Using Supervised Anomaly Detection Algorithms to Localize Anomalies in Unlabeled Time Series Training Data

Matthew Gerdes
Oracle
Austin, Texas, USA
matthew.gerdes@oracle.com

Guang Wang*
Oracle
Austin, Texas, USA
guang.wang@oracle.com

Abstract

Machine learning techniques are increasingly being used in the area of multivariate time series anomaly detection. However, their effectiveness—particularly for supervised approaches—is often limited by the scarcity of labeled training data. Identifying anomalies in the unlabeled training data is very challenging without the knowledge of subject matter experts. Therefore, researchers usually assume that anomalies in training data are sparse enough to be negligible—an assumption often violated in real-world scenarios. Conventional attempts to remove anomalies in the training data are based on simplistic outlier detection methods, such as three standard deviation thresholds, or methods intended for univariate analysis, which inadequately handles complex multivariate data. This paper introduces a preprocessing method designed to enhance existing supervised anomaly detection models. Our method employs an existing supervised algorithm to localize faults in unlabeled multivariate training data through a recursive process of partitioning and fault inferencing, progressively narrowing down faults to smaller regions and thereby benefiting supervised detection tasks in multivariate time series. The method is positioned as an ancillary technique intended to benefit a broad set of existing supervised anomaly detection algorithms, as opposed to a standalone anomaly detection technique. The capability of our method is demonstrated on both synthetic and published datasets where the labeled ground-truth defects are available. We show how our method improves the supervised model with the unlabeled training data, resulting in greater anomaly detection accuracy.

Keywords

Anomaly Detection, Unlabeled Training Data, Supervised Learning, Time Series

ACM Reference Format:
Matthew Gerdes and Guang Wang. 2025. Using Supervised Anomaly Detection Algorithms to Localize Anomalies in Unlabeled Time Series Training Data. In *Proceedings of The 11th Mining and Learning from Time Series Workshop: From Classical Methods to LLMs (MILETS '25).* ACM, New York, NY, USA, 9 pages.

*Corresponding author

MILETS '25, Toronto, Canada

1 Introduction

Data-driven machine learning models have been commonly leveraged in the area of multivariate time series anomaly detection. One significant challenge is the unavailability of labeled training data for all measurements under normal operating conditions of the system of interest, which degrades anomaly detection tasks downstream of the training. Labeling anomalies is very difficult and sometimes even infeasible, mainly because manually labeling the data by subject matter experts is impractical especially when large scale sensor network is deployed [2].

Conventional attempts to remove anomalies in the training data are based on simplistic outlier detection methods, such as three standard deviation thresholds [3]. Unsupervised techniques are often utilized when no prior knowledge of the training dataset is available. However, many unsupervised techniques employ clustering type algorithms that arguably assume the anomalous data reside inside small clusters [20], or are built through linear projection and transformation, which is unable to handle non-linearity and exploit the hidden inter-correlations of the multivariate time series [15]. More sophisticated unsupervised methods such as Generative Adversarial Networks (GAN, [6]) have proven helpful, but the fact that its training objective often results in saddle point convergence makes the GAN models hard to train.

On the other hand, supervised techniques suffer greater challenges from analyzing unlabeled training data than the unsupervised techniques. The key prerequisite for creating a proper supervised anomaly detection model has always been that the training dataset has labeled instances for normal as well as anomaly classes [4]. Otherwise the model will ultimately learn the wrong data distribution, leading to biased training models and causing false alarms and missed alarms. Supervised techniques for time series anomaly detection have been intensively studied and widely adopted for decades, from the conventional approaches such as Support Vector Machines [16] and Artificial Neural Networks [12], to more recent Long Short Term Memory (LSTM) models that have been found to perform well with long term temporal dependencies [11, 14, 22]. However, the availability of labeled training data and the impact of the unlabeled training data have received little attention. The state-of-art supervised approaches to anomaly detection commonly assume that the number of anomalies in the training data is small enough that its negative impact on the model can be ignored, which is often not the case. In fact, we found the presence of unlabeled anomalies in the training data will result in those same anomalies not being flagged later during the monitoring of the same assets.

Motivated by this issue, we propose a novel preprocessing method aimed specifically at improving existing supervised anomaly detection models. Our method recursively applies a chosen supervised

anomaly detection technique to localize anomalous regions within unlabeled multivariate datasets. Specifically, we subdivide unlabeled training data into partitions, build supervised models on combinations of the partitions, and recursively infer the partitions that are most likely to contain anomalies from the output of these models. The partitions flagged as anomalous are further subdivided into smaller partitions in the subsequent iterations, to localize the smallest segments containing anomalies (thereby maximizing the non-anomalous data that will be used for model training), while the partitions flagged as normal are retained. The recursive process is terminated when certain stopping criteria are met. The combination of all retained partitions after the recursive process is expected to constitute a refined training dataset covering the normal operating status of the system. The proposed method is positioned as an ancillary technique that benefits supervised anomaly detection algorithms, as opposed to a standalone anomaly detection algorithm.

The case studies presented to demonstrate and validate our proposed technique utilize a nonlinear multivariate pattern recognition approach known as the Multivariate State Estimation Technique (MSET, [21]), but the methodology is designed to be easily adaptable to any existing multivariate techniques. We illustrate the risk of using unlabeled training data for supervised detection, and validate the performance of our technique with both synthetic data and published data with known ground-truth defects available. We demonstrate that, after removing the suspect regions flagged by our technique from the unlabeled training data, the same supervised anomaly detector used in our method achieves improved detection performance—specifically, a significant reduction in missed alarms.

The paper is organized as follows. In Section 2, the multivariate time series anomaly detection model and the datasets used in this work are introduced. Section 3 introduces the methodology and mathematical derivation of our proposed method. In Section 4, we present two case studies using both synthetic and real world datasets, examine the performance of our technique through detailed analysis, and discuss the results, applications, and benefits of our technique in detail. Finally, Section 5 concludes the paper.

2 Background and Data

2.1 Multivariate Time Series Anomaly Detection Model

The supervised model used in this work employs a nonlinear, nonparametric, multivariate pattern recognition technique, called the Multivariate State Estimation Technique (MSET). It was originally developed by Argonne National Laboratory (ANL) to discover anomalies in time series sensor data in nuclear power applications [7, 10]. Over the years, MSET has been evolved and scaled to the big data prognostic applications commonly seen in safety-critical industries including aerospace, utilities, and computer systems [8, 9]. The mathematical derivation of the latest MSET algorithm is outlined in this section.

The main objective of MSET is to make a quantitative assessment of the current operation status by using the degree of similarity between historical normal operating data and the current surveillance observations. First, the degree of similarity between two matrices A and B of the same column size is defined by $A \otimes B$, where $\otimes$ represents a proprietary non-linear matrix operator.

Assume the historical data $\mathbf{D}$ from the monitored system under normal operation consisting of m measurements and n sensors is available. A data subset D, consisting of m' measurements and n sensors that preserves prominent non-linear dynamic and inter-correlations between the sensors, is selected:

$$D = \begin{bmatrix} X_{1,1} & \dots & X_{1,n} \\ \vdots & \ddots & \vdots \\ X_{m',1} & \dots & X_{m',n} \end{bmatrix} \in \mathbb{R}^{[m' \times n]} \tag{1}$$

To proceed, the pairwise correlation between the measurements in D can be quantified by:

$$D^\top \otimes D, \tag{2}$$

resulting in a symmetric and positive definite matrix.

To minimize the Euclidean norm between the estimated and measured data vectors X^{obs}, a weight w is defined by:

$$w = (D^\top \otimes D)^* (D^\top \otimes X^{\text{obs}}), \tag{3}$$

where sign $*$ indicates pseudoinverse calculation, which can accommodate a singular matrix caused by two or more repeated or highly correlated sensor signals in the dataset (i.e., high collinearity).

MSET estimates X^{est} are produced for new observations X^{obs} by:

$$X^{\text{est}} = Dw = D(D^\top \otimes D)^* (D^\top \otimes X^{\text{obs}}). \tag{4}$$

The residual errors between the MSET estimates and the actual observations are:

$$e = X^{\text{est}} - X^{\text{obs}}. \tag{5}$$

Finally, the residuals e go through a statistical binary hypothesis test called Sequential Probability Ratio Test (SPRT, [23]), which makes the anomaly detection decisions considering the log likelihood ratio (LLR) as a function of the number of observations:

$$\text{LLR} = \log \left[\frac{\prod_{i=1}^{n} P(e_i|\text{normal})}{\prod_{i=1}^{n} P(e_i|\text{abnormal})} \right]. \tag{6}$$

The SPRT algorithm quantifies both mean and variance shifts between the normal distribution and any degraded distribution to flag anomalies.

2.2 Benchmark Dataset

A number of public benchmark datasets have been commonly used to evaluate various anomaly detection techniques. However, Wu et al. [25] recently did a careful evaluation of these datasets and concluded that the majority of the faults in the datasets suffer from one or more of four flaws. To address the reasonable concerns outlined in [25], we have developed a compendium of realistic types of time series sensor fault signatures that we have observed over the years in real anomaly detection use cases across a variety of industries, which provides a reliable test bed with "known ground truth" injected faults and "known ground truth" absence of faults, for evaluating our proposed technique.

The compendium of fault types is summarized as follows:

- Ramp: A basic type of anomaly, emulating signal drift, where degradation initiates "inside" the noise level of the signals, develops over time, and eventually goes outside of the (undegraded) signal's range.
- Mean Shift: A graduate or abrupt change point in the mean level of a time series (e.g., caused by sensor decalibration bias, or sudden changes in the ambient condition, or actual onset of asset degradation).
- Gain Change: A class of "inlier failures" in IoT industries, where the physical transducer gradually loses its response to the physical parameter it is sensing, which the instrumentation specialists call "Loss-of-Gain" failures and cannot be detected by common outlier detection methods.
- Time Lag: A type of clock-skew error in one or more distributed Data Acquisition modules.
- Phase Shift: A frequency modulation error in one or more distributed Data Acquisition modules.
- Signal Dropout: One sensor out of an array of sensors suddenly disappears (data acquisition fault) or goes to a perfectly flat line (transducer "stuck-at" fault), resulting in the loss of its share in subsequent accumulated measurements.

To create datasets to analyze, a synthetic signal generator is often used for machine learning tuning and validation[13]. We leverage the Telemetry Parameter Synthesis System (TPSS) for machine learning tuning and validation[24], which generates a set of synthetic time series telemetry signals (based on real measured signals) using Fourier decomposition and reconstruction, with customization of sampling rates, signal-to-noise ratio, degree of serial correlation, amplitudes, mean, and a "dispersion factor" for distributing the signals more widely and arbitrarily across a user-defined range. The synthetic signals exhibit statistically indistinguishable characteristics from the real-world signals from which they were derived.

The aforementioned library of fault types are synthesized and injected individually or collectively into one or more synthetic signals at the same and/or different times under a wide variety of single-fault, multiple-sequential fault, and multiple concurrent fault scenarios. Because the ground truth is known in the synthetic datasets, they are very useful for tuning, validation, and assessment of our algorithm.

Additionally, we also use a real world dataset generated and published by iTrust [5], which contains multivariate time series measurements from a down-scaled model of Singapore's water treatment system (SWaT) built with the aim to publish time series data dedicated to the anomaly detection research. It has labeled anomalies in the testing dataset for validation[1].

3 Methodology

The proposed technique partitions the data and performs multiple rounds of analysis to localize the regions that most likely contain faults. The partitioning strategy is designated by 2^R, where R is the number of rounds. In the first round, it divides the data to two halves, and then uses a supervised anomaly detection model (e.g., the MSET model introduced in Section 2) to conduct training and testing using the two partitions following a scheme of cross-validation approach. Anomaly decisions on the measurements are made by the model during the testing process for each partition, and then further evaluated through a quantitative "fault inferencing" method, where the severity of the anomalous measurements are characterized at the partition level. Afterwards, the partitions flagged as normal will be included in the model training in the subsequent rounds of analysis. The partitions flagged as suspect are concatenated for the next round of partitioning, where either a cross-validation analysis or sequential testing (more details later) will be conducted to further narrow down the regions containing faults. The program stops progressing to the next round when one of the two terminating conditions is reached. Since our technique localizes the faults through a recursive procedure of partitioning and fault inferencing, we call it Recursive Fault Localization Process (RFLP).

A detailed example is provided for more clarity. Given a training dataset $\mathbf{D}$, the algorithm starts with splitting the data into two partitions of equal size. An MSET model is trained using the first partition, and tested on the second partition. Then the two partitions are swapped and the training-testing procedure is repeated. From here, two possible scenarios and terminating conditions are considered:

1) No anomaly is localized in either partition. Both partitions are assumed to be anomaly-free, but we still perform an additional round of partitioning and cross-validation process to ensure that no anomaly is missed due to a possible but unlikely scenario where an identical unlabeled fault signature is in both partitions. If no anomaly is found in any of the partitions again, the algorithm terminates with the entire dataset being considered anomaly-free.

2) k and $2^R - k$ partitions are flagged as anomalous and anomaly-free respectively. The aggregated anomalous partition $P^{(R)}_{\text{anom}}$ and aggregated clean partition $P^{(R)}_{\text{cln}}$ in Round R are expressed in Eqn. (7) and Eqn. (8) respectively.

$$P^{(R)}_{\text{anom}} = \mathcal{P}^{(R)}_1 + \cdots + \mathcal{P}^{(R)}_k, k \in [1,\ 2^R] \tag{7}$$

$$P^{(R)}_{\text{cln}} = \mathcal{P}'^{(R)}_1 + \cdots + \mathcal{P}'^{(R)}_{2^R-k}, k \in [1,\ 2^R] \tag{8}$$

where $\mathcal{P}$ and $\mathcal{P}'$ represent the suspicious partitions that are deemed as anomalous and anomaly-free respectively, and $+$ is the symbol for concatenation operation. To proceed, $P^{(R)}_{\text{anom}}$ is further subdivided into smaller partitions, while $P^{(R)}_{\text{cln}}$ is carried over and included as part of the training data to build a new model for the next round of analysis, in which each of the smaller partitions is tested with the new model, followed by the fault inferencing process. The recursive process propagates up to $R_{\max}$ rounds of analysis, and a detailed discussion on determining the maximum number of rounds $R_{\max}$ is provided in Section 4.3.3. At the end of the final round, the partitions deemed as anomalous across all the signals are concatenated at the time level before being removed from the dataset, resulting in a refined dataset for training.

The criteria for a partition to be flagged as anomalous during the fault inferencing process is introduced. As explained in Section 2.1, the SPRT algorithm makes a binary decision on whether a given observation is anomalous. It assigns a decision value of 0 to an

[1]https://itrust.sutd.edu.sg/itrust-labs_datasets/, accessible as of May 2025.

observation deemed as clean, and 1 to an observation deemed as anomalous. We characterize the severity of the anomalous observations at the partition level with an "anomaly severity" metric that is called "Tripping Frequency" (TF, [17]). It examines the density and growth rate of the time series anomalous points using a unitless measure, which is amenable to signals with different units. Specifically, after the SPRT algorithm has made a time series anomaly decision for the measurements inside a partition, we compute the cumulative sum of the decision values from the beginning of the partition up to a time step of every time step of the partition, resulting in a continuous function. We then fit a linear regression model to the cumulative function, and compute the slope of the model, which is mathematically summarized by Eqns. (9)-(11):

$$\begin{aligned} Y_i &= \sum_{j=1}^{i} x_j, \\ \hat{\gamma} &= \arg\min_{\gamma} \Big(\sum_{i}^{L_R} (Y_i - f(i,\gamma))^2 \Big), \\ f(i,\gamma) &= Y_i - i\gamma, \end{aligned} \tag{9}$$

where $x_j = 0$ or 1 is the SPRT decision value for the jth element of the time series and Y_i is the cumulative sum of the decision values determined at the ith time step of the time series, $\hat{\gamma} \in [0,\ 1]$ is the derived slope from the regression model. L_R indicates the size of each suspicious partition (i.e., $\mathcal{P}^{(R)}$ or $\mathcal{P}'^{(R)}$) in round R:

$$L_R = \frac{\mathbf{card}(P_{\text{anom}}^{(R-1)})}{2^R}, \tag{10}$$

where "**card**" indicates cardinality, and

$$P_{\text{anom}}^{(R-1)} = \begin{cases} \mathcal{P}_1^{(R-1)} + \cdots + \mathcal{P}_k^{(R-1)}. & R > 1 \\ \text{Full Dataset } \mathbf{D}. & R = 1 \end{cases} \tag{11}$$

TF characterizes the severity of the anomalous points labeled by SPRT by discriminating the adjacent or consecutive anomalous points that are indicative of a developing anomaly, from the sparsely spaced anomalous points that are often caused by the statistical variations in the data. A TF threshold is needed to define the severity. A larger TF threshold makes the RFLP algorithm prioritize removing the obvious anomalies and preserving as much training data as possible. A smaller TF threshold results in a conservative decision-making process, which can remove subtle anomalies in the training data at the cost of some anomaly-free data being unnecessarily removed. Through extensive empirical testing on synthetic data, the TF threshold is determined to be 0.1. Hence, if a partition containing some anomalous measurements labeled by SPRT is shown to have $\hat{\gamma} > 0.1$, it is flagged as anomalous (per $\mathcal{P}^{(R)}$ in Eqn. 7). This empirical TF threshold is found to be sensitive enough to detect large instantaneous faults, such as spike changes, and subtle developing faults that span a longer time period, without being too rigorous by filtering out the partitions with the abnormal points contained therein appearing to be random noise. Ultimately, prognostic performance requirements specific to the use case are needed to determine a more optimal TF threshold. For such cases, a TF threshold could be developed as a function of the sample size and the economic-safety profiles of the machine learning users.

Last, for added clarity, the pseudocode implementation of the RFLP technique using MSET as the chosen supervised anomaly detector is presented in Algorithm 1.

Algorithm 1 Pseudocode implementation of the RFLP technique using MSET

Input: unlabeled training data $\mathbf{D} \in \mathbb{R}^{[m \times n]}$
$P_{\text{anom}}^{(0)} \leftarrow \mathbf{D}$
for $R \in \{1, \ldots, R_{\max}\}$ **do**
  $L_R \leftarrow \frac{\mathbf{card}(P_{\text{anom}}^{(R-1)})}{2^R}$
  for $k \in \{1, \ldots, 2^R\}$ **do**
    // parse partitions for testing
    $\mathcal{P}_k^{(R)} \leftarrow [(k-1)L_R + 1 \ldots kL_R, j]\,, j \in [1, n]$
    // parse partitions for training
    $P_{\text{cln}}^{(R)} \leftarrow \left[\mathbf{D} \backslash \mathcal{P}'^{(R)}_k, j\right], j \in [1, n]$
    $\rightarrow$ train model on $P_{\text{cln}}^{(R)}$ and test $\mathcal{P}'^{(R)}_k$
    $\rightarrow$ calculate TF, solve $\hat{\gamma}$
    if $\hat{\gamma} > 0.1$ **then**
      $P_{\text{anom}}^{(R)} \leftarrow P_{\text{anom}}^{(R)} + \mathcal{P}_k^{(R)}, k \in [1, 2^R]$
    end if
  end for
  if $R = 1$ & $P_{\text{anom}}^{(R)}$ is empty **then**
    $R = R + 1$
  else if $P_{\text{anom}}^{(R)}$ is empty **then**
    // No partitions flagged as anomalous
    $\rightarrow$ Terminate
  end if
end for
Output: refined training data $\mathbf{D}$

4 Results and Discussion

4.1 Case Study with Synthetic Data

A detailed example using synthetic data is presented to demonstrate the performance of the RFLP technique on discerning separate faults in same signal. Per Section 2.2, we generate 100 synthetic signals, each of which is composed of three sinusoidal time series of 5k samples each with different mean, variance and noise ratio. Two faults of different type, length, and magnitude are synthesized and deliberately superimposed onto the first half and the second half of one of the synthetic signals.

Figure 1 illustrates the synthetic signals before (blue) and after (red) the faults are injected. The visual difference between the two signals that reveals the faults is in red. The fault on the left half is a mild phase shift of a small decrease in the wave frequency that spans 310 data points, and the other fault is a sudden time lag of 300 data points. The magnitude of both faults is close to the noise band of the original signal (blue), so the anomalous proportions (red) stay within the normal range of the original signal, and do not appear obvious to the common outlier detection methods.

Our technique was able to localize both injected faults through 4 rounds of fault localization. At the end of the 3rd round of analysis, the anomalous measurements that constitute the two faults

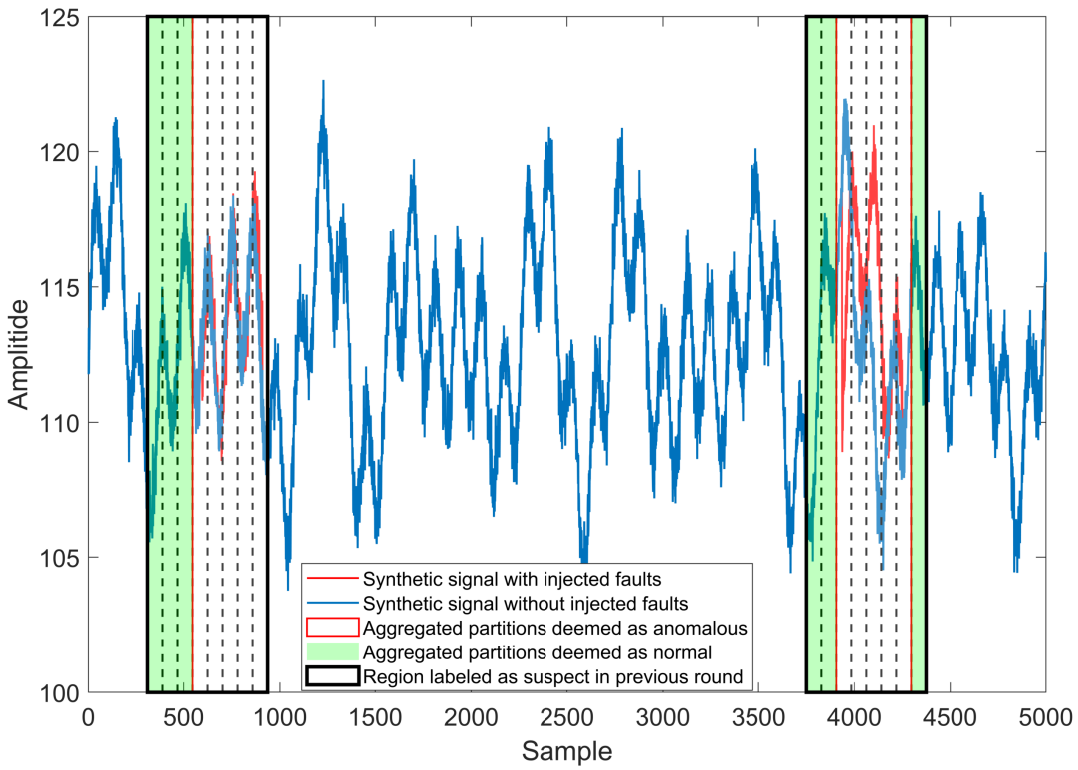

Figure 1: Example validation of the RFLP technique on one of the 100 synthetic signals. Two faults of distinct types (red) reside in the signal (blue) at different locations. Prior to the final round of partitioning and fault inferencing, the RFLP technique has successfully narrowed down the anomalous points that constitute the two faults to two suspect regions (black rectangle), which are then further subdivided in the final round (dashed line). Ultimately, 5 partitions in each region (bracketed by red rectangle) are deemed as anomalous, with the rest of the partitions flagged as normal (green shade).

have been narrowed down to two regions of the same size (black rectangles in Figure 1). The data outside the two regions are used to train a MSET model. In the final round of analysis, each region is divided into 8 partitions (indicated by the dashed line), resulting in a total of 16 partitions. After they are evaluated with the MSET model, and subsequently the fault inferencing process, 5 partitions in each regions are deemed as anomalous (aggregated and bracketed by red rectangle), while the other partitions are deemed as normal (green highlights). The aggregated partitions completely bracket the injected faults, demonstrating that the proposed RFLP technique functions as designed.

While the maximum number of rounds of fault localization needs to be pre-defined in our work, the number of model training iterations is not fixed, but adapts as the use case requires, which depends on the complexity and locations of the faults. For example, if the faults are present in both sides of the signal (Figure 1), then it is a challenging scenario because one fault resides in the training partition while another fault is inside the testing partition in the first round of analysis. This scenario typically requires more iterations of training-testing to discern both faults than other scenarios where the faults are close to each other. In our case, 10 MSET models were built for this example. Nevertheless, with the most lightweight model settings, this example took less than 30 sec to complete on an Intel workstation.

4.2 Case Study with Published Data

A total of 22 sensor signals are included in the SWaT dataset. The length of the time series is about 497k measurements for the training data, and 450k measurements for the testing data. The labels for the ground truth faults are available only for the testing data. The MSET technique introduced in Section 2 serves as the supervised anomaly detector in the case study.

It is worth noting that the use of a particular anomaly detector is not the primary focus of our works. Our method is designed to supplement any given anomaly detection algorithm with the goal of improving its performance by enhancing the quality of the training data. As such, the relative performance improvement achieved on a given anomaly detector is of primary interest in our study.

4.2.1 Validation on Unlabeled Training Data. We first showcase one of the SWaT training signals as an example in Figure 2. The

signal (blue) starts out from the global minimal value, and then almost monotonically increases, resembling an up-ramp, until at a later time when the signal values start to be range-bound. This up-ramp is present in all 22 signals at the same time, though with different magnitudes. Although the training dataset lacks explicit labeling information, the researchers of the SWaT system state in their work [5] that the system needs some time to stabilize and reach its normal steady state operation. The system stabilization period is typically a consequence of the system overcoming inertia-from static friction-until it reaches its dynamic equilibrium. Therefore, the system is not in a normal operating state during this time. By visualizing this signal, it is obvious that the up-ramp at the start of the data (a 5-hour period, highlighted by red shade) was part of the system's stabilization process. As a consequence, the measurements within this time period exhibit completely different characteristics and dynamics from the measurements under steady state. Including this time period in model training would bias the resulting model.

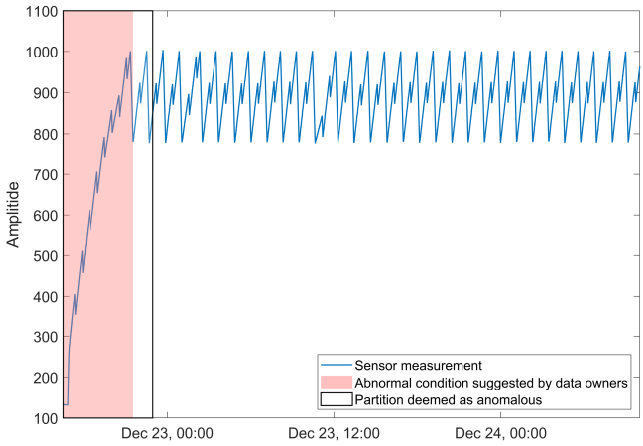

Figure 2: Measurements taken during the system's stabilization period (red shaded) and the anomalous partition (black bracket) identified by our technique on one of the SWaT training signals. Only the first 25% of the data is displayed for readability.

To proceed, we apply the RFLP technique to the dataset, and as a result, a total of 5 different models were trained[2]. In the final round of analysis, two anomalous partitions out of 16 partitions were flagged, which are aggregated and marked by a black rectangle in Figure 2 for illustration clarity. The output of the technique completely brackets the up-ramp, which is deemed successful per the descriptions about the system stabilization time in [5]. It is worthwhile to note that, although this up-ramp can easily be identified visually by human operators, our technique autonomously detects the abnormal measurements caused by the system stabilization so they can be excluded them from the training data without requiring expert intervention. The impact of having these abnormal measurements in the training data on the downstream supervised detection is further investigated in Section 4.2.2.

[2]models breakdown: 2 models in the first round, and 1 model each in the following three rounds, per Section 3.

Table 1: Performance comparison of supervised anomaly detection on SWaT testing data between the common approach (Case 1) and our proposed approach (Case 2)

	FAP	MAP
Case 1	0.0069	0.4771
Case 2	0.0073	0.1667

4.2.2 Benefits of Removing Abnoarmal Measurements from Training Data. To further point out the value of the RFLP technique with respect to the stated objective, two supervised detection use cases using SWaT data are formulated and benchmarked, in which the standard use of our method is compared to the commonly adopted approach under the assumption that the negative impact of unlabeled training data is negligible [1].

Case 1 presents the commonly adopted approach for supervised detection with unlabeled training data: a supervised model is trained on the entire SWaT training dataset—treating it as fault-free—and then evaluated using the testing data.

Case 2 describes our proposed approach for improving the supervised model: the region flagged as anomalous by the RFLP method is removed (i.e., the black rectangle in Figure 2), and the remaining measurements are used to train the supervised model, which is subsequently evaluated on the testing data.

To quantify how our method improves the performance of the supervised model, a set of benchmark metrics is needed. While there has been substantial research dedicated to developing new and robust benchmarks for evaluating anomaly detection algorithms[18, 19], this study adopts the conventional performance metrics: False Alarm Probability (FAP) and Missed Alarm Probability (MAP):

$$\begin{aligned} \text{FAP} &= \frac{\text{TP}}{N_{\text{fault}}}, \\ \text{MAP} &= \frac{\text{FP}}{N_{\text{total}}}, \end{aligned} \tag{12}$$

where TP and FP are true predictions and false predictions, respectively, and N_{fault} and N_{total} indicate the number of known anomalous samples and total number of samples, respectively. The FAP and MAP are computed for each signal in the testing data, and their averages across all signals are summarized in Table 1.

In both cases, FAP is small and nearly identical. The small difference can be attributed to minor statistical variations between the original and the RFLP-processed training data used in the models. However, in Case 2, the MAP improves by a large margin (31 percentage points) because a fault in form of a big step change that is present in 6 signals (one of them is shown in the center of Figure 3) was missed in Case 1 but was successfully captured in Case 2. Other faults in form of pulse and mean shift that were successfully detected in Case 1 were still detected in Case 2.

The cause of that severe but "obvious" step-change fault being missed in Case 1 is due to the large up-ramp in the beginning of the training data (red shade in Figure 2) not being excluded from the model. The step-change fault in the testing data consisting of a big up-ramp and a down-ramp was not flagged during the testing in Case 1, because the model had "learned" a similar pattern that

appears in the training data. This strengthens our assertion that the presence of unlabeled anomalies in the training data for a system can lead to the same types of fault signatures being missed during subsequent monitoring of the same system.

4.2.3 Additional Validation. Sections 4.2.1 and 4.2.2 validate the standard use of RPLP – remove anomalies in the training data and build a model with the remaining "clean" data, and then use the model to predict anomalies in the testing data. In this section, we apply the RFLP method to the labeled SWaT testing data to evaluate its performance from a different perspective—by comparing its output to the ground truth anomalies. Essentially, we treat the labeled testing data as if it were unlabeled training data, providing an additional validation of the RFLP method's ability to localize anomalies without prior labeling. This approach further verifies the capability of the RFLP technique to isolate faults from normal data.

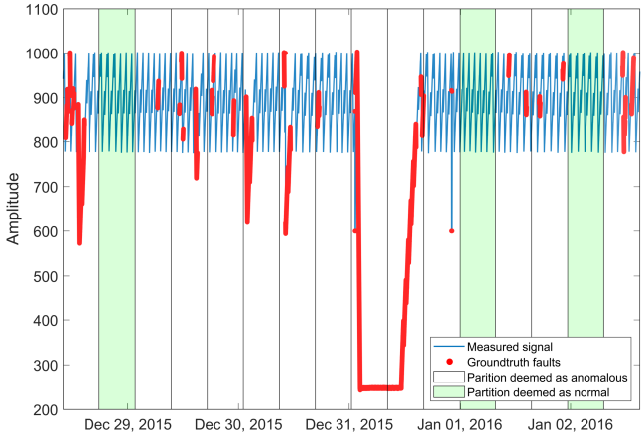

Figure 3: The comparison between the output of the RFLP technique and provided ground truth faults on one of the SWaT testing signals. Blue and red colors distinguish normal and anomalous measurements, and green and white highlights differentiate the partitions flagged as normal and anomalous, respectively, by the RFLP technique.

Figure 3 shows one of the SWaT testing signals in blue with the known ground truth faults in red. Green and white highlights differentiate the partitions deemed as normal and anomalous, respectively, by the RFLP technique. Due to the wide distribution of faults, the algorithm continued using the entire testing dataset to localize faults into narrower partitions. As a result, in the final round of analysis, the signal was evenly divided by 16 partitions. The anomalous partitions identified by the algorithm were able to bracket all ground truth faults. Notably, none of the 3 partitions that contained no ground truth faults were falsely labeled as anomalous, and all 13 partitions containing ground truth faults were correctly identified. The same results were observed in other 21 signals. Some normal measurements were included within the partitions flagged as anomalous, which relates to the maximum number of rounds $R_{\max}$ defined in our experiment. By progressing to more rounds of partitioning (i.e., smaller partition sizes) and fault inferencing, our procedure could increase the amount of anomaly-free data available to train a model, but that comes with a computational overhead trade-off, which is discussed in Section 4.3.3. Nevertheless, the results presented in Figure 3 provide further evidence that our method functions as intended.

4.3 Discussion

4.3.1 Contribution and Positioning. Although our work employs a supervised anomaly detector to localize faulty regions in unlabeled training data, its core contribution is a complementary preprocessing method designed to support supervised anomaly detection in multivariate time series. As such, it is positioned as an ancillary technique that improves the performance of the existing anomaly detectors, rather than operating as a standalone anomaly detector. Therefore, the method itself cannot substitute the anomaly detection model. To the best of the authors' knowledge, this method is new and has not been presented before, and the research work presented by Aubet et al. [1] is the closest competing approach that shares the same objective, though it is distinctly different from our work in that it focuses on the univariate signals and identifies the anomalous regions through an interpolation approach. The concept of our method can also be applied to the univariate time series, though we would have to develop a different recursive framework if a univariate kernel is employed. We were motivated to start out with multivariate time series because we found multivariate anomaly detection is more susceptible to ignoring the faults in the unlabeled training data than the univariate counterpart.

In this work, a multivariate anomaly detection technique (MSET) is selected as the supervised anomaly detector. However, the proposed method is readily adaptable to other ML models including regression based multivariate techniques. Ultimately, our technique is designed to enhance, rather than replace existing supervised anomaly detection detectors when labeled training data is unavailable.

4.3.2 Cost of Using Unlabeled Training Data in Supervised Detection. The RFLP technique employs a recursive procedure to localize faults in the unlabeled training data for supervised detection applications. As a consequence, the computational cost can be significant as the number of partitions grows. The alternative options are to use unlabeled training data or to use a manual, labor intensive process to identify anomalies. Using unlabeled training blindly often leads to many missed alarms causing expensive unplanned outages or maintenance. The missed alarms can even be catastrophic failure in safety-critical applications. Manually removing anomalies is labor intensive and often infeasible. The data users would need to 1) consult with the subject matter experts for the assets under surveillance to try to identify the possible faults, or 2) cross reference the Service Request log database for the assets under surveillance. Thus, we believe that the training compute cost of the RFLP technique is much cheaper than the other options for most applications. Also, the computational cost of our technique represents a one-time upfront expense for the use cases where the customer has no anomaly labeling. It does not increase the operational cost for the supervised detection downstream of the training, but does

significantly enhance the detection performance, as demonstrated in Section 4.2.2.

4.3.3 Limitations and Future Work. The limitation of the RFLP approach is the maximum number of rounds of partitioning $R_{\max}$ needs to be predetermined upfront. This introduces a trade-off between computational cost and the amount of "clean" training data recovered, warranting further discussion.

The overall strategy of the RFLP technique is to localize faults to one or more suspect regions through recursive rounds of partitioning. These suspect regions become narrower as the algorithm progresses, and by the time it progresses to the final round, the goal is to conservatively ensure that the faults have been narrowed down to one or several small regions. The rationale behind this strategy is that removing a small percentage of normal data along with the anomalous region does not penalize MAPs, as sufficient clean data remains available for training the model. However, leaving a small percentage of abnormal data in the region flagged as normal can negatively impact MAPs. Therefore, the algorithm is expected to terminate conservatively, allowing some normal observations to remain within the region flagged as anomalous to minimize the risk of retaining undetected faults. Further shrinking the suspect region through additional rounds of partitioning would increase computational cost while offering only marginal benefit, as the small amount of additional clean data recovered is unlikely to substantially enhance model performance (per the FAP metric in Table 1).

Therefore, an optimal number of rounds $R_{\max}$ needs to be determined to halt the recursive subdividing process and save compute cost while ensuring that the suspect region in the beginning of the final round is longer than the fault duration. Although a universally optimal $R_{\max}$ couldn't be derived in this study, we have found after empirical experimentation across many permutations of fault types, duration, and severity, that a $R_{\max}$ of 4 is usually adequate to localize the anomalous behavior without excessively conservative inclusion of normal observations in the final anomalous partitions. This $R_{\max}$ has been validated on a variety of datasets including the synthetic data with small sample size (5k) and real world data with much bigger sample size (497k) that are used in this work. Nevertheless, a logical extension to this work is to derive an adaptive $R_{\max}$ for any given multivariate dataset in an automated fashion.

5 Conclusions

In this paper, we address the challenge posed by the lack of labeled training data and highlight the impact of blindly using unlabeled training data in supervised anomaly detection. We present a preprocessing strategy that mitigates the challenges by localizing the most likely anomalies within the unlabeled training data to small regions through a recursive process of partitioning and fault inferencing. By excluding these suspect regions from training, the resulting supervised model demonstrates significantly improved performance. A mathematical multivariate technique is used as the supervised anomaly detector in this work. However, the proposed method is readily adaptable to other anomaly detection models, including heuristic based models that have been used in many research work. The capability of our technique has been validated on both synthetic and published datasets with ground truth fault signatures available. For the published dataset, we demonstrate that our technique effectively localizes the abnormal regions in the unlabeled training data, and as a result, the anomaly detection performance is improved with the Missed Alarm Probability reduced by 31%. While our method is capable of narrowing down the faults to regions that contain faults, it does not produce definitive anomaly decisions at the timestamp level. Therefore, it is not intended to function as a standalone anomaly detection model, but rather as an ancillary technique that enhances the performance of supervised anomaly detection. In summary, our contribution complements—rather than competes with—existing anomaly detection algorithms, and offers a practical solution for handling unlabeled time series training data in supervised multivariate machine learning applications.

References

[1] François-Xavier Aubet, Daniel Zügner, and Jan Gasthaus. 2021. Monte Carlo EM for Deep Time Series Anomaly Detection. *arXiv preprint arXiv:2112.14436* (2021).

[2] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. 2020. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3395–3404.

[3] Mohammad Braei and Sebastian Wagner. 2020. Anomaly detection in univariate time-series: A survey on the state-of-the-art. *arXiv preprint arXiv:2004.00433* (2020).

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41, 3 (2009), 1–58.

[5] Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. 2016. A dataset to support research in the design of secure water treatment systems. In *International conference on critical information infrastructures security*. Springer, 88–99.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[7] KC Gross, RM Singer, SW Wegerich, JP Herzog, R VanAlstine, and F Bockhorst. 1997. *Application of a model-based fault detection system to nuclear plant signals*. Technical Report. Argonne National Lab., IL (United States).

[8] Kenny Gross and Guang Chao Wang. 2019. AI Decision Support Prognostics for IoT Asset Health Monitoring, Failure Prediction, Time to Failure. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 244–248.

[9] Kenny C Gross and Mengying Li. 2017. Method for Improved IoT Prognostics and Improved Prognostic Cyber Security for Enterprise Computing Systems. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*. The Steering Committee of The World Congress in Computer Science, Computer …, 328–334.

[10] Kenny C Gross and Wendy Lu. 2002. Early Detection of Signal and Process Anomalies in Enterprise Computing Systems.. In *ICMLA*. 204–210.

[11] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 387–395.

[12] R Kozma, M Kitamura, M Sakuma, and Y Yokoyama. 1994. Anomaly detection by neural network models and statistical time series analysis. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, Vol. 5. IEEE, 3207–3210.

[13] Kwei-Herng Lai, Daochen Zha, Guanchu Wang, Junjie Xu, Yue Zhao, Devesh Kumar, Yile Chen, Purav Zumkhawaka, Minyang Wan, Diego Martinez, et al. 2021. Tods: An automated time series outlier detection system. In *Proceedings of the aaai conference on artificial intelligence*, Vol. 35. 16060–16062.

[14] Ming-Chang Lee, Jia-Chun Lin, and Ernst Gunnar Gran. 2020. RePAD: real-time proactive anomaly detection for time series. *arXiv preprint arXiv:2001.08922* (2020).

[15] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. 2019. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In *International Conference on Artificial Neural Networks*. Springer, 703–716.

[16] Junshui Ma and Simon Perkins. 2003. Time-series novelty detection using one-class support vector machines. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, Vol. 3. IEEE, 1741–1745.

[17] Tahereh Masoumi and Kenny C Gross. 2016. SimSPRT-II: Monte Carlo simulation of sequential probability ratio test algorithms for optimal prognostic performance. In *2016 International Conference on Computational Science and Computational*

Intelligence (CSCI). IEEE, 496–501.

[18] John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S Tsay, Aaron Elmore, and Michael J Franklin. 2022. Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection. *Proceedings of the VLDB Endowment* 15, 11 (2022), 2774–2787.

[19] John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S Tsay, Themis Palpanas, and Michael J Franklin. 2022. TSB-UAD: an end-to-end benchmark suite for univariate time-series anomaly detection. *Proceedings of the VLDB Endowment* 15, 8 (2022), 1697–1711.

[20] Kamran Shaukat, Talha Mahboob Alam, Suhuai Luo, Shakir Shabbir, Ibrahim A Hameed, Jiaming Li, Syed Konain Abbas, and Umair Javed. 2021. A review of time-series anomaly detection techniques: A step to future perspectives. In *Future of Information and Communication Conference*. Springer, 865–877.

[21] Ralph M Singer, Kenny C Gross, James P Herzog, Ronald W King, and Stephen Wegerich. 1997. *Model-based nuclear power plant monitoring and fault detection: Theoretical foundations*. Technical Report. Argonne National Lab., IL (United States).

[22] Akash Singh. 2017. Anomaly detection for temporal data using long short-term memory (lstm).

[23] Abraham Wald. 2004. *Sequential analysis*. Courier Corporation.

[24] GC Wang and KC Gross. 2018. Telemetry parameter synthesis system for enhanced tuning and validation of machine learning algorithmics. In *IEEE 2018 Intn'l Symposium on Internet of Things & Internet of Everything (CSCI-ISOT), Las Vegas, NV*.

[25] Renjie Wu and Eamonn Keogh. 2021. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering* (2021).