross-boundary windows and aggregation techniques can influence findings. The experimental conclusions are thus restricted to the existing data representation and processing techniques.

The third is that the contamination experiment only applies to PSM and pre-programmed synthetic perturbation. Every ratio has only a single fixed contamination realization; the five seeds represent model training randomness, and they do not consider the contamination location, composition of types, or real sample data. Also, there are changing parameters of normalizations depending on the contaminated training set, which restricts the possibility of attributing it to the model-based mechanisms of anti-contamination within the model.

To sum up, the boundaries between different protocols are clearly defined: Q99 and the initial Raw/PA outputs treat the first 99 padding scores in a different way; Q99 employs training-sequence scores, but not a separate calibration sample; offline missing-value processing cannot meet the strict causality requirements of online processing. This paper thus views these findings as an empirical comparison of current pipelines, but not as a full validation of online systems or a total ablation of an established protocol.

7.2 Future Research Directions

Future research may develop the available evidence in three ways. To begin with, it may be possible to further distinguish between the implications of organizing data and evaluating protocols in conditions of restoring the entity boundaries and unification of valid evaluation positions. Secondly, provide independent historical calibration data, examine the threshold modification when there is a movement of distributions and alarm costs, and confirm online applicability through strictly causative preprocessing. Third, modify the generation of contamination, perturbation types, and intensities across various datasets and isolate the impact of changes in scale of preprocessing and learning of model parameters.

The longer-term research may include both detecting occurred anomalies and predicting future anomalies. The Anomaly-to-Prompt approach by Park et al. particularly examines the prediction of future anomalies, which implies that the goals of early warnings should be based on the redefinition of training data, prediction range, and the limits of evaluation time [40]. Extensions to COUTA that are related need to first explain what has been modified in the task and then examine whether detection representations may facilitate early warning and not directly compare the current detection latency with forecasting ability.

8. Conclusion

This paper carries out multi-dataset assessment on SMAP, MSL, and PSM, evaluation-protocol comparison, and PSM training contamination sensitivity analysis without altering the COUTA model, and it provides the average and standard deviation of results using five model random seeds.

According to experiments, COUTA performs differently with regard to dataset and evaluation setup under the existing configuration. SMAP and MSL have a large PA-F1, but they have low raw point-wise performance and Train-Q99 recall; PSM has a relatively high anomaly coverage in Q99 and generates perceptible normal-point false alarms. In the PSM contamination experiment, Raw F1 reduces by 0.5476 to 0.4515, but PA-F1 rises by 0.8849 to 0.9470, which means that various evaluation protocols could provide varying assessments of the contamination sensitivity of the same detection pipeline.

The research identifies a traceable multi-seed evaluation outcome set, measures the difference between offline assessment and fixed-threshold warning, and the directional divergence between point-wise testing and PA testing in training contamination. The results offer a clean experimental reference in explaining how COUTA performed and improved further and encourage a more comprehensive reporting strategy: explain at the same time the quality of raw scores, the source of threshold information, the extent of events covered, and the load of false alarms, as well as relate the findings on contamination resistance to particular preprocessing and evaluation settings.

References

- [1] Darban, Z. Z., Webb, G. I., Pan, S., et al. (2024). Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*, 57 (1), Article 15. <https://doi.org/10.1145/3691338>
- [2] Sánchez-Ferrera, A., Calvo, B., & Lozano, J. A. (2026). A review on self-supervised learning in time series anomaly detection: Recent advances and open challenges. *ACM Computing Surveys*, 58 (5), Article 119. <https://doi.org/10.1145/3770575>
- [3] Paparrizos, J., Boniol, P., Liu, Q., et al. (2025). Advances in time-series anomaly detection: Algorithms, benchmarks, and evaluation measures. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2* (pp. 6151–6161). ACM. <https://doi.org/10.1145/3711896.3736565>
- [4] Liu, Q., & Paparrizos, J. (2024). The elephant in the room: Towards a reliable time-series anomaly detection benchmark. In *Advances in Neural Information Processing Systems* (Vol. 37, pp. 108231–108261). <https://doi.org/10.52202/079017-3437>
- [5] Qiu, X., Li, Z., Qiu, W., et al. (2025). TAB: Unified benchmarking of time series anomaly detection methods. *Proceedings of the VLDB Endowment*, 18 (9), 2775–2789. <https://doi.org/10.14778/3746405.3746407>

- [6] Zhou, X., Brif, C., & Lourentzou, I. (2026). mTSBench: Benchmarking multivariate time series anomaly detection and model selection at scale. *Transactions on Machine Learning Research*.
- [7] Xu, H., Wang, Y., Jian, S., et al. (2024). Calibrated one-class classification for unsupervised time series anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 36 (11), 5723–5736. <https://doi.org/10.1109/TKDE.2024.3393996>
- [8] Kim, S., Choi, K., Choi, H. S., et al. (2022). Towards a rigorous evaluation of time-series anomaly detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36 (7), 7194–7201. <https://doi.org/10.1609/aaai.v36i7.20680>
- [9] Bhattacharya, D., Mukherjee, S., Kamanchi, C., et al. (2025). Towards unbiased evaluation of time-series anomaly detector. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 1–5). IEEE. <https://doi.org/10.1109/ICASSP49660.2025.10890568>
- [10] Sørbo, S., & Ruocco, M. (2024). Navigating the metric maze: A taxonomy of evaluation metrics for anomaly detection in time series. *Data Mining and Knowledge Discovery*, 38, 1027–1068. <https://doi.org/10.1007/s10618-023-00988-8>
- [11] Ghorbani, R., Reinders, M. J. T., & Tax, D. M. J. (2024). PATE: Proximity-aware time series anomaly evaluation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 872–883). ACM. <https://doi.org/10.1145/3637528.3671971>
- [12] Boniol, P., Krishna, A. K., Bruel, M., et al. (2025). VUS: Effective and efficient accuracy measures for time-series anomaly detection. *The VLDB Journal*, 34, Article 32. <https://doi.org/10.1007/s00778-025-00907-x>
- [13] Wagner, D., Nair, A., Franks, B. J., et al. (2026). Formally exploring time-series anomaly detection evaluation metrics. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics* (Vol. 300, pp. 3448–3456). PMLR.
- [14] Sun, S. H., Sankararaman, A., & Narayanaswamy, B. M. (2024). Online adaptive anomaly thresholding with confidence sequences. In *Proceedings of the 41st International Conference on Machine Learning* (Vol. 235, pp. 47105–47132). PMLR.
- [15] Alnegheimish, S., He, Z., Reimherr, M., et al. (2025). M²AD: Multi-sensor multi-system anomaly detection through global scoring and calibrated thresholding. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics* (Vol. 258, pp. 4384–4392). PMLR.
- [16] Yang, Y., Zhang, C., Zhou, T., et al. (2023). DCdetector: Dual attention contrastive representation learning for time series anomaly detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 3033–3045). ACM. <https://doi.org/10.1145/3580305.3599295>
- [17] Sun, Y., Pang, G., Ye, G., et al. (2024). Unraveling the “anomaly” in time series anomaly detection: A self-supervised tri-domain solution. In *2024 IEEE 40th International Conference on Data Engineering* (pp. 981–994). IEEE. <https://doi.org/10.1109/ICDE60146.2024.00080>
- [18] Fang, Y., Xie, J., Zhao, Y., et al. (2024). Temporal-frequency masked autoencoders for time series anomaly detection. In *2024 IEEE 40th International Conference on Data Engineering* (pp. 1228–1241). IEEE. <https://doi.org/10.1109/ICDE60146.2024.00099>
- [19] Nam, Y., Yoon, S., Shin, Y., et al. (2024). Breaking the time-frequency granularity discrepancy in time-series anomaly detection. In *Proceedings of the ACM Web Conference 2024* (pp. 4204–4215). ACM. <https://doi.org/10.1145/3589334.3645556>
- [20] Dai, Z., He, L., Yang, S. H., et al. (2024). SARAD: Spatial association-aware anomaly detection and diagnosis for multivariate time series. In *Advances in Neural Information Processing Systems* (Vol. 37, pp. 48371–48410). <https://doi.org/10.52202/079017-1533>
- [21] Wu, X., Qiu, X., Li, Z., et al. (2025). CATCH: Channel-aware multivariate time series anomaly detection via frequency patching. In *Proceedings of the 13th International Conference on Learning Representations*.

- [22] Yin, T., Fu, S., Zhang, Z., et al. (2025). ScatterAD: Temporal-topological scattering mechanism for time series anomaly detection. In *Advances in Neural Information Processing Systems* (Vol. 38, pp. 52266–52297).
- [23] Li, M., Liu, K., Chen, H., et al. (2025). TSINR: Capturing temporal continuity via implicit neural representations for time series anomaly detection. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1* (pp. 671–682). ACM. <https://doi.org/10.1145/3690624.3709266>
- [24] Zhou, Q., Pei, C., Sun, F., et al. (2025). KAN-AD: Time series anomaly detection with Kolmogorov–Arnold networks. In *Proceedings of the 42nd International Conference on Machine Learning* (Vol. 267, pp. 79136–79149). PMLR.
- [25] Chen, Y., Zhang, C., Ma, M., et al. (2023). ImDiffusion: Imputed diffusion models for multivariate time series anomaly detection. *Proceedings of the VLDB Endowment*, 17 (3), 359–372. <https://doi.org/10.14778/3632093.3632101>
- [26] Liu, C., He, S., Zhou, Q., et al. (2024). Large language model guided knowledge distillation for time series anomaly detection. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence* (pp. 2162–2170). IJCAI. <https://doi.org/10.24963/ijcai.2024/239>
- [27] Goswami, M., Szafer, K., Choudhry, A., et al. (2024). MOMENT: A family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning* (Vol. 235, pp. 16115–16152). PMLR.
- [28] Mejri, N., Lopez-Fuentes, L., Roy, K., et al. (2024). Unsupervised anomaly detection in time-series: An extensive evaluation and analysis of state-of-the-art methods. *Expert Systems with Applications*, 256, 124922. <https://doi.org/10.1016/j.eswa.2024.124922>
- [29] Sarfraz, M. S., Chen, M. Y., Layer, L., et al. (2024). Position: Quo vadis, unsupervised time series anomaly detection? In *Proceedings of the 41st International Conference on Machine Learning* (Vol. 235, pp. 43461–43476). PMLR.
- [30] Huet, A., Navarro, J. M., & Rossi, D. (2022). Local evaluation of time series anomaly detection algorithms. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 635–645). ACM. <https://doi.org/10.1145/3534678.3539339>
- [31] Carmona, C. U., Aubet, F. X., Flunkert, V., et al. (2022). Neural contextual anomaly detection for time series. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence* (pp. 2843–2851). IJCAI. <https://doi.org/10.24963/ijcai.2022/394>
- [32] Ho, T. K. K., & Armanfard, N. (2025). Contaminated multivariate time-series anomaly detection with spatio-temporal graph conditional diffusion models. In *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence* (Vol. 286, pp. 1710–1729). PMLR.
- [33] Darban, Z. Z., Yang, Y., Webb, G. I., et al. (2025). DACAD: Domain adaptation contrastive learning for anomaly detection in multivariate time series. *IEEE Transactions on Knowledge and Data Engineering*, 37 (8), 4485–4496. <https://doi.org/10.1109/TKDE.2025.3569909>
- [34] Hundman, K., Constantinou, V., Laporte, C., et al. (2018). Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 387–395). ACM. <https://doi.org/10.1145/3219819.3219845>
- [35] Abdulaal, A., Liu, Z., & Lancewicki, T. (2021). Practical approach to asynchronous multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (pp. 2485–2494). ACM. <https://doi.org/10.1145/3447548.3467174>
- [36] Hundman, K., et al. (n.d.). Telemanom: A framework for using LSTMs to detect anomalies in multivariate sensor data. GitHub. Retrieved September 13, 2026, from <https://github.com/khundman/telemanom>
- [37] eBay. (n.d.). RANSynCoders: Pooled Server Metrics dataset and implementation. GitHub. Retrieved September 13, 2026, from <https://github.com/eBay/RANSynCoders>

- [38] THUML. (n.d.). Time-Series-Library: Dataset collection [Data set]. Hugging Face. Retrieved September 13, 2026, from <https://huggingface.co/datasets/thuml/Time-Series-Library>
- [39] Mueller, A. (2025). Open challenges in time series anomaly detection: An industry perspective [Preprint]. arXiv. <https://arxiv.org/abs/2502.05392>
- [40] Park, M. Y., Lee, W. J., Kim, S. T., et al. (2025). When will it fail?: Anomaly to prompt for forecasting future anomalies in time series. In Proceedings of the 42nd International Conference on Machine Learning (Vol. 267, pp. 48086–48103). PMLR.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Copyrights

Copyright for this article is retained by the author (s), with first publication rights granted to the journal. This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).