Graphical Abstract

Adaptable and Interpretable Framework for Anomaly Detection in Streaming Energy Systems

Marek Wadinger, Michal Kvasnica

Highlights

Adaptable and Interpretable Framework for Anomaly Detection in Streaming Energy Systems

Marek Wadinger, Michal Kvasnica

- Combines interpretability with adaptation to change points in single anomaly detection system
- Isolates root cause of anomalies considering relationships between features
- Demonstrates interpretability by providing process limits for each feature
- Provides comparable detection accuracy to established general methods

Adaptable and Interpretable Framework for Anomaly Detection in Streaming Energy Systems

Marek Wadinger[a,*], Michal Kvasnica[a]

[a]*Institute of Information Engineering, Automation and Mathematics, Slovak University of Technology in Bratislava, Radlinskeho 9, Bratislava, 812 37, Bratislava, Slovakia*

Abstract

In this paper, we introduce an Adaptable and Interpretable Framework for Anomaly Detection (AID) designed for streaming energy systems utilizing IoT devices. AID leverages adaptive conditional probability distribution modeling to capture the normal operation of dynamic systems and isolate root causes of anomalies. The framework dynamically updates parameter of multivariate Gaussian distribution, allowing it to adapt to non-stationarity. AID interprets anomalies as significant deviations from conditional probability, encompassing interactions as well as both spatial and temporal irregularities by exposing them as features. Additionally, dynamic process limits are drawn to pinpoint root causes. The framework also alerts individual signal as outliers in sampling. Two real-world case studies showcase AID's capabilities. The first study focuses on Battery Energy Storage Systems (BESS), demonstrating AID's effectiveness in capturing system anomalies, providing less conservative signal limits, and leveraging a physical model for temperature anomaly detection. The second case study delves into monitoring temperature profiles of battery modules, where AID successfully identifies hardware faults, emphasizing its importance in energy storage system safety and profitability. A benchmark evaluation on industrial data shows that AID delivers comparable results to other self-learning adaptable anomaly detection methods, with the added advantage of root cause isolation.

Keywords: Anomaly detection, Root cause isolation, Iterative learning control, Statistical learning, IoT

*

Email address: marek.wadinger@stuba.sk (Marek Wadinger)
URL: uiam.sk/~wadinger (Marek Wadinger)

1. Introduction

Anomaly detection systems play a critical role in risk-averse systems by identifying abnormal patterns and adapting to novel expected patterns in data. These systems are particularly vital in the context of Internet of Things (IoT) devices that continuously stream high-fidelity data to control units.

In this rapidly evolving field with long-spanning roots, Chandola et al. conducted an influential review of prior research efforts across diverse application domains (Chandola et al. (2009)). Recent studies have underscored the need for holistic and tunable anomaly detection methods accessible to operators (Laptev et al. (2015); Kejariwal (2015); Cook et al. (2020)).

Cook et al. denote substantial aspects that pose challenges to anomaly detection in IoT, including the temporal, spatial, and external context of measurements, multivariate characteristics, noise, and nonstationarity (Cook et al. (2020)). To address these complexities, Zhang et al. have successfully employed spatially distributed sensors and time-relative modulation. Their approach has proven effective, particularly in the context of complex nonlinear systems, offering potential solutions to some of the challenges posed by IoT data (Zhang et al. (2024)). Huang et al., on the other hand, tackled the problems of detecting global outliers, local outliers, and outlier clusters simultaneously. Their proposed approach, based on density estimation, relies on the notion that density distributions should exhibit minimal variations in local areas. To achieve this, they introduce a novel turning ratio metric, which reduces reliance on hyperparameters and enhances anomaly detection (Huang et al. (2023)).

Additionally, feature engineering techniques play a crucial role in capturing contextual properties and enhancing anomaly detection performance (Fan et al. (2019)). However, it's worth noting that feature engineering may introduce categorical variables and significantly increase dimensionality of the data, requiring specific methods for handling large data, sizeable data storage, and substantial computational resources (Talagala et al. (2021)). Recently, Li et al. introduced an attribute-weighted outlier detection algorithm, designed for high-dimensional datasets with mixtures of categorical and numerical data. Their approach assigns different weights to individual attributes based on their importance in anomaly detection and uses these weights to calculate distances between data points. Notably, Li et

al. demonstrated the superior performance of their algorithm compared to state-of-the-art methods (Li and Liu (2024)). Another strategy for handling high-dimensional data involves using deep learning methods with synthetic normal data to enhance the detection of outliers with subtle deviations, as proposed in Du et al. (2024).

Nevertheless, the presence of nonstationarity, often stemming from concept drift (a shift in data patterns due to changes in statistical distribution) and change points (permanent alterations in system state), presents a substantial challenge (Salehi and Rashidi (2018)). In practical scenarios, those changes tend to be unpredictable in both their spatial and temporal aspects.Consequently, they require systems with solid outlier rejection capabilities of intelligent tracking algorithms (Barbosa Roa et al. (2019)). This underscores the critical importance of an anomaly detection method's ability to adapt to evolving data structures, especially in long-term deployments. Nevertheless, as (Tartakovsky et al. (2013)) remarked, the immediate detection is not a feasible option unless there is a high tolerance for false alarms.

The former scalability problem now introduces a significant latency in detector adaptation (Wu et al. (2021)). Incremental learning methods allowed adaptation while restraining the storage of the whole dataset. The supervised operator-in-the-loop solution offered by Pannu et al. showed the detector's adaptation to data labeled on the flight (Pannu et al. (2012)). Others approached the problem as sequential processing of bounded data buffers in univariate signals (Ahmad et al. (2017)) and multivariate systems (Bosman et al. (2015)).

1.1. Related Work

Recent advances in anomaly detection have broadened its scope to include root cause identification governed by the development of explanatory methods capable of diagnosing and tracking faults across the system. Studies can be split into two groups of distinct approaches. The first group approaches explainability as the importance of individual features (Carletti et al. (2019); Nguyen et al. (2019); Amarasinghe et al. (2018)). Those studies allow an explanation of novelty by considering features independently. The second group uses statistical learning creating models explainable via probability. For instance, integration of variational Bayesian inference probabilistic graph neural network allowed Zhang et al. to model the posterior distribution of sensor dependency for gas leakage localization on unlabeled

data (Zhang et al. (2023)). Yang et al. recently proposed a Bayesian network (BN) for fault detection and diagnosis. In this BN, individual nodes of the network represent normally distributed variables, whereas the multiple regression model defines weights and relationships. Using the predefined structure of the BN, the authors propose an offline training with online detection and diagnosis (Yang et al. (2022)).

Given the infrequent occurrence of anomalies and their potential absence in training data, the incorporation of synthetic data or feature extraction for various detected events emerges to assist diagnosis of the system. Brito et al. designed synthetic faults based on expert knowledge and introduced them into a transfer learning classifier to exploit faults in rotating machinery, with a subsequent explanation layer (Brito et al. (2023)). Conversely, We et al. leveraged feature selection to expose various types of abnormal behavior. The team presents competetive performance while using change in relationships to provide causal inference (Wu et al. (2024)).

However, it is crucial to underscore that offline training, as previously emphasized, is inherently inadequate when it comes to adapting to anticipated novel patterns, rendering it unsuitable for sustained, long-term operation on IoT devices.

This paper emphasizes the importance of combining adaptability in interpretable anomaly detection and proposes a method that addresses this challenge. Here we report the discovery and characterization of an adaptive anomaly detection method for streaming IoT data. The ability to diagnose multivariate data while providing root cause isolation, inherent in the univariate case, extends our previous contribution to the field as presented in (Wadinger and Kvasnica (2023)). The proposed algorithm represents a general method for a broad range of safety-critical systems where anomaly diagnosis and identification are paramount.

1.2. Novelty of proposed approach

The idea of using statistical outlier detection is well-established. We highlighting impactful contributions of (Yamanishi and Takeuchi (2002)) and (Yamanishi et al. (2004)). The authors propose a method for detecting anomalies in a time series. The method is based on the assumption that the continuous data is generated by a mixture of Gaussian distributions, while discrete data is modeled as histogram density. The authors solve the problem of change point detection as well. However, the adaptation system is unaware of such changes, making the moving window the only source of adaptation. Our

self-supervised approach offers intelligent adaptation w.r.t. detected change points. Moreover, the author of the study does not attempt to isolate the root cause of the anomaly. We do so by computing the conditional probability of each measurement given the rest of the measurements and drawing limits defining the normal event probability threshold.

A limited number of studies have focused on adaptation and interpretability within the framework of anomaly detection. Two recent contributions in this area are (Steenwinckel (2018)) and (Steenwinckel et al. (2021)). In (Steenwinckel (2018)), the authors emphasize the importance of combining prior knowledge with a data-driven approach to achieve interpretability, particularly concerning root cause isolation. They propose a novel approach that involves extracting features based on knowledge graph pattern extraction and integrating them into the anomaly detection mechanism. This graph is subsequently transformed into a matrix, and adaptive region-of-interest extraction is performed using reinforcement learning techniques. To enhance interpretability, a Generative Adversarial Network (GAN) reconstructs a new graphical representation based on selected vectors. However, it's important to note that the validation of this idealized approach is pending further investigation. Lately, (Steenwinckel et al. (2021)) introduced a comprehensive framework for adaptive anomaly detection and root cause analysis in data streams. While the adaptation process is driven by user feedback, the specific mechanism remains undisclosed. The authors present an interpretation of their method through a user dashboard, featuring visualizations of raw data. This dashboard is capable of distinguishing between track-related problems and train-related issues, based on whether multiple trains at the same geographical location approach the anomaly. Meanwhile, our attempts aim to develop a self-supervised method capable of learning without human supervision which is often limited in time and poses significant delays in adaptation, while interpretation offers straightforward statistical reasoning and root cause isolation.

1.3. Validation

Two case studies show that our proposed method, based on dynamic joint normal distribution, has the capacity to explain novelties, isolate the root cause of anomalies, and allow adaptation to change points, advancing recently developed anomaly detection techniques for long-term deployment and cross-domain usage. We observe similar detection performance, albeit with lower scalability, when comparing our approach to well-established unsupervised

anomaly detection methods in streamed data which create a bedrock for many state-of-the-art contributions, such as One-Class SVM (Amer et al. (2013); Liu et al. (2014); Krawczyk and Woźniak (2015); Miao et al. (2019); Gözüaçık and Can (2021)), and Half-Space Trees (Wetzig et al. (2019); Lyu et al. (2020)).

1.4. Broader Impact

Potential applications of the proposed method are in the field of energy storage systems, where the ability to detect anomalies and isolate their root cause, whilst adapting to changes in operation and environment, is crucial for the safety of the system. The proposed method is suitable for the existing infrastructure of the system, allowing detection and diagnosis of the system based on existing data streams. The dynamic process limits allow operational metrics monitoring, making potential early detection and prevention easier. Using adaptable methods without interpretability, on the other hand, may pose safety risks and lower total financial benefits, as the triggered false alarms may need to be thoroughly analyzed, resulting in prolonged downtimes.

The main contribution of the proposed solution to the developed body of research is that it:

- Enriches interpretable anomaly detection with adaptive capabilities
- Identifies systematic outliers and root cause
- Uses self-learning approach on streamed data
- Utilizes existing IT infrastructure
- Establishes dynamic limits for signals

1.5. Paper Organization

The rest of the paper is structured as follows: We begin with the problem and motivation in **Section 1**, providing context. Next, in **Section 2**, we lay the theoretical groundwork. Our proposed adaptive anomaly detection method is detailed in **Section 3**. We then demonstrate real-world applications in **Section 4**. Finally, we conclude the paper in **Section 5**, summarizing findings and discussing future research directions.

2. Preliminaries

In this section, we present the fundamental ideas that form the basis of the developed approach. Subsection 2.1 explains Welford's online algorithm, which can adjust distribution to changes in real-time. Subsection 2.2 proposes a two-pass implementation that can reverse the impact of expired samples. The math behind distribution modeling in Subsection 2.3 establishes the foundation for the Gaussian anomaly detection model discussed in Subsection 2.5, followed by conditional probability computation in Subsection 2.4. The last subsection of the preliminaries is devoted to the definition of anomalies.

2.1. Welford's Online Algorithm

Welford introduced a numerically stable online algorithm for calculating mean and variance in a single pass. The algorithm allows the processing of IoT device measurements without the need to store their values Welford (1962).

Given measurement x_i where $i = 1, ..., n$ is a sample index in sample population n, the corrected sum of squares S_n is defined as

$$S_n = \sum_{i=1}^{n} (x_i - \bar{x}_n)^2, \tag{1}$$

with the running mean $\bar{x}_n$ defined as previous mean $\bar{x}_{n-1}$ weighted by proportion of previously seen population $n-1$ corrected by current sample as

$$\bar{x}_n = \frac{n-1}{n}\bar{x}_{n-1} + \frac{1}{n}x_n = \bar{x}_{n-1} + \frac{x_n - \bar{x}_{n-1}}{n}. \tag{2}$$

Throughout this paper, we consider the following formulation of an update to the corrected sum of squares:

$$S_n = S_{n-1} + (x_n - \bar{x}_{n-1})(x_n - \bar{x}_n), \tag{3}$$

as it is less prone to numerical instability due to catastrophic cancellation. Finally, the corresponding unbiased variance is

$$s_n^2 = \frac{S_n}{n-1}. \tag{4}$$

This implementation of the Welford method requires the storage of three scalars: $\bar{x}_{n-1}$; n; S_n.

2.2. Inverse Welford's Algorithm

Based on (2), it is clear that the influence of the latest sample over the running mean decreases as the population n grows. For this reason, regulating the number of samples used for sample mean and variance computation has crucial importance over adaptation. Given access to the instances used for computation and expiration period $t_\mathrm{e} \in \mathbb{N}_0^{n-1}$, reverting the impact of x_{n-t_e} can be written as follows

$$S_{n-1} = S_n - (x_{n-t_\mathrm{e}} - \bar{x}_{n-1})(x_{n-t_\mathrm{e}} - \bar{x}_n), \tag{5}$$

where the reverted mean is given as

$$\bar{x}_{n-1} = \frac{n}{n-1}\bar{x}_n - \frac{1}{n-1}x_{n-t_\mathrm{e}} = \bar{x}_n - \frac{x_{n-t_\mathrm{e}} - \bar{x}_n}{n-1}. \tag{6}$$

Finally, the unbiased variance follows the formula:

$$s_{n-1}^2 = \frac{S_{n-1}}{n-2}. \tag{7}$$

2.3. Statistical Model of Multivariate System

Multivariate normal distribution generalizes the multivariate systems to the model where the degree to which variables are related is represented by the covariance matrix. Gaussian normal distribution of variables is a reasonable assumption for process measurements, as it is a common distribution that arises from stable physical processes measured with noise. The general notation of multivariate normal distribution is:

$$\mathbf{X} \ \sim \ \mathcal{N}_k(\boldsymbol{\mu},\, \boldsymbol{\Sigma}), \tag{8}$$

where k-dimensional mean vector is denoted as $\boldsymbol{\mu} = (\bar{x}_1, ..., \bar{x}_k)^T \ \in \mathbb{R}^k$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{k\times k}$ is the $k \times k$ covariance matrix, where k is the index of last random variable.

The probability density function (PDF) $f(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ of multivariate normal distribution is denoted as:

$$f(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{k/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}, \tag{9}$$

where $\boldsymbol{x}$ is a k-dimensional vector of measurements x_i at time i, $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$, and $\boldsymbol{\Sigma}^{-1}$ is the inverse of $\boldsymbol{\Sigma}$.

The cumulative distribution function (CDF) of a multivariate Gaussian distribution describes the probability that all components of the random matrix $\boldsymbol{X}$ take on a value less than or equal to a particular point $\boldsymbol{x}$ in space, and can be used to evaluate the likelihood of observing a particular set of measurements or data points. The CDF is often used in statistical applications to calculate confidence intervals, perform hypothesis tests, and make predictions based on observed data. In other words, it gives the probability of observing a random vector that falls within a certain region of space. The standard notation of CDF is as follows:

$$F(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \int_{-\infty}^{\boldsymbol{x}} f(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \mathrm{d}\boldsymbol{x}, \tag{10}$$

where $\mathrm{d}\boldsymbol{x}$ denotes the integration over all k dimensions of $\boldsymbol{x}$.

As the equation (10) cannot be integrated explicitly, an algorithm for numerical computation was proposed in Genz (2000).

Given the PDF, we can also determine the value of $\boldsymbol{x}$ that corresponds to a given quantile q using a numerical method for inversion of CDF (ICDF) often denoted as percent point function (PPF) or $F(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})^{-1}$. An algorithm that calculates the value of the PPF for univariate normal distribution is reported below as Algorithm 1.

Algorithm 1 Percent-Point Function for Normal Distribution

Input: quantile q, sample mean $\bar{x}_n$ (2), sample variance s_n^2 (4)
Output: threshold value $\tilde{x}_{\mathrm{q},n}$

Initialisation :
$f \leftarrow 10$; $l \leftarrow -f$; $r \leftarrow f$;
LOOP Process
while $F(l; \bar{x}_n, s_n^2) > 0$ **do**
  $r \leftarrow l$;
  $l \leftarrow lf$;
end while
while $F_X(r) - q < 0$ **do**
  $l \leftarrow r$;
  $r \leftarrow rf$;
end while
$\tilde{x}_{\mathrm{q},n} = \arg\min_{x_n} \| F(x_n; \bar{x}_n, s_n^2) - q \|$ s.t. $l \leq x_n \leq r$
return $\tilde{x}_{\mathrm{q},n} \sqrt{s_n^2} + \bar{x}_n$

The Algorithm 1 for PPF computation is solved using an iterative root-finding algorithm such as Brent's method Brent (1972).

2.4. Conditional Probability Distribution

Considering that we observe particular vector $\boldsymbol{x}_i$, we can update probability distributions, calculated according to the rules of conditional probability, of individual measurements within the vector given the rest of the measurements in $\boldsymbol{x_i}$. Let's assume multivariate normal distribution (8) and without loss of generality, that the vector $\boldsymbol{x}_i$ can be partitioned into subset variable x_a, and complement vector $\boldsymbol{x_b}$ as follows

$$\boldsymbol{x}_i = \begin{bmatrix} x_a \\ \boldsymbol{x_b} \end{bmatrix} \text{ with dimensions } \begin{bmatrix} 1 \times 1 \\ (k-1) \times 1 \end{bmatrix}, \tag{11}$$

where $a = 1, ..., k$ and $\boldsymbol{b} = \{1, 2, \ldots, k\}$ where $a \notin \boldsymbol{b}$. This partitioning allows us to define block-wise mean and covariance as follows:

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_a \\ \boldsymbol{\mu_b} \end{bmatrix} \text{ with dimensions } \begin{bmatrix} 1 \times 1 \\ (k-1) \times 1 \end{bmatrix}, \tag{12}$$

and

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma^2_{aa} & \boldsymbol{\Sigma}_{a\boldsymbol{b}} \\ \boldsymbol{\Sigma}_{\boldsymbol{b}a} & \boldsymbol{\Sigma}_{\boldsymbol{bb}} \end{bmatrix} \text{ with dimensions } \begin{bmatrix} 1 \times 1 & 1 \times (k-1) \\ (k-1) \times 1 & (k-1) \times (k-1) \end{bmatrix}. \tag{13}$$

where a and $\boldsymbol{b}$ represent distinct components within the vector.

Subsequently, we can derive the conditional distribution of any subset variable x_a, given the complementary vector $\boldsymbol{x_b}$. This conditional distribution conforms to a univariate normal distribution, characterized by:

$$X_a \mid \mathbf{X}_{\boldsymbol{b}} \sim \ \mathcal{N}(\mu_{a|\boldsymbol{b}},\, \sigma^2_{a|\boldsymbol{b}}). \tag{14}$$

where $\mu_{a|\boldsymbol{b}}$ denotes the conditional mean and $\sigma^2_{a|\boldsymbol{b}}$ represents the conditional variance. These crucial parameters can be computed using the Schur complement.

For a general matrix M expressed as:

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \tag{15}$$

the Schur complement of the block matrix M is denoted as:

$$M \mid D = A - BD^{-1}C. \tag{16}$$

Applying Equation (16), we can calculate the conditional variance $\sigma^2_{a|\boldsymbol{b}}$ using the covariance matrix notation from Equation (13) as follows:

$$\sigma^2_{a|\boldsymbol{b}} = \sigma^2_{aa} - \boldsymbol{\Sigma}_{a\boldsymbol{b}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{bb}}\boldsymbol{\Sigma}_{\boldsymbol{b}a}, \tag{17}$$

while the conditional mean, denoted as $\mu_{a|\boldsymbol{b}}$, is determined by:

$$\mu_{a|\boldsymbol{b}} = \mu_a + \boldsymbol{\Sigma}_{a\boldsymbol{b}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{bb}}(\boldsymbol{x}_{\boldsymbol{b}} - \boldsymbol{\mu}_b). \tag{18}$$

It is important to note that $\boldsymbol{\Sigma}^{-1}_{\boldsymbol{bb}}$ in Equations (17) and (18) signifies the inverse of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{bb}}$. Thus, the conditional variance $\sigma^2_{a|\boldsymbol{b}}$ essentially represents the Schur complement of $\boldsymbol{\Sigma}_{\boldsymbol{bb}}$ within the overall covariance matrix $\boldsymbol{\Sigma}$.

2.5. Gaussian Anomaly Detection

From a viewpoint of statistics, outliers are commonly denoted as values that significantly deviate from the mean. Under the assumption that the spatial and temporal characteristics of a system, observed over a moving window, can be suitably represented as normally distributed features, we assert that any anomaly can be identified as an outlier.

From a statistical viewpoint, outliers can be denoted as values that significantly deviate from the mean. Assuming that the spatial and temporal characteristics of the system over the moving window can be encoded as normally distributed features, we can claim, that any anomaly may be detected as an outlier.

In empirical fields like machine learning, the three-sigma rule (3σ) provides a framework for characterizing the region of a distribution within which normal values are expected to fall with high confidence. This rule renders approximately 0.265% of values in the distribution as anomalous.

The 3σ rule establishes the probability that any sample x_a of a random vector X lies within a given CDF over a semi-closed interval as the distance from the conditional mean $\mu_{a|\boldsymbol{b}}$ of 3 conditional variances $\sigma^2_{a|\boldsymbol{b}}$ and gives an approximate value of q as

$$q = P\{|x_a - \mu_{a|\boldsymbol{b}}| < 3\sigma^2_{a|\boldsymbol{b}}\} = 0.99735. \tag{19}$$

Utilizing a probabilistic model of normal behavior, we can determine threshold values x_{l} and x_{u} corresponding to the closed interval of the CDF

where this probability is established. The inversion of Equation (10) facilitates this calculation, yielding:

$$x_{\mathrm{l}} = F((1 - P\{|x_a - \mu_{a|\boldsymbol{b}}| < 3\sigma^2_{a|\boldsymbol{b}}\}); \mu_{a|\boldsymbol{b}}, \sigma^2_{a|\boldsymbol{b}})^{-1}, \tag{20}$$

for the lower limit, and

$$x_{\mathrm{u}} = F((P\{|x_a - \mu_{a|\boldsymbol{b}}| < 3\sigma^2_{a|\boldsymbol{b}}\}); \mu_{a|\boldsymbol{b}}, \sigma^2_{a|\boldsymbol{b}})^{-1}, \tag{21}$$

for the upper limit. These lower and upper limits together form vectors $\boldsymbol{x}_{\mathrm{l}}$ and $\boldsymbol{x}_{\mathrm{u}}$, respectively, defining the region of normal system operation. This region is conceptualized as a hypercube in the feature space, with each dimension bounded by the corresponding feature limits, as computed using Equations (20) and (21) for all $a = 1, ..., k;\ \boldsymbol{b} = \{1, 2, \ldots, k\}$ where $a \notin \boldsymbol{b}$.

Such threshold is computed for each feature of the system, resulting in a vector of lower and upper limits. The vector of limits is used to define the region of normal operation of the system. The region is defined as a hypercube in the feature space, where each dimension is defined by the limits of the corresponding feature.

The predicted state of the system, denoted as y_i, and the presence of anomalies in signals $\boldsymbol{y}_{\mathrm{s},i}$ at time i are determined based on the maximum distance of observations from the center of the hypercube. The center of the hypercube corresponds to the vector of conditional means $\mu_{a|\boldsymbol{b}}$ with respect to other features. The calculation of this distance involves the cumulative distribution function (CDF) of observations and conditional distributions, as follows:

$$F(x_a; \mu_{a|\boldsymbol{b}}, \sigma^2_{a|\boldsymbol{b}}) : a = 1, ..., k;\ \boldsymbol{b} = \{1, 2, \ldots, k\} \text{ where } a \notin \boldsymbol{b}. \tag{22}$$

Subsequently, anomalies in individual inputs are defined as follows:

$$\boldsymbol{y}_{\mathrm{s},i} = \begin{cases} 0 & \text{if } T \leq (22) \\ 1 & \text{if } T > (22), \end{cases} \tag{23}$$

where T represents a threshold that distinguishes between normal signal measurement ($\boldsymbol{y}_{\mathrm{s},i} = 0$) and abnormal ($\boldsymbol{y}_{\mathrm{s},i} = 1$).

For the predicted state of the system, the maximum value from $\boldsymbol{y}_{\mathrm{s},i}$ is considered, resulting in:

$$y_i = \begin{cases} 1 & \text{if } 1 \in \boldsymbol{y}_{\mathrm{s},i} \\ 0 & \text{otherwise,} \end{cases} \tag{24}$$

defining the discrimination boundary between system operation where $y_i = 0$ indicates normal system operation, and $y_i = 1$ indicates anomalous operation.

2.6. Anomaly Definition

In the realm of data analysis, anomalies are conspicuous deviations from the anticipated patterns within a dataset. Traditionally, the task of anomaly detection has relied upon unsupervised methodologies, wherein the identification of "outliers" entails the comparison of data points in both temporal and spatial contexts. This approach, often referred to as point-wise anomaly detection, classifies a data point as an anomaly when it exhibits significant dissimilarity from its neighboring data points (Iglesias Vázquez et al. (2023)).

The concept of point anomalies, influenced by factors such as temporal and spatial aspects, can be further categorized into conditional and contextual anomalies (Ruff et al. (2021)).

Nevertheless, this conventional method may not be suitable for scenarios characterized by collective anomalies, where clusters of abnormal data points coexist. A more pragmatic approach defines anomalies as deviations from established "normal" patterns, resembling the principles of semi-supervised learning. Change point detection, in a similar vein, can be regarded as a relative approach that takes into account the varying dynamics of changes, whether they occur gradually or abruptly (Iglesias Vázquez et al. (2023)).

It is imperative to recognize that the interpretation of anomalies, outliers, and novelties can vary upon the application. Anomalies typically garner significant attention, while outliers are often treated as undesirable noise and are typically excluded during data preprocessing. Novelties, on the other hand, signify new observations that necessitate model updates to adapt to an evolving environment (Ruff et al. (2021)).

Notwithstanding the differences in terminology, methods employed for the identification of data points residing in low-probability regions, irrespective of whether they are referred to as "anomaly detection," "outlier detection," or "novelty detection," share fundamental similarities (Iglesias Vázquez et al. (2023)).

3. Novelty Detection and Interpretation Framework

In this section, we propose an adaptive and interpretable detection framework for multivariate systems with streaming IoT devices. This approach models the system as a dynamic joint normal distribution, enabling it to effectively adapt to pervasive nonstationary effects on processes. Our method handles various factors, including change points, concept drift, and seasonal effects. Our primary contribution lies in the fusion of an adaptable self-supervised system with root cause identification capabilities. This combination empowers the online statistical model to diagnose anomalies through two distinct avenues. Firstly, it employs conditional probability calculations to assess the system's operating conditions' normality. Secondly, it identifies outliers within individual signal measurements and features based on dynamic alert-triggering process limits. In the following sections, we describe our proposed methodology across three subsections. The initial subsection delves into the process of initializing the model's parameters. The subsequent section describes online training and adaptation, while the final subsection expounds upon the model's detection and diagnostic capabilities. For a concise representation of the proposed method, Algorithm 2 is provided.

3.1. Model Parameters Initialization

The model initialization is governed by defining two tunable hyperparameters of the model: the expiration period (t_e) and the threshold (T). The expiration period determines the window size for time-rolling computations, impacting the proportion of outliers within a given timeframe, and directly influencing the relaxation (with a longer expiration period) or tightening (with a shorter expiration period) of dynamic signal limits. Additionally, we introduce a grace period, which defaults to $3/4t_\mathrm{e}$, allowing for model calibration. During this grace period, system anomalies are not flagged to prevent false positives and speed up self-supervised learning in Subsection 3.2. The length of the expiration period inversely correlates with the model's ability to adapt to sudden changes. The adaptation to shifts in the data-generating process, such as changes in mean or variance, is managed through the adaptation period t_a. A longer t_a results in slower adaptation but potentially longer alerts, which can be valuable during extended outlier periods. In most cases, $t_\mathrm{a} = 1/4t_\mathrm{e}$ offers optimal performance.

As a general rule of thumb, expiration period t_e should be determined based on the slowest observed dynamics within the multivariate system. The

threshold T defaults to the three-sigma probability of q in (19). Adjusting this threshold can fine-tune the trade-off between precision and recall. A higher threshold boosts recall but may lower precision, while a lower threshold enhances precision at the cost of recall. The presence of one non-default easily interpretable hyperparameter facilitates adaptability to various scenarios. We recommend starting with the default values of other parameters and making adjustments based on real-time model performance.

3.2. Online training

Training in RAID follows an incremental learning approach, processing each new sample upon arrival. Incremental learning allows online parameter updates, albeit with a potential computational delay affecting response latency.

In the case of a dynamic joint probability distribution, the parameters are $\boldsymbol{\mu_i}$ and $\boldsymbol{\Sigma_i}$ at time instance i. Update of the mean vector $\boldsymbol{\mu_i}$ and covariance matrix $\boldsymbol{\Sigma_i}$ is governed by Welford's online algorithm using equation (2) and (4) respectively. Samples beyond the expiration period t_{e} are disregarded during the second pass. The effect of expired samples is reverted using inverse Welford's algorithm for mean (6) and variance (7), accessing the data in the buffer. For details, refer to Subsection 2.2.

It's worth noting that adaptation relies on two self-supervised methods. Adaptation routine runs if the observation at time instance i is considered normal. Adaptation period t_{a} allows the model to update the distribution on outliers as well. Given the predicted system anomaly state from (24) as y_i over the window of past observations $\boldsymbol{y}_i = \{y_{i-t_{\mathrm{a}}}, ..., y_i\}$, the following test holds when adaptation is performed on outlier:

$$\frac{\sum_{y \in \boldsymbol{y}_i} y}{n(\boldsymbol{y}_i)} > 2 * (T - 0.5). \tag{25}$$

Here $n(\boldsymbol{y}_i)$ denotes the dimensionality of $\boldsymbol{y}_i$. The logic of the (25) follows the probabilistic approach to anomalies that assumes a number of anomalies are lower or equal to the conditional probability at both tails of the distribution

3.3. Online prediction

In the prediction phase, multiple metrics are evaluated to assess the state of the modeled system.

Firstly, we calculate the parameters of the conditional distribution concerning the dynamic multivariate Gaussian distribution. These calculations

are performed for the process observation vector $\boldsymbol{x}_i$ at time instance i. Specifically, we compute the conditional mean using (18) and the conditional variance using (17). These computations yield univariate conditional distributions for individual signals and features. These conditional distributions play a crucial role in assessing the abnormality of signals and features concerning other observed values. This assessment relies on the strength of relationships defined by the covariance matrix of the dynamic multivariate Gaussian distribution. Consequently, our approach inherently considers the interactions between input signals and features. The determination of anomalous behavior is governed by (23).

Any anomaly detected within one of the features triggers an alert at the system level. The decision regarding the overall system's anomalous behavior is guided by (24). Nevertheless, individual determinations of anomalies serve as a diagnostic tool for isolating the root cause of anomalies.

To assist operators in their assessments, we establish a hypercube defined by lower and upper threshold values, denoted as $\boldsymbol{x}_{\mathrm{l}}$ and $\boldsymbol{x}_{\mathrm{u}}$, respectively. These thresholds are derived from (20) and (21), incorporating updated model parameters. Lower and upper thresholds play a pivotal role as dynamic process limits. They replace the conservative process limits provided in sensor documentation, accounting for factors such as aging and actual environmental conditions that influence sensor operation.

Our framework anticipates unexpected novel behavior, including signal loss. This anticipation involves calculating the cumulative distribution function (CDF) over the univariate normal distribution of sampling, focusing on the differences between subsequent timestamps. We operate under the assumption that, over the long term, the distribution of sampling times remains stable. As a result, we employ a one-pass update mechanism utilizing (2) and (4). To proactively detect subtle changes in sampling patterns, self-supervised learning is employed, leveraging anomalies weighted by the deviation from $(1 - F(x_i; \mu, \sigma^2))$ for training.

The system is vigilant in identifying change points. When the adaptation test specified in (25) is satisfied, change points are flagged and isolated. This initiation of change points triggers updates to the model, ensuring it adapts to evolving data patterns effectively.

Algorithm 2 Online Detection and Identification Workflow

Input: expiration period t_{e}

Output: system anomaly y_i, signal anomalies $\boldsymbol{y}_{\mathrm{s},i}$, sampling anomaly $y_{\mathrm{t},i}$, change-point $y_{\mathrm{c},i}$, lower thresholds $\boldsymbol{x}_{\mathrm{l},i}$, upper thresholds $\boldsymbol{x}_{\mathrm{u},i}$,

Initialisation :

$i \leftarrow 1;\ n \leftarrow 1;\ T \leftarrow (19);\ \boldsymbol{\mu} \leftarrow \boldsymbol{x_0};\ \boldsymbol{\Sigma} \leftarrow \mathbf{1}_{k\times k};\ \mu_{\mathrm{t}} \leftarrow 0;\ \sigma_{\mathrm{t}}^2 \leftarrow 1;$

compute $F(\boldsymbol{x_0}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ using algorithm in Genz (2000);

LOOP Process

loop

  $\boldsymbol{x}_i, t_i \leftarrow$ RECEIVE();

  $\boldsymbol{y}_{\mathrm{s},i} \leftarrow$ PREDICT$(\boldsymbol{x}_i, T)$ using (23);

  $y_i \leftarrow$ PREDICT$(\boldsymbol{y}_{\mathrm{s},i})$ using (24);

  $\boldsymbol{x}_{\mathrm{l},i},\ \boldsymbol{x}_{\mathrm{u},i} \leftarrow$ GET$(T, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ using Algorithm 1;

  $y_{\mathrm{t},i} \leftarrow$ PREDICT$(t_i - t_{i-1})$ using (23);

  $\mu_{\mathrm{t}}, \sigma_{\mathrm{t}}^2 \leftarrow$ UPDATE$(t_i - t_{i-1}, \mu_{\mathrm{t}}, \sigma_{\mathrm{t}}^2)$ using (2), (4);

  if not (24) **or** (25) **then**

    $\boldsymbol{\mu}, \boldsymbol{\Sigma} \leftarrow$ UPDATE$(\boldsymbol{x}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}, n)$ using (2), (4);

    if (25) **then**

      $y_{\mathrm{c},i} \leftarrow 1;$

    else

      $y_{\mathrm{c},i} \leftarrow 0;$

    end if

    $n \leftarrow n + 1;$

    for $\boldsymbol{x}_{i-t_{\mathrm{e}}}$ **do**

      $\boldsymbol{\mu}, \boldsymbol{\Sigma} \leftarrow$ REVERT$(\boldsymbol{x}_{i-t_{\mathrm{e}}}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, n)$ using (6), (7);

      $n \leftarrow n - 1;$

    end for

  end if

  $i \leftarrow i + 1;$

end loop

4. Case Study

This section provides a benchmark and two case studies that showcase the effectiveness and applicability of our proposed approach. In the following Subsections, we investigate the properties and performance of the approach using streamed benchmark system data and signals from IoT devices in a mi-

crogrid system. The successful deployment demonstrates that this approach is suitable for existing process automation infrastructure.

The case studies were realized using Python 3.10.1 on a machine employing an 8-core Apple M1 CPU and 8 GB RAM.

4.1. Benchmark

In this subsection, we compare the proposed method with adaptive unsupervised detection methods without an interpretability layer. Two of the well-established methods, providing iterative learning capabilities over multivariate time-series data are One-Class Support Vector Machine (OC-SVM) and Half Spaced Trees (HS-Trees). Both methods represent the backbone of multiple state-of-the-art methods for cases of anomaly detection on dynamic system data, with a brief list of recent applications in Introduction 1.3. Comparison is conducted on real benchmarking data, annotated with labels of whether the observation was anomalous or normal. The dataset of Skoltech Anomaly Benchmark (SKAB) Katser and Kozitsin (2020) is used for this purpose, as no established benchmarking multivariate data were found regarding energy storage systems. It represents a combination of experiments with the behavior of rotor imbalance as a subject to various functions introduced to control action as well as slow and sudden changes in the amount of water in the circuit. The system is described by 8 features. The data were preprocessed according to best practices for the given method, namely: standard scaling for OC-SVM, normalization for HS-Trees, and no scaling for our proposed method. The optimal quantile threshold value for both reference methods is found using Bayesian Optimization. Due to no further knowledge about the process, the parameters of the proposed method were optimized using Bayesian Optimization as well. Results are provided within Table 1, evaluating F1 score, Recall and Precision. A value of 100% at each metric represents a perfect detection. The latency represents the average computation time per sample of the pipeline including training and data preprocessing.

The results in Table 1 suggest, that our algorithm provides slightly better performance than reference methods. Based on the Scoreboard for various algorithms on SKAB's Kaggle page, our iterative approach performs comparably to the evaluated batch-trained model. Such a model has all the training data available before prediction unlike ours, evaluating the metrics iteratively on a streamed dataset.

Table 1: Metrics evaluation on SKAB dataset

Metric	AID	OC-SVM	HS-Trees
F1 [%]	**48.70**	44.42	34.10
Recall [%]	49.90	**56.67**	32.57
Precision [%]	**47.56**	36.52	35.77
Avg. Latency [ms]	1.55	0.44	**0.21**

4.2. Battery Energy Storage System (BESS)

In the first case study, we verify our proposed method on BESS. The BESS reports measurements of State of Charge (SoC), supply/draw energy set-points, and inner temperature, at the top, middle, and bottom of the battery module. Tight battery cell temperature control is needed to optimize performance and maximize the battery's lifespan. Identifying anomalous events and removal of corrupted data might yield significant improvement in the process control level and increase the reliability and stability of the system.

The default sampling rate of the signal measurement is 1 minute. However, network communication of the IoT devices is prone to packet dropout, which results in unexpected non-uniformities in sampling. The data are normalized to the range $[0, 1]$ to protect the sensitive business value. The proposed approach is deployed to the existing infrastructure of the system, allowing real-time detection and diagnosis of the system.

The industrial partner provided a physical model of the battery cell temperature, defined as follows:

$$\begin{aligned} T_{\mathrm{bat},i+1} = T_{\mathrm{bat},i} + T_{\mathrm{s}}(&q_{\mathrm{fan}} V_{\mathrm{b,max}} \rho c_{\mathrm{p}} (T_{\mathrm{out}} - T_{\mathrm{bat},i}) + V_{\mathrm{c,max}} q_{\mathrm{circ.fan}} \rho c_{\mathrm{p}} T_{\mathrm{bat},i} \\ &+ q_{\mathrm{circ.fan}} (P_{\mathrm{cool}} q_{\mathrm{cool}} P_{\mathrm{heat}} q_{\mathrm{heat}}) + c_{\mathrm{scale}} Q_{\mathrm{bat}} + q_{\mathrm{inner\ fans}} \\ &- (V_{\mathrm{b,max}} q_{\mathrm{fan}} V_{\mathrm{c,max}} q_{\mathrm{circ.fan}}) \rho c_{\mathrm{p}} T_{\mathrm{bat},i}) / (m_{\mathrm{bat}} c_{\mathrm{p,b}}) \end{aligned} \tag{26}$$

When combined with an averaged measurement of battery cell temperature, we could compute the difference between real and predicted temperature. Such deviation can be useful in detecting unexpected patterns in temperature. Nevertheless, it may be inaccurate as the physical model is simplified and does not account for spatial aspects, like temperature gradients as well as different dynamic effects of charging and discharging on

temperature. Therefore, the raw measured temperature is used as well. The deviation between demanded power and delivered power was used to aid the identification of the state, as the increased difference might be related to other unexpected and novel patterns.

Fig. 1 depicts the operation of the BESS over March 2022. Multiple events of anomalous behavior happened within this period, confirmed by the operators, that are observable through a sudden or significant shift in measurements in a given period. As the first step, the detection mechanism was initialized, following the provided guidelines for parameter selection in Subsection 3.1. The expiration period was set to $t_e = 7$ days, due to the weekly seasonality of human behavior impacting battery usage. The threshold was kept at default value $T = 0.99735$. A grace period, during which the model learns from both normal and anomalous data (though normal are expected), is shortened to 2.5 days to observe the effect of BESS calibration happening on 3^{rd} day from deployment.

The deployment and operation of the anomaly detection system were successful as shown by its adaptation of changepoint on 7^{th} March 2022 that appeared due to the relocation of the battery storage system outdoors. The model was adapted online based on Subsection 3.2. The sudden shift in environmental conditions, due to the transfer of the system to outside changed the dynamics of the system's temperature. However, new behavior was adopted by the top-level anomaly isolation system within five days, reducing potential false alerts afterward, by observably shifting the conditional mean to lower temperatures. Perhaps more interesting are the alerted changepoints.

Calibration of the BESS, usually observed as deviations of setpoint from real power demand and multiple peaks in temperature was captured as well.

The system identified 6 deviations in sampling, denoted by the red bars in Fig. 1. 4 anomalies with shorter duration represented packet loss. The prolonged anomaly was notified during the transfer of the battery pack. The longest dropout observed happened across 20^{th} March up to 21^{st}. Unexpectedly, the change point detection module triggered an alarm at the end of the loss, resulting in adaptation and a sharp shift in drawn limits for Power Setpoint Deviation. Red dots represent anomalies at the signal level given by equation (23). The dynamic signal limits are surpassed in one or multiple signals during the system's anomalies. The root cause isolation allows the pairing of anomalies with specific features. Conditional probability, against which the anomalies are evaluated allows consideration of signal relationships within individual limits.

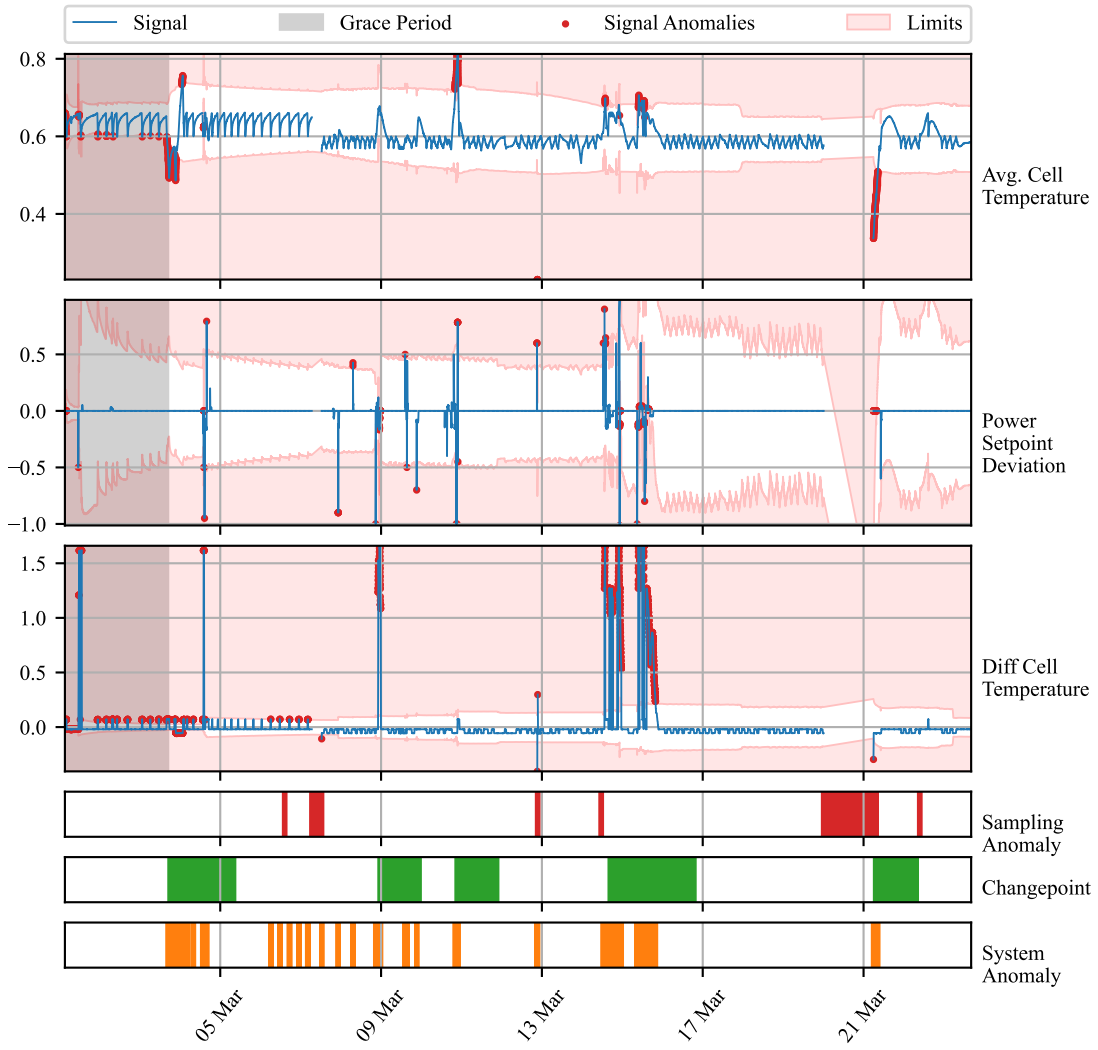

Figure 1: Time Series of BESS measurements (blue line) of process variables. The y-axis renders the values after the normalization of raw inputs. Root causes of anomalies are marked within specific signals as red dots. The light red area represents out-of-limits values for individual signals. Non-uniform sampling is marked as red bars. Green bars represent the times, at which changepoint was detected. All the signal anomalies are depicted as orange bars below the graph.

4.3. Kokam Battery Temperature Module(BESS)

A second case study is concerned with monitoring temperature profiles of individual modules of battery pack deployed at end user. During the operation, a hardware fault of the cooling fan happened. Our industrial partner was interested in finding out, whether such an event could be captured by an anomaly detection system. The data for 12 modules, each coming with 6 channels of measurement were retrieved in 30-second sampling and processed in a streamed manner. We found it informative to compute the deviation of the observed value from the average of all the above-mentioned measurements.

Our anomaly detection system was, once again, initialized with an expiration period of 7 days. The grace period was shortened to 1 day. The threshold value was shifted to a 4 sigma value of 99.977% to minimize the number of alarms.

In Figure 2 we observe 5 days of deviations between the observed temperature measured by channels of module 9 and the average temperature of all modules. After the grace period, we observe multiple system alarms raised by various channels. Until the noon of 22$^{\text{nd}}$ August, they seem to be spread out randomly between individual channels. During the late evening of 22$^{\text{nd}}$, anomalies were reported by both channels 4 and 5 for a prolonged period, followed by an anomalous rise in temperature measured by channel 6 early in the morning on 23$^{\text{rd}}$ August. The fan fault was observed approximately at 5 pm on 23$^{\text{rd}}$ August. Our anomaly detection system instantly raised an alarm, notifying us of anomalous behavior reported by channels 1 - 3. The prolonged duration of the alarm triggered the changepoint alarm approximately 2 hours later. This resulted in a slightly faster adaptation of the system to the new operation under increased temperature. Surprisingly, the temperature decreased during the next day, notifying us of the fan being in operation, to fail again 30 minutes later after the battery modules were cooled down to the previous setpoint. The anomaly detection system was triggered once again, although adaptation loosened the region of normal operation to allow itself to adapt. No significant anomalies in sampling were observed during the period.

5. Conclusion

In this paper, we examine the capacity of adaptive conditional probability distribution to model the normal operation of dynamic systems employing

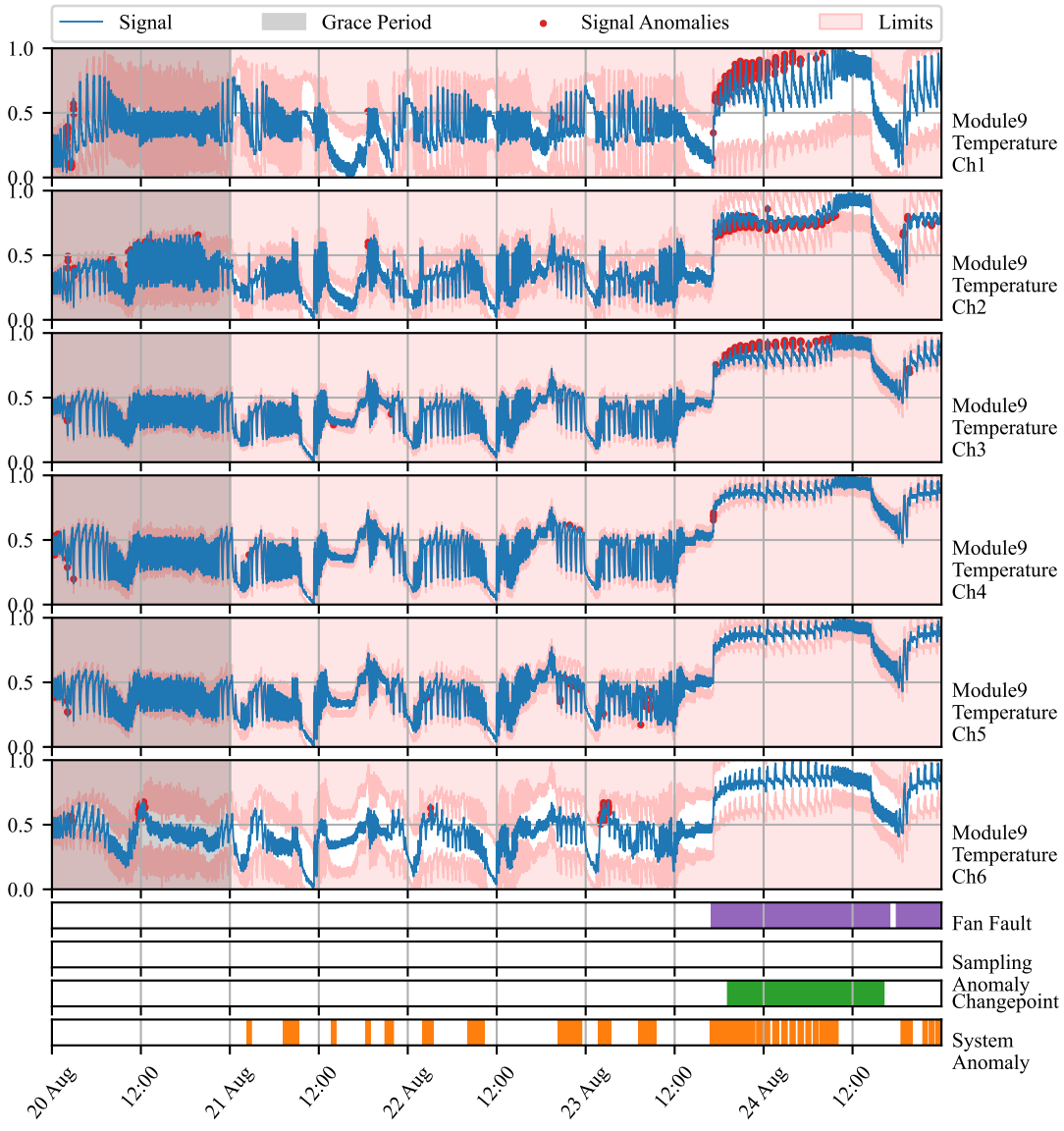

Figure 2: Time Series of BESS measurements (green line) of process variables. Non-uniform ticks on the x-axis mark days of interest (NOTE: some marks are hidden due to the readability). The y-axis renders the normalized process variables. System anomalies are marked as red dots. Non-uniform sampling is detected at blue vertical lines. Yellow vertical lines denote changepoint adaptation

streaming IoT devices and isolate the root cause. The dynamics of the systems are elaborated in the model using Welford's online algorithm with the capacity to update and revert sufficient parameters of underlying multivariate Gaussian distribution in time making it possible to elaborate non-stationarity in the process variables. Moreover, the self-supervision allows protection of the distribution from the effect of outliers and increased speed of adaptation in cases of changes in operation.

We assume the Gaussian distribution of measurements over a bounded time frame related to the system dynamics. We consider such an assumption reasonable, with support of multiple trials where the Kolmogorov-Smirnov test did not reject this hypothesis. The statistical model provides the capacity for the interpretation of the anomalies as extremely deviating observations from the mean vector. Another assumption held in this study is that any anomaly, spatial or temporal, can be transformed in such a way that makes it an outlier given that we expose such effects as features as shown in case studies.

Our approach establishes the system's operation state at the global anomaly level by considering interactions between input measurements and engineered features and computing distance from mean of conditional probability. At the second level, dynamic process limits based on PPF at threshold probability, given multivariate distribution parameters, help isolate the root cause of anomalies. This level serves the diagnostic purpose of the model operation. The individual signals contribute to the global anomaly prediction, while the proposed dynamic limits offer less conservative restrictions on individual process operation. In parallel, the detector allows discrimination of signal losses due to packet drops and sensor malfunctioning.

The ability to detect and identify anomalies in the system, isolate the root cause of anomaly to specific signal or feature, and identify signal losses is shown in two case studies on real data. Unlike many anomaly detection approaches, the proposed AID method does not require historical data or ground truth information about anomalies, relieving general limitations. Moreover, it combines adaptability and interpretability, which is an area yet to be explored.

The benchmark performed on industrial data showed the ability to provide comparable results to other self-learning adaptable anomaly detection methods. This is an important property for our model which allows, in addition, the root cause isolation. The first case study, performed on real operation data of BESS, examined the battery energy storage system and

demonstrated the ability to capture system anomalies and provide less conservative limits to signals. The physical model aided decisions about the normality of the measured temperature of BESS.

The second case study exposed the ability to detect anomalies in the temperature profiles of battery modules within the battery pack, considering spatial measurements made by multiple sensors distributed around the module and the average temperature of all the modules within the pack. Hardware fault observed on this deployed device was captured by our model, giving another proof of its importance in energy storage systems monitoring, where tight temperature control plays a significant role in the safety and profitability of the system.

Future works on the method will include improvement to the change point detection mechanism, decrease in the latency on high dimensional data, and false positive rate reduction, from which general plug-and-play models suffer.

References

V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (2009). URL: https://doi.org/10.1145/1541880.1541882. doi:10.1145/1541880.1541882.

N. Laptev, S. Amizadeh, I. Flint, Generic and scalable framework for automated time-series anomaly detection, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 1939–1947. URL: https://doi.org/10.1145/2783258.2788611. doi:10.1145/2783258.2788611.

A. Kejariwal, Introducing practical and robust anomaly detection in a time series, 2015. URL: https://blog.twitter.com/engineering/en_us/a/2015/introducing-practical-and-robust-anomaly-detection-in-a-time-series.

A. A. Cook, G. Mısırlı, Z. Fan, Anomaly detection for iot time-series data: A survey, IEEE Internet of Things Journal 7 (2020) 6481–6494. doi:10.1109/JIOT.2019.2958185.

K. Zhang, J. Chen, C.-G. Lee, S. He, An unsupervised spatiotemporal fusion network augmented with random mask and time-relative information modulation for anomaly detection of machines

with multiple measuring points, Expert Systems with Applications 237 (2024) 121506. URL: https://www.sciencedirect.com/science/article/pii/S0957417423020080. doi:https://doi.org/10.1016/j.eswa.2023.121506.

J. Huang, D. Cheng, S. Zhang, A novel outlier detecting algorithm based on the outlier turning points, Expert Systems with Applications 231 (2023) 120799. URL: https://www.sciencedirect.com/science/article/pii/S0957417423013015. doi:https://doi.org/10.1016/j.eswa.2023.120799.

C. Fan, Y. Sun, Y. Zhao, M. Song, J. Wang, Deep learning-based feature engineering methods for improved building energy prediction, Applied Energy 240 (2019) 35–45. URL: https://www.sciencedirect.com/science/article/pii/S0306261919303496. doi:https://doi.org/10.1016/j.apenergy.2019.02.052.

P. D. Talagala, R. J. Hyndman, K. Smith-Miles, Anomaly detection in high-dimensional data, Journal of Computational and Graphical Statistics 30 (2021) 360–374. URL: https://doi.org/10.1080/10618600.2020.1807997. doi:10.1080/10618600.2020.1807997. arXiv:https://doi.org/10.1080/10618600.2020.1807997.

J. Li, Z. Liu, Attribute-weighted outlier detection for mixed data based on parallel mutual information, Expert Systems with Applications 236 (2024) 121304. URL: https://www.sciencedirect.com/science/article/pii/S0957417423018067. doi:https://doi.org/10.1016/j.eswa.2023.121304.

X. Du, J. Chen, J. Yu, S. Li, Q. Tan, Generative adversarial nets for unsupervised outlier detection, Expert Systems with Applications 236 (2024) 121161. URL: https://www.sciencedirect.com/science/article/pii/S0957417423016639. doi:https://doi.org/10.1016/j.eswa.2023.121161.

M. Salehi, L. Rashidi, A survey on anomaly detection in evolving data: [with application to forest fire risk prediction], SIGKDD Explor. Newsl. 20 (2018) 13–23. URL: https://doi.org/10.1145/3229329.3229332. doi:10.1145/3229329.3229332.

N. Barbosa Roa, L. Travé-Massuyès, V. H. Grisales-Palacio, Dyclee: Dynamic clustering for tracking evolving environments, Pattern Recognition 94 (2019) 162–186. URL: https://www.sciencedirect.com/science/article/pii/S0031320319301992. doi:https://doi.org/10.1016/j.patcog.2019.05.024.

A. G. Tartakovsky, A. S. Polunchenko, G. Sokolov, Efficient computer network anomaly detection by changepoint detection methods, IEEE Journal of Selected Topics in Signal Processing 7 (2013) 4–11. doi:10.1109/JSTSP.2012.2233713.

H. Wu, J. He, M. Tömösközi, Z. Xiang, F. H. Fitzek, In-network processing for low-latency industrial anomaly detection in softwarized networks, in: 2021 IEEE Global Communications Conference (GLOBECOM), 2021, pp. 01–07. doi:10.1109/GLOBECOM46510.2021.9685489.

H. S. Pannu, J. Liu, S. Fu, Aad: Adaptive anomaly detection system for cloud computing infrastructures, in: 2012 IEEE 31st Symposium on Reliable Distributed Systems, 2012, pp. 396–397. doi:10.1109/SRDS.2012.3.

S. Ahmad, A. Lavin, S. Purdy, Z. Agha, Unsupervised real-time anomaly detection for streaming data, Neurocomputing 262 (2017) 134–147. URL: https://www.sciencedirect.com/science/article/pii/S0925231217309864. doi:https://doi.org/10.1016/j.neucom.2017.04.070, online Real-Time Learning Strategies for Data Streams.

H. H. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, A. Liotta, Ensembles of incremental learners to detect anomalies in ad hoc sensor networks, Ad Hoc Networks 35 (2015) 14–36. URL: https://www.sciencedirect.com/science/article/pii/S1570870515001481. doi:https://doi.org/10.1016/j.adhoc.2015.07.013, special Issue on Big Data Inspired Data Sensing, Processing and Networking Technologies.

M. Carletti, C. Masiero, A. Beghi, G. A. Susto, Explainable machine learning in industry 4.0: Evaluating feature importance in anomaly detection to enable root cause analysis, in: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), 2019, pp. 21–26. doi:10.1109/SMC.2019.8913901.

Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, M. C. Chan, Gee: A gradient-based explainable variational autoencoder for network anomaly detection, in: 2019 IEEE Conference on Communications and Network Security (CNS), 2019, pp. 91–99. doi:10.1109/CNS.2019.8802833.

K. Amarasinghe, K. Kenney, M. Manic, Toward explainable deep neural network based anomaly detection, in: 2018 11th International Conference on Human System Interaction (HSI), 2018, pp. 311–317. doi:10.1109/HSI.2018.8430788.

X. Zhang, J. Shi, X. Huang, F. Xiao, M. Yang, J. Huang, X. Yin, A. Sohail Usmani, G. Chen, Towards deep probabilistic graph neural network for natural gas leak detection and localization without labeled anomaly data, Expert Systems with Applications 231 (2023) 120542. URL: https://www.sciencedirect.com/science/article/pii/S0957417423010448. doi:https://doi.org/10.1016/j.eswa.2023.120542.

W.-T. Yang, M. S. Reis, V. Borodin, M. Juge, A. Roussy, An interpretable unsupervised bayesian network model for fault detection and diagnosis, Control Engineering Practice 127 (2022) 105304. URL: https://www.sciencedirect.com/science/article/pii/S0967066122001502. doi:https://doi.org/10.1016/j.conengprac.2022.105304.

L. C. Brito, G. A. Susto, J. N. Brito, M. A. V. Duarte, Fault diagnosis using explainable ai: A transfer learning-based approach for rotating machinery exploiting augmented synthetic data, Expert Systems with Applications 232 (2023) 120860. URL: https://www.sciencedirect.com/science/article/pii/S0957417423013623. doi:https://doi.org/10.1016/j.eswa.2023.120860.

Z. Wu, X. Yang, X. Wei, P. Yuan, Y. Zhang, J. Bai, A self-supervised anomaly detection algorithm with interpretability, Expert Systems with Applications 237 (2024) 121539. URL: https://www.sciencedirect.com/science/article/pii/S0957417423020419. doi:https://doi.org/10.1016/j.eswa.2023.121539.

M. Wadinger, M. Kvasnica, Real-time outlier detection with dynamic process limits, in: Proceedings of the 2023 24th International Conference on Process Control (PC), 2023. In press.

K. Yamanishi, J.-i. Takeuchi, A unifying framework for detecting outliers and change points from non-stationary time series data, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '02, Association for Computing Machinery, New York, NY, USA, 2002, pp. 676–681. URL: https://doi.org/10.1145/775047.775148. doi:10.1145/775047.775148.

K. Yamanishi, J.-i. Takeuchi, G. Williams, P. Milne, On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms, Data Mining and Knowledge Discovery 8 (2004) 275–300. URL: https://doi.org/10.1023/B:DAMI.0000023676.72185.7c. doi:10.1023/B:DAMI.0000023676.72185.7c.

B. Steenwinckel, Adaptive anomaly detection and root cause analysis by fusing semantics and machine learning, in: A. Gangemi, A. L. Gentile, A. G. Nuzzolese, S. Rudolph, M. Maleshkova, H. Paulheim, J. Z. Pan, M. Alam (Eds.), The Semantic Web: ESWC 2018 Satellite Events, Springer International Publishing, Cham, 2018, pp. 272–282.

B. Steenwinckel, D. De Paepe, S. Vanden Hautte, P. Heyvaert, M. Bentefrit, P. Moens, A. Dimou, B. Van Den Bossche, F. De Turck, S. Van Hoecke, F. Ongenae, Flags: A methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert knowledge with machine learning, Future Generation Computer Systems 116 (2021) 30–48. URL: https://www.sciencedirect.com/science/article/pii/S0167739X20329927. doi:https://doi.org/10.1016/j.future.2020.10.015.

M. Amer, M. Goldstein, S. Abdennadher, Enhancing one-class support vector machines for unsupervised anomaly detection, in: Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description, ODD '13, Association for Computing Machinery, New York, NY, USA, 2013, pp. 8–15. URL: https://doi.org/10.1145/2500853.2500857. doi:10.1145/2500853.2500857.

B. Liu, Y. Xiao, P. S. Yu, L. Cao, Y. Zhang, Z. Hao, Uncertain one-class learning and concept summarization learning on uncertain data streams, IEEE Transactions on Knowledge and Data Engineering 26 (2014) 468–484. doi:10.1109/TKDE.2012.235.

B. Krawczyk, M. Woźniak, One-class classifiers with incremental learning and forgetting for data streams with concept drift, Soft Computing 19 (2015) 3387–3400. URL: https://doi.org/10.1007/s00500-014-1492-5. doi:10.1007/s00500-014-1492-5.

X. Miao, Y. Liu, H. Zhao, C. Li, Distributed online one-class support vector machine for anomaly detection over networks, IEEE Transactions on Cybernetics 49 (2019) 1475–1488. doi:10.1109/TCYB.2018.2804940.

Ö. Gözüaçık, F. Can, Concept learning using one-class classifiers for implicit drift detection in evolving data streams, Artificial Intelligence Review 54 (2021) 3725–3747. URL: https://doi.org/10.1007/s10462-020-09939-x. doi:10.1007/s10462-020-09939-x.

R. Wetzig, A. Gulenko, F. Schmidt, Unsupervised anomaly alerting for iot-gateway monitoring using adaptive thresholds and half-space trees, in: 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), 2019, pp. 161–168. doi:10.1109/IOTSMS48152.2019.8939201.

Y. Lyu, W. Li, Y. Wang, S. Sun, C. Wang, Rmhsforest: Relative mass and half-space tree based forest for anomaly detection, Chinese Journal of Electronics 29 (2020) 1093–1101. URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cje.2020.09.010. doi:https://doi.org/10.1049/cje.2020.09.010. arXiv:https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/cje.2020.09.01

B. P. Welford, Note on a method for calculating corrected sums of squares and products, Technometrics 4 (1962) 419–420. URL: https://www.tandfonline.com/doi/abs/10.1080/00401706.1962.10490022. doi:10.1080/00401706.1962.10490022. arXiv:https://www.tandfonline.com/doi/pdf/10.1080/00401706.1962.10490022.

A. Genz, Numerical computation of multivariate normal probabilities, Journal of Computational and Graphical Statistics 1 (2000). doi:10.1080/10618600.1992.10477010.

R. P. Brent, Algorithms for minimization without derivatives, Prentice-Hall, Englewood Cliffs, N.J, 1972. URL: https://openlibrary.org/books/OL4739237M/Algorithms_for_minimization_without_derivatives.

F. Iglesias Vázquez, A. Hartl, T. Zseby, A. Zimek, Anomaly detection in streaming data: A comparison and evaluation study, Expert Systems with Applications 233 (2023) 120994. URL: https://www.sciencedirect.com/science/article/pii/S0957417423014963. doi:https://doi.org/10.1016/j.eswa.2023.120994.

L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, K.-R. Müller, A unifying review of deep and shallow anomaly detection, Proceedings of the IEEE 109 (2021) 756–795. doi:10.1109/JPROC.2021.3052449.

I. D. Katser, V. O. Kozitsin, Skoltech anomaly benchmark (skab), https://www.kaggle.com/dsv/1693952, 2020. doi:10.34740/KAGGLE/DSV/1693952.