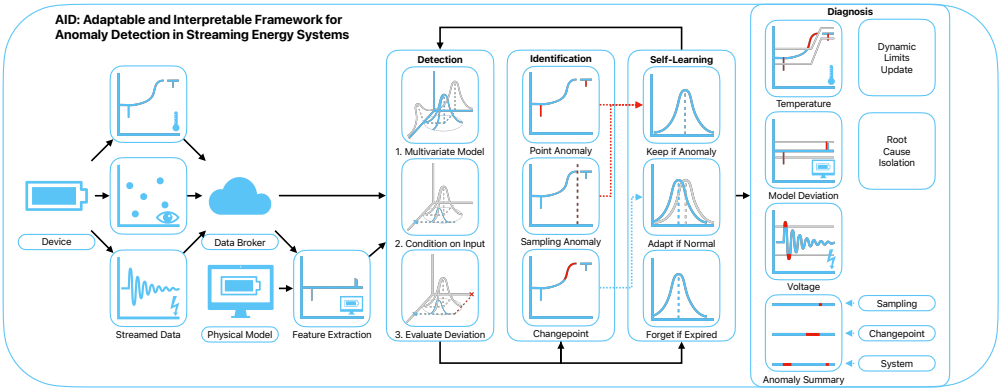
Graphical Abstract

Adaptable and Interpretable Framework for Anomaly Detection in SCADA-based IoT Systems

Marek Wadinger, Michal Kvasnica



Highlights

Adaptable and Interpretable Framework for Anomaly Detection in SCADA-based IoT Systems

Marek Wadinger, Michal Kvasnica

- First interpretable anomaly detector with self-supervised adaptation
- Delivers comparable performance to established general methods
- Isolates root cause of anomalies while considering interactions
- Demonstrates interpretability by providing process limits for signals
- Uses self-learning approach on streamed IoT data

Adaptable and Interpretable Framework for Anomaly Detection in SCADA-based IoT Systems

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

[a] *Institute of Information Engineering, Automation and Mathematics, Slovak University of Technology in Bratislava,*
[b] *Tesla 50Hz s.r.o.,*

Abstract

In this paper, we introduce an Adaptable and Interpretable Framework for Anomaly Detection (AID) designed for industrial systems utilizing IoT data streams on top of well-established SCADA systems. AID leverages dynamic conditional probability distribution modeling to capture the normal operation of dynamic systems and isolate the root causes of anomalies. The self-supervised framework dynamically updates parameters of underlying model, 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. Crucially, AID provides dynamic process limits to integrate with existing alarm handling mechanisms in SCADA and pinpoint root causes at the level of individual inputs. Two industrial-scale case studies showcase AID's capabilities. The first study showcases AID's effectiveness on energy storage system, adapting to changes, setting less conservative process limits for SCADA, and ability to leverage a physical model. The second study monitors battery module temperatures, where AID identifies hardware faults, emphasizing its relevance to energy storage safety. A benchmark evaluation on real data shows that AID delivers comparable performance to other self-learning adaptable anomaly detection methods, with the significant advancement in diagnostic capabilities for improved system reliability and performance.

Keywords: Anomaly detection, Root cause isolation, Iterative learning, Statistical learning, Self-supervised learning

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 complexity issues, Zhang et al. have successfully employed spatially distributed sensors and time-relative modulation. Their approach has proven effective, particularly in the context of complex non-linear 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 is worth noting that feature engineering may introduce categorical variables and significantly increase the 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, immediate detection is not a feasible option unless there is a high tolerance for false alarms.

The adaptation of batch models at scale 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, the 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 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 competitive 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 in real industrial systems. Here we report the discovery and characterization of an adaptive anomaly detection method for existing SCADA systems, employing streaming IoT data. The ability to diagnose multivariate data while providing root cause isolation via statistical learning, extends our previous contribution to the field as presented in (Wadinger and Kvasnica (2023)). The proposed algorithm represents a general method that aids a range of existing 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 highlight the impactful contributions of Yamanishi et al. in (Yamanishi and Takeuchi (2002); 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 facilitates intelligent adaptation concerning detected change points, to increase the speed of adaptation where the probability of concept drift is high. By leveraging its ability to adapt to changes in operational states, our proposed method operates autonomously when such changes occur. Moreover, Yamanishi et al. (2004) does not attempt to isolate the root cause of the anomaly. Our approach extends statistical outlier detection by incorporating interpretability. This is achieved by evaluating the inverse cumulative distribution function of the latest conditional probabilities for each measurement, considering the remainder of the measurements, and establishing limits that define the threshold for normal event probabilities.

A limited number of studies have focused on adaptation and interpretability within the framework of anomaly detection. Two recent contributions in this area are made by Steenwinckel et al. as reported in (Steenwinckel (2018); 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 is 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 efforts are directed towards the development of a self-supervised method that can learn autonomously, reducing the reliance on human supervision, which is often constrained by time limitations and can lead to significant delays in adaptation. Our method is distinguished by its

straightforward statistical reasoning and the ability to isolate the root cause of anomalies. The interpretability of our method is demonstrated through the establishment of dynamic process limits for each signal, leveraging conditional probabilities derived from the signal and other system measurements and features. This provides operators with a clear understanding of the system's state and the underlying causes of anomalies and aids existing alarm handling mechanisms in SCADA which utilize process limits. To the best of our knowledge, this study appears to be one of the initial attempts to introduce a self-supervised approach for adaptive anomaly detection and root cause isolation in SCADA-based systems utilizing IoT data streams.

1.3. 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 causes while adapting to changes in operation and environment, is crucial for the system safety. The proposed method is designed to be integrated into the existing infrastructure of the systems, utilizing IoT data streams on top of well-established SCADA systems. SCADA systems continuously monitor these process data in real-time, embodying alarm handling mechanisms, which are designed to notify operators of the system's abnormal behavior and drive attention to the root of the problem. By comparing the current values to the upper and lower process limits, they take action when a variable exceeds or falls below these limits. However, safe operating limits are often established based on a combination of equipment design limits and the dynamics of the process (Stauffer and Chastain-Knight (2021)). Those are indifferent to the actual state of the system and environmental conditions. The proposed method allows the establishment of dynamic process limits, which are based on the current state of the system and its environment. This allows the system to operate closer or further from its design limits, increasing its safety and profitability. 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.

1.4. Validation

Two case studies show that our proposed method has the capacity to explain anomalies, isolate the root cause, 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, on benchmark data 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)).

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

- Enriches interpretable anomaly detection with adaptive capabilities
- Isolates root cause of anomalies while considering interactions
- Uses self-learning approach on streamed IoT data
- Demonstrates interpretability by providing process limits for signals
- Blends with existing SCADA architecture

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 industrial-scale 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 through data. Therefore, 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, significant loss of precision due to subtracting two nearly equal numbers. 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. Inverting 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

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

where the reverted mean is given as

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

Finally, the unbiased variance follows the formula:

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

Notably, inversion allows the algorithm to keep a constant rate of adaptation at the cost of storing a bounded data buffer.

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 (Mishra and Datta-Gupta (2018)). 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 vector $\boldsymbol{X}$ take on a value less than or equal to a particular point q in space, and can be used to evaluate the likelihood of observing a particular set of measurements or data points. 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:

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

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 is part of standard statistical software tools.

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

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{aa}^2 & \boldsymbol{\Sigma_{ab}} \\ \boldsymbol{\Sigma_{ba}} & \boldsymbol{\Sigma_{bb}} \end{bmatrix} \text{ with dimensions } \begin{bmatrix} 1 \times 1 & 1 \times (k-1) \\ (k-1) \times 1 & (k-1) \times (k-1) \end{bmatrix}. \tag{13}$$

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:

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

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 by applying the Schur complement as follows:

$$\sigma^2_{a|\boldsymbol{b}} = \sigma^2_{aa} - \boldsymbol{\Sigma}_{a\boldsymbol{b}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{bb}}\boldsymbol{\Sigma}_{\boldsymbol{b}a}, \tag{15}$$

for the conditional variance $\sigma^2_{a|\boldsymbol{b}}$, while the conditional mean, denoted as $\mu_{a|\boldsymbol{b}}$, is determined by:

$$\mu_{a|\boldsymbol{b}} = \mu_a + \boldsymbol{\Sigma}_{a\boldsymbol{b}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{bb}}(\boldsymbol{x}_{\boldsymbol{b}} - \boldsymbol{\mu}_b). \tag{16}$$

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.

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 falls 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

$$q = P\{|x_a - \mu_{a|\boldsymbol{b}}| < 3\sigma^2_{a|\boldsymbol{b}}\} = 0.99735. \tag{17}$$

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:

$$x_\mathrm{l} = F((1 - P\{|x_a - \mu_{a|\boldsymbol{b}}| < 3\sigma^2_{a|\boldsymbol{b}}\}); \mu_{a|\boldsymbol{b}}, \sigma^2_{a|\boldsymbol{b}})^{-1}, \tag{18}$$

for the lower limit, and

$$x_{\mathrm{u}} = F((P\{|x_a - \mu_{a|\boldsymbol{b}}| < 3\sigma^2_{a|\boldsymbol{b}}\}); \mu_{a|\boldsymbol{b}}, \sigma^2_{a|\boldsymbol{b}})^{-1}, \tag{19}$$

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 (18) and (19) for all $a = 1, ..., k;\ \boldsymbol{b} = \{1, 2, \ldots, k\}$ where $a \notin \boldsymbol{b}$. The approximation of a confidence ellipse as a hypercube can be employed to represent the region of normal system operation for individual variables of a multivariate system, rendering it as an aid for visual representation.

The predicted state of the system, denoted as y_i, and the normality of signals $\boldsymbol{y}_{\mathrm{s},i}$ at time i are determined based on the maximum distance of observations from the center of the probabilistic density. The center of the probabilistic density 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:

$$F(x_a; \mu_{a|\boldsymbol{b}}, \sigma^2_{a|\boldsymbol{b}}) : a = 1, ..., k;\ \boldsymbol{b} = \{1, 2, \ldots, k\} \text{ where } a \notin \boldsymbol{b}. \tag{20}$$

Subsequently, operation states of individual inputs are defined as follows:

$$\boldsymbol{y}_{\mathrm{s},i} = \begin{cases} 0 & \text{if } T \leq (20) \\ 1 & \text{if } T > (20), \end{cases} \tag{21}$$

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 overall abnormality of the system, any anomaly in signals $\boldsymbol{y}_{\mathrm{s},i}$ is considered, resulting in:

$$y_i = \begin{cases} 1 & \text{if } 1 \in \boldsymbol{y}_{\mathrm{s},i} \\ 0 & \text{otherwise}, \end{cases} \tag{22}$$

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

This subsection provides an overview of the definition of anomalies in data analysis and their categorization, setting conventions for this paper.

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. Adaptive Anomaly Detection and Interpretation Framework

In this section, we propose an adaptive and interpretable detection framework (AID) 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 mechanisms. 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 1 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, introduced in Subsection 3.2. The length of the expiration period inversely correlates with the model's ability to adapt to sudden changes. The adaptation and detection of 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 when collective anomalies are expected to occur. In most cases, $t_\mathrm{a} = t_\mathrm{e}$ 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 (17). Adjusting this threshold can fine-tune the trade-off between precision and recall. A lower threshold boosts recall but may lower precision, while a higher 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

AID training 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 is worth noting that adaptation relies on two self-supervised methods. Adaptation routine runs if the observation at time instance i is considered normal. Furthermore, adaptation period t_a allows the model to update the distribution on collective anomalies as well, thus speeding up the adaptation to change points. Given the predicted system anomaly state from (22) 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:

$$\frac{\sum_{y \in \boldsymbol{y}_i} y}{n(\boldsymbol{y}_i)} > T. \tag{23}$$

Here $n(\boldsymbol{y}_i)$ denotes the dimensionality of $\boldsymbol{y}_i$. The logic of the (23) follows that over an adaptation period t_a, the changepoint can be discriminated from collective anomalies and point anomalies by their minimum duration, while T allows some overlap with previous normal conditions.

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 (16) and the conditional variance using (15). 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 (21).

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 (22). 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 (18) and (19), 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 spatial factors, such as multipoint measurements, temporal 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), for efficiency. 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 (23) is satisfied, change points are flagged and isolated. This initiation of change points triggers updates to the model, as stated in Subsection 3.2. ensuring it adapts to evolving data patterns, such as changes in operation state, effectively.

Algorithm 1 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 :

$i \leftarrow 1;\ n \leftarrow 1;\ T \leftarrow (17);\ \boldsymbol{\mu} \leftarrow \boldsymbol{x_0};\ \boldsymbol{\Sigma} \leftarrow \mathbf{1}_{k\times k};\ \mu_{\mathrm{t}} \leftarrow 0;\ \sigma_{\mathrm{t}}^2 \leftarrow 1;$

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 (21);

  $y_i \leftarrow$ PREDICT$(\boldsymbol{y}_{\mathrm{s},i})$ using (22);

  $\boldsymbol{x}_{\mathrm{l},i},\ \boldsymbol{x}_{\mathrm{u},i} \leftarrow$ GET$(T, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ using (18), (19);

  $y_{\mathrm{t},i} \leftarrow$ PREDICT$(t_i - t_{i-1})$ using (21);

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

  if (22) $= 0$ **or** (23) **then**

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

    if (23) **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 presents two case studies on real industrial-scale energy storages and a real data benchmark to demonstrate the effectiveness and applicability of our proposed approach. We investigate the properties and performance of the approach using signals from IoT devices in an energy system

and streamed benchmark system data. The successful deployment demonstrates that this approach is suitable for existing industrial systems utilizing IoT data streams on top of well-established SCADA systems.

4.1. Battery Energy Storage System TERRA

In the first case study, we demonstrate our proposed method on real industrial-scale battery energy storage system (BESS) TERRA, depicted in Fig 1. TERRA has an installed capacity of 151 kWh distributed among 10 modules with 20 cells. The Inverter's nominal power is 100 kW. The TERRA reports measurements of State of Charge (SoC), supply/draw energy set-points, and inner temperature, at 6 positions (channels) of each battery module. A substantial size of the system, which is 2.4x2.4x1.2m (HxWxD), requires a proper cooling mechanism. The cooling is handled by forced air from the HVAC system and inner fans, while the fire safety system is passive. Tight battery temperature control is needed to optimize performance and maximize the safety and 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.

Figure 1: Photograph of the actual studied TERRA energy storage unit with open doors (left), and closed doors (right).

The AID is integrated into the existing software infrastructure of the system, allowing detection and diagnosis of the system using streamed IoT

data. Here we replay a 9-day stream of historical measurements of the device, to demonstrate key features of AID.

For demonstration purposes, the expiration period t_e is set to 4 days, as the system is expected to adapt to the new behavior, due to the transfer of the module to the outside. The grace period was reduced to 1 day, to observe the reaction to concept drift. The threshold T is set to 3.5σ to reduce the number of alarms. The frequency will be higher as the detector is protected and self-supervised. The adaptation period t_a is changed to 3 hours as this is the time constant of the temperature to the unit change of supply/draw power demand.

Figure 2 depicts the average cell temperature measurement of the TERRA for all 10 modules. The data are normalized to the range $[0, 1]$ to protect the sensitive business value. The light red area represents the region out of dynamic operating limits as provided by AID. On 7^{th} March 2022, the system was relocated from the inside of the building to the outside power socket. The system was expected to adapt to the new behavior within 4 days as specified by t_e. Nevertheless, due to the protection of the model from learning the anomalous data, the new behavior could not be captured as the system was not operating within the safe limits. The adaptation started three days later, as only some of the measurements within the safe region after transfer were learned. Therefore, the importance of self-supervised adaptation to changes in data is crucial. As we can see, the change points detection according to (23) alerted such change shortly after the TERRA was connected to a data broker, while the length of the adaptation period enabled discrimination from collective anomaly.

In Figure 3 we depict the same measurement with a changepoint adaptation mechanism in place. The mechanism speeds up the adaptation to the new behavior, as the system is allowed to learn from anomalous data when they represent the changed behavior. The adaptation took approximately 6 times shorter.

The default sampling rate of the incoming signal measurements is 1 minute. However, network communication of the IoT devices is prone to packet dropout, which results in unexpected non-uniformities in sampling from the perspective of the SCADA system. The transfer of TERRA was accompanied by the disconnection of IoT sensors from the data broker which might be considered an anomaly. The system can detect such anomalies as well, as depicted in Figure 4. Along with known disconnection, the system alerted two more non-uniformities of shorter extend, scaled in the figure for

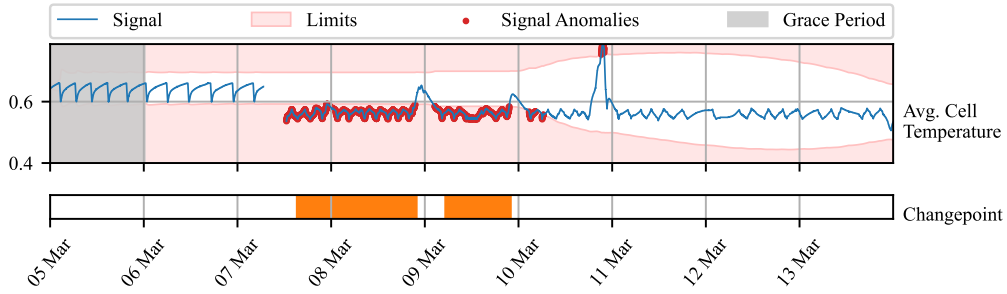


Figure 2: Time series of TERRA measurements observed over 9 days (blue line). The y-axis renders the average temperature of all cells and modules after the normalization to the range of $[0, 1]$. The light red area represents an area out of dynamic operating limits for individual signals. Observations out of the limits are marked by a red dot. Orange bars represent the times, at which changepoints were detected. Grace period is grayed out.

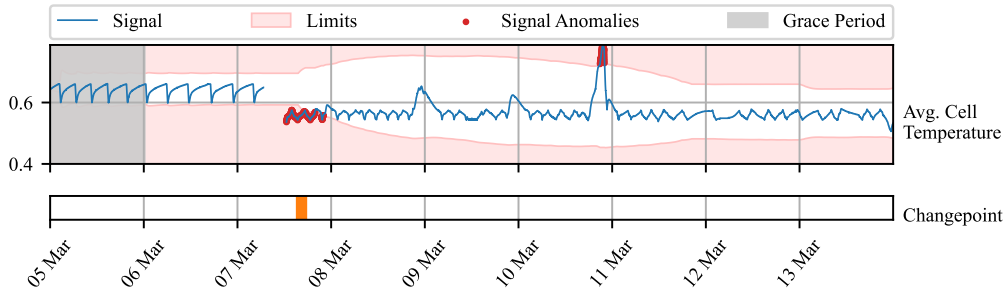


Figure 3: Time series of TERRA measurements observed over 9 days (blue line). The y-axis renders the average temperature of all cells and modules after the normalization to the range of $[0, 1]$. The light red area represents an area out of dynamic operating limits for individual signals. Observations out of the limits are marked by a red dot. Orange bars represent the times, at which changepoints were detected. Grace period is grayed out.

better visibility. The short loss of signal was caused by the packet drop, as it impacted only a few consecutive measurements. Various confidence levels could be used to further analyze and map potential causes to the duration of the outage.

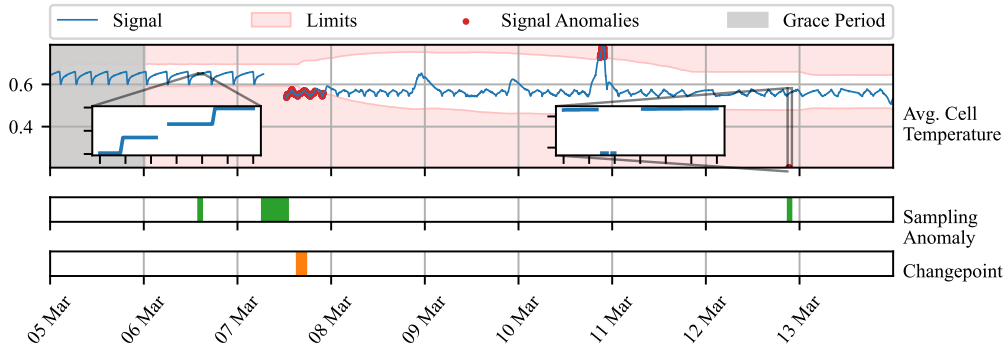


Figure 4: Time series of TERRA measurements observed over 9 days (blue line). The y-axis renders the average temperature of all cells and modules after the normalization to the range of $[0, 1]$. The light red area represents an area out of dynamic operating limits for individual signals. Observations out of the limits are marked by a red dot. Orange bars represent the times, at which changepoints were detected. Green bars represent periods where sampling anomaly was alerted. Grace period is grayed out.

Lastly, we want to acknowledge the outlier, left uncaptured due to increased variance of the distribution in a period of adaptation. Observing multiple variables, where some might be influenced less by the change in behavior, might be beneficial in such cases. The industrial partner provided a physical model of the battery module temperature, defined as follows:

$$\begin{aligned} T_{\text{bat},i+1} = T_{\text{bat},i} + T_{\text{s}}(&q_{\text{fan}} V_{\text{b,max}} \rho c_{\text{p}} (T_{\text{out}} - T_{\text{bat},i}) + V_{\text{c,max}} q_{\text{circ.fan}} \rho c_{\text{p}} T_{\text{bat},i} \\ &+ q_{\text{circ.fan}} (P_{\text{cool}} q_{\text{cool}} P_{\text{heat}} q_{\text{heat}}) + c_{\text{scale}} Q_{\text{bat}} + q_{\text{inner fans}} \\ &- (V_{\text{b,max}} q_{\text{fan}} V_{\text{c,max}} q_{\text{circ.fan}}) \rho c_{\text{p}} T_{\text{bat},i}) / (m_{\text{bat}} c_{\text{p,b}}) \end{aligned} \tag{24}$$

When combined with an averaged measurement of battery module temperature, we could compute the difference between real and predicted temperature. Such deviation can be useful in detecting unexpected patterns in temperature due to the impact of external disturbance and aging. 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. For instance, in Fig. 1 during the first two days we see, that the cooling dynamic is not captured well, resulting in a subtle positive difference between average cell temperature and the temperature predicted by the model. In combination with the raw measured average of the temperature, the AID captures the outlier on 9^{th} March which could not be captured in a univariate setting. The physical model exposes temporal aspects of the behavior as it considers the dynamics of its inputs. The rapid increase in temperature w.r.t the modeled dynamics due to environmental conditions will draw a sharp positive peak in the difference between the real and predicted temperature, which will slowly vanish. Based on the significance of the deviation, the peak will be notified as a single-point anomaly or collective anomaly.

This case study demonstrated AID's effectiveness within the context of the energy storage system, specifically the TERRA system. The AID system exhibited adaptability to changes in the operational environment, contributing to its versatility and robustness. Additionally, it facilitated the establishment of dynamic operating limits for SCADA systems, considering context of the device such as environmental conditions or aging. Furthermore, the AID system showcased its capability to opearte with a physical model, enhancing the precision of anomaly detection processes. This highlights the potential of AID as a valuable tool within complex industrial systems.

4.2. Kokam Battery Module

A second case study presents temperature profile monitoring of individual modules of battery pack TERRA deployed at the premises of the end user. During the operation, a hardware fault of module's 9 cooling fan occurred on 23^{rd} August 2023 at 17:12:30. Our industrial partner was interested in finding out, whether such an event could be captured by an anomaly detection system. Each of the 10 modules, embodies 20 cells measured by 6 spatially distributed sensors as shown in Figure 6. The measurements are sent in 30-second intervals and processed in a streamed manner by SCADA. With the availability of the temperature profiles for all the modules, we computed the deviation of the observed value from the average of all the modules' temperature measurements. The ground truth information about the fan fault was provided to the best of the operator's knowledge. However, this information serves for evaluation only, as the system operates in a self-supervised manner.

Our anomaly detection system was, in this case, initialized for the operation in production. The expiration period of 7 days, allowed the system

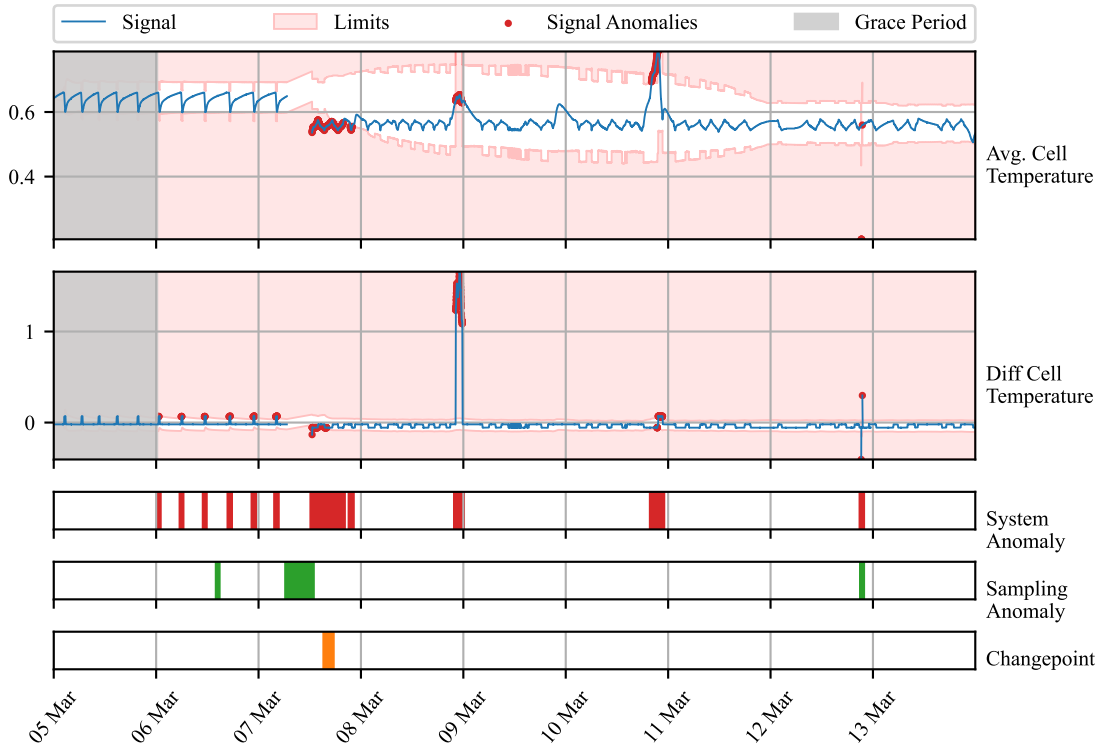


Figure 5: Time series of TERRA measurements observed over 9 days (blue line). The y-axis renders the average temperature of all cells and modules after the normalization to the range of $[0, 1]$. The light red area represents an area out of dynamic operating limits for individual signals. Observations out of the limits are marked by a red dot. Orange bars represent the times, at which changepoints were detected. Green bars represent periods where sampling anomaly was alerted. Red bars denote the period where any of the signals contained anomaly. Grace period is grayed out.

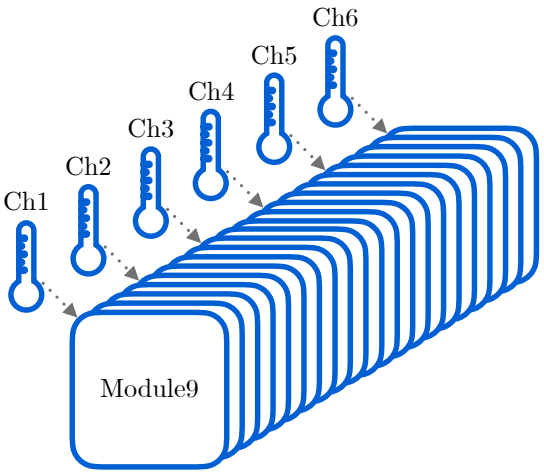


Figure 6: Module 9 with 20 cells and 6 sensors measuring the temperature at each 4^{th} cell.

to adapt to weekly seasonality due to the usage of the battery following work week. The grace period was kept at the default value, equal to t_{e}. The threshold value was shifted to a 4 sigma value of 99.977% which makes the the frequency of anomalous events approximately once a week given 30-second sampling. The adaptation period was held constant as the deployed system is not expected to change its behavior dramatically on a daily basis.

In Figure 7 we observe 4 days of operation around the period of fan fault occurrence. The deviations between the observed temperature measured by channels of module 9 and the average temperature of all modules are displayed. The dynamic operating limits tightly envelop temperatures measured by the sensors in the middle of the module (refer to Figure 6), while measurements at both sides deviate more due to the proximity to the walls and sources of disturbance. We observed multiple alarms raised by various channels individually before the fan fault. These anomalies, while not addressed here further, could be subjects of interest for further investigation by system operators. Meanwhile, the fan fault at the center of our focus is alarmed based on three measurements, namely channels 1, 2, and 3. From the zoomed views, we can observe a sharp increase in the temperature deviation. The alarm is on until 24^{th} August at noon, when significant fluctuations vanish followed by temporary settling of the temperature. On 25^{th} August

at 11:21, increased temperature fluctuations are followed by an increase of temperature similar to the initial one. AID alerts this fault again based on measurements by channels 1, 2, and 3.

Time series of TERRA measurements observed over 9 days (blue line). The y-axis renders the average temperature of all cells and modules after the normalization to the range of $[0, 1]$. The light red area represents an area out of dynamic operating limits for individual signals. Observations out of the limits are marked by a red dot. Orange bars represent the times, at which changepoints were detected. Green bars represent periods where sampling anomaly was alerted. Red bars denote the period where any of the signals contained anomaly. Grace period is grayed out.

Interestingly, during the presence of a fault in the fan, two more periods where the fan started operating again followed as depicted in Figure 8. Periods of operation were interrupted again on 27^{th} and 28^{th} August respectively in the early morning hours. In both of the cases, AID detected the presence of the fault at the moment of occurrence. In the first case, channel 3 reported an anomaly slightly before the increase in temperature, due to abnormal fluctuation happening prior to faults.

This case study demonstrates the effectiveness of the AID framework in identifying hardware faults within the context of energy storage systems. It showcases the system's ability to harness spatially distributed sensors that measure the same process variable. The AID system successfully pinpointed a fault in a cooling fan during real-world production operations, underlining its practical utility and its relevance in enhancing the safety of energy storage systems. Furthermore, the incorporation of adaptation mechanisms ensures that the system can be deployed over extended periods without necessitating resource-intensive retraining. Additionally, the concept of dynamic operating limits introduced in this study holds promise for integration with Supervisory Control and Data Acquisition (SCADA) monitoring systems, enabling proactive responses in situations where human life, equipment, or the environment may be at risk.

4.3. Real Data Benchmark

The benchmarking comparison in this subsection evaluates the AID framework against adaptive unsupervised detection methods, specifically One-Class Support Vector Machine (OC-SVM) and Half-Space Trees (HS-Trees). These methods are widely recognized for their iterative learning capabilities

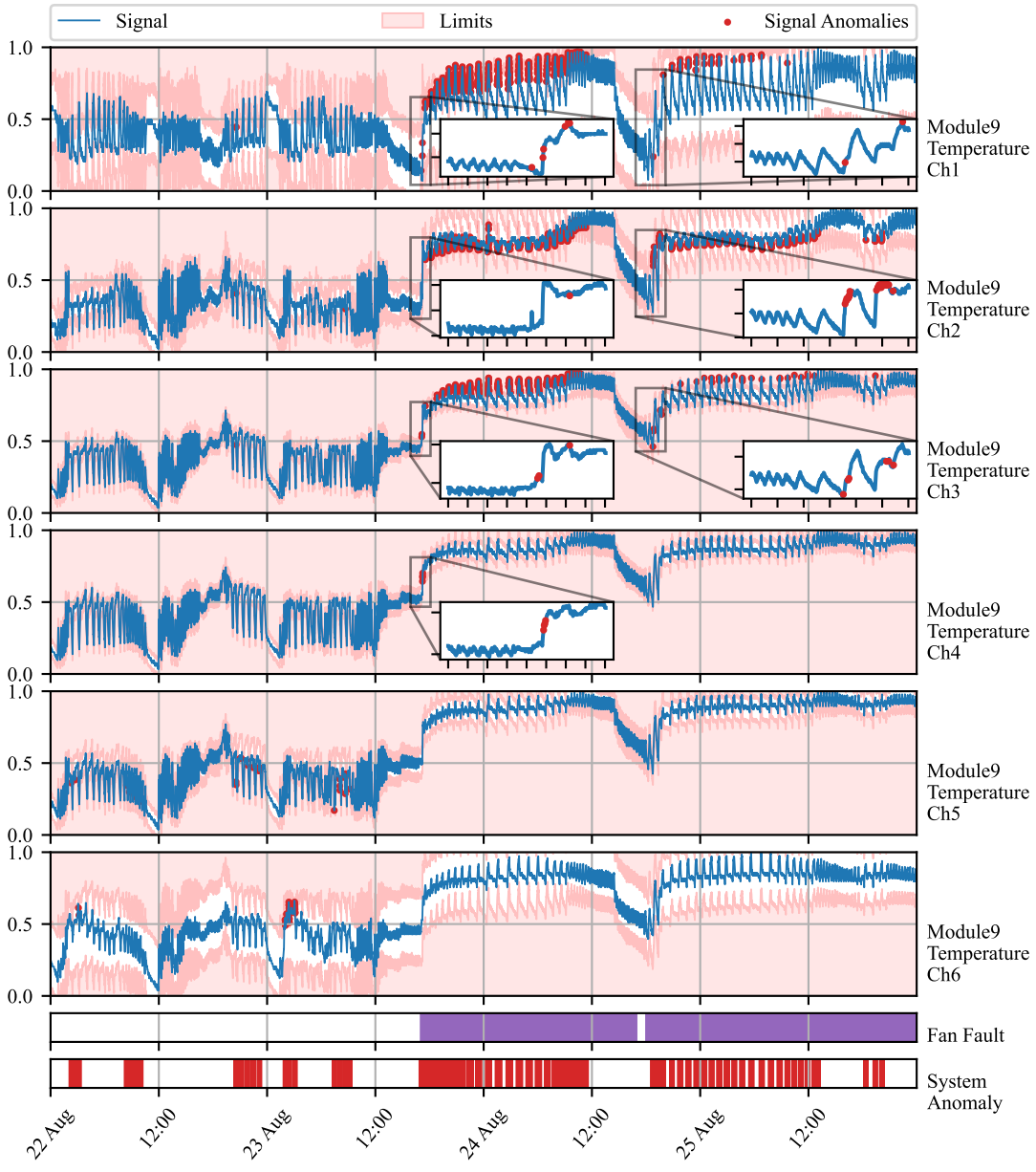


Figure 7: Time series of battery module 9 measurements (blue line). The y-axis renders the normalized deviations of temperature from the average of all modules. Signal anomalies are marked as red dots. The light red area represents an area out of dynamic operating limits. True fan faults are marked by purple bars. Green bars represent the times of changepoint detection. Red bars denote the period where any of the signals contained anomaly.

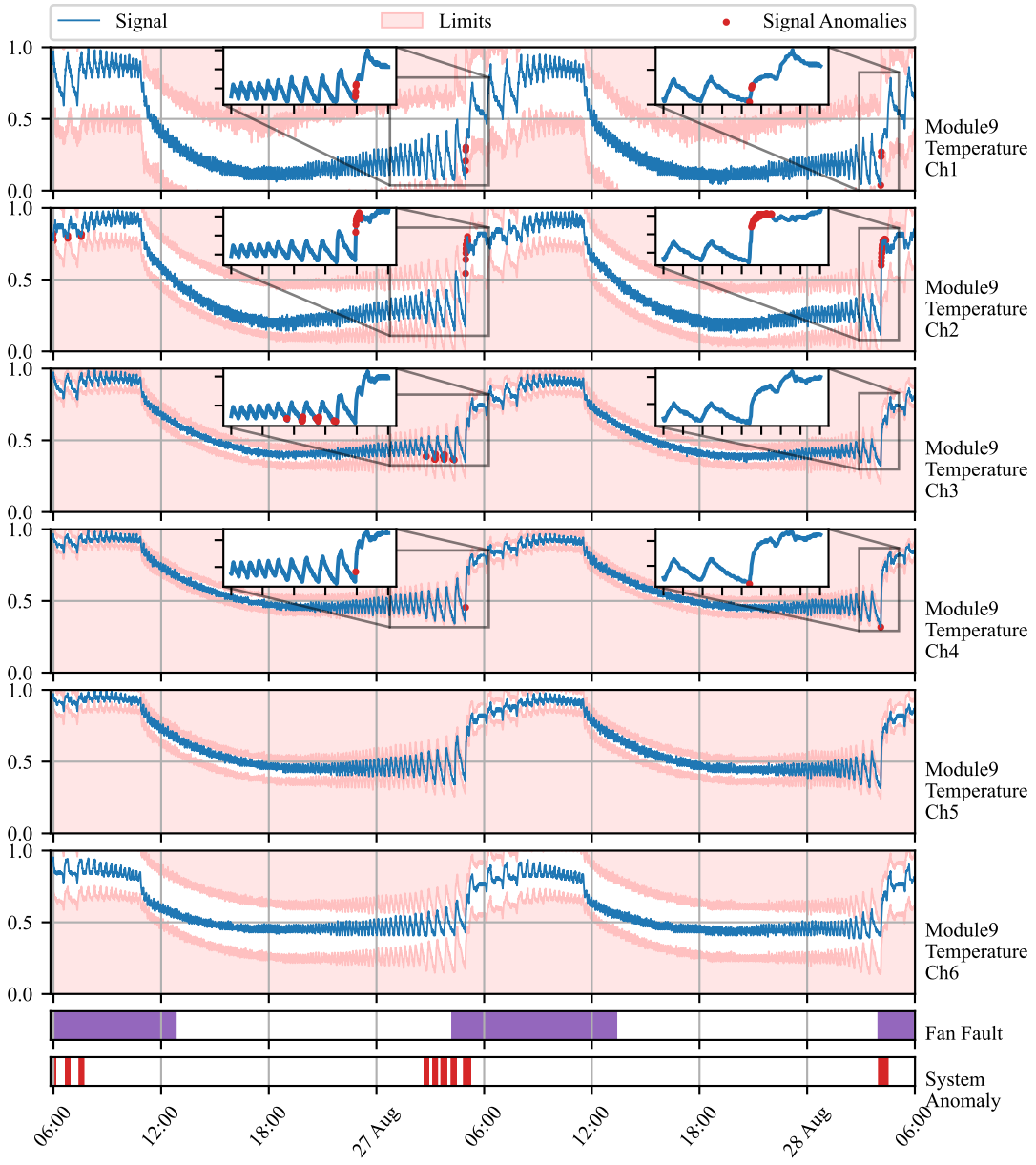


Figure 8: Time series of battery module 9 measurements (blue line). The y-axis renders the normalized deviations of temperature from an average of all modules. Signal anomalies are marked as red dots. The light red area represents an area out of dynamic operating limits. True fan faults are marked by purple bars. Green bars represent the times of changepoint detection. Red bars denote the period where any of the signals contained anomaly.

on multivariate time-series data, making them suitable for anomaly detection in dynamic systems, as previously discussed in the Introduction 1.4.

The comparison is based on the Skoltech Anomaly Benchmark (SKAB) dataset, a real-world dataset with annotated labels distinguishing between anomalous and normal observations (Katser and Kozitsin (2020)). SKAB is used for this purpose, as no established benchmarking multivariate data were found regarding energy storage systems similar to the ones studied in Subsection 4.1 and Subsection 4.2. The SKAB dataset involves experiments related to rotor imbalance, where various control actions and changes in water volume are introduced to the system. It encompasses eight features and exhibits both gradual and sudden drifts.

To ensure fairness in the benchmark, data preprocessing adheres to best practices for each method. OC-SVM employs standard scaling, while HS-Trees use normalization. Our proposed AID method requires no scaling. Preprocessing is performed online, simulating a real production environment, with running mean and variance for standard scaling and running peak-to-peak distance for normalization, as supported by the online machine learning library "river" (Montiel et al. (2021)).

The optimal hyperparameters for both reference methods are found using Bayesian Optimization. Due to no further knowledge about the data generating process, and equity in benchmark, the hyperparameters of our proposed method were optimized using Bayesian Optimization as well. 20 steps of random exploration with 100 iterations of Bayesian Optimization were used, increasing default values set in the Bayesian Optimization library, to allow thorough exploration and increase the possibility of finding global optima in each case (Nogueira (2014)). The hyperparameters are optimized with the F1 score as a cost function first, to maximize both precision and recall on anomalous samples.

As adaptation is required and anticipated within benchmark datasets, the performance is evaluated iteratively, similarly to the operation after deployment. The metric is updated with each new sample and its final value is used to drive Bayesian Optimization. The performance is evaluated using the best-performing model, found by Bayesian Optimization. The performance of the proposed method is evaluated on the same data as the models are optimized for.

Hyperparameter search ranges are specified, with values centered around default library values for OC-SVM and HS-Trees. The ranges are intentionally set wide to facilitate comprehensive exploration. The quantile filter

threshold used in OC-SVM and HS-Trees aligns with the threshold used in AID. These hyperparameter ranges are presented in Table 1.

Table 1: Hyperparameter Ranges for Detection Algorithms

Algorithm	**Hyperparameters**	**Default**	**Ranges**
AID	Threshold	0.99735	(0.85, 0.99994)
	t_e	-	(150, 10000)
	t_a	t_e	(50, 2000)
	Grace Period	t_e	(50, 1000)
OC-SVM	Threshold	-	(0.85, 0.99994)
	Learning Rate	0.01	(0.005, 0.02)
HS-Trees	Threshold	-	(0.85, 0.99994)
	N Trees	10	(0, 20)
	Max Height	8	(2, 14)
	Window Size	250	(100, 400)

The results for models optimized for the F1 score are summarized in Table 2, which includes precision, recall, F1 score, and average latency. Macro values are enclosed in brackets, representing the mean of the metric for both anomalies and normal data. A perfect detection achieves 100% in each metric. According to the Scoreboard for various algorithms on SKAB's Kaggle page, all iterative approaches perform comparably to the batch-trained isolation forest and autoencoder, validating the optimization process. Notably, the proposed AID method outperforms both reference methods in terms of F1 score, recall, and precision, despite having a 30-fold higher latency per sample. This highlights the scalability as a candidate for further development. Nevertheless, in this case, sampling of the benchmark data still offers enough time to deliver predictions with sufficient frequency.

Optimal hyperparameters found during Bayesian Optimization are detailed in Table 3. None of the parameters are at the edge of the provided ranges, serving as necessary proof of ranges being broad enough. Nevertheless, sufficient proof is not possible as multiple parameter ranges are not bounded by designed limits.

5. Conclusion

In this paper, we demonstrate the capacity of adaptive conditional probability distribution to model the normal operation of dynamic systems em-

Table 2: Evaluation of models optimized for F1 score on SKAB dataset (Katser and Kozitsin (2020)). The best-performing model is highlighted in bold. Values in brackets represent macro values of the metric.

Algorithm	Precision [%]	Recall [%]	F1 [%]	Avg. Latency [ms]
AID	**41** (59)	**80** (59)	**54** (53)	1.45
HS-Trees	36 (51)	74 (51)	48 (44)	**0.05**
OC-SVM	39 (54)	63 (54)	48 (52)	**0.05**

Table 3: Optimal hyperparameters of methods optimized for F1 score

Algorithm	**Hyperparameters**	**Found**
AID	Threshold	0.96442
	t_e	1136
	t_a	396
	Grace Period	546
OC-SVM	Threshold	0.86411
	Learning Rate	0.01956
HS-Trees	Threshold	0.99715
	N Trees	1
	Max Height	7
	Window Size	283

ploying streaming IoT data and isolate the root cause of anomalies. AID dynamically adapts to non-stationarity by updating multivariate Gaussian distribution parameters over time. Additionally, self-supervision enhances the model by protecting it from the effects of outliers and increasing the speed of adaptation in response to autonomously detected changes in operation.

Our statistical model isolates the root causes of anomalies as extreme deviations from the conditional means vector, considering spatial and temporal effects encoded in features, as demonstrated in our case studies. This approach establishes the system's operational state by analyzing the distribution of signal measurements, computing the distance from the mean of conditional probability, and setting dynamic process limits based on multivariate distribution parameters. Additionally, the detector alerts for non-uniform sampling due to packet drops and sensor malfunctions. These adaptable lim-

its can be seamlessly integrated into SCADA architecture, enhancing context awareness and enabling plug-and-play compatibility with existing infrastructure.

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 data from operated industrial-scale energy storages. These case studies highlight the model's ability to adapt, diagnose the root cause of anomalies, and leverage both physical models and spatially distributed sensors. Unlike many anomaly detection approaches, the proposed AID method does not require historical data or ground truth information about anomalies, alleviating the general limitations of detection methods employed in the energy industry.

The benchmark performed on industrial data indicates that our model provides comparable results to other self-learning adaptable anomaly detection methods. This is an important property of our model, as it also allows for root cause isolation.

AID represents a significant advancement in the safety and profitability of evolving systems that utilize well-established SCADA architecture and streaming IoT data. By providing dynamic operating limits, AID seamlessly integrates with existing alarm mechanisms commonly employed in SCADA systems. To the best of our knowledge, this study appears to be one of the initial attempts to introduce a self-supervised approach for adaptive anomaly detection and root cause isolation in SCADA-based systems utilizing IoT data streams.

Future work on this method will include improvements to the change point detection mechanism, reduction in latency for high-dimensional data, and minimizing the false positive rate, which is a challenge for general plug-and-play models. We will also explore the ability to operate with non-parametric models, in contrast to Gaussian distribution.

Our framework is openly accessible on GitHub at the following URL: `https://github.com/MarekWadinger/online_outlier_detection`.

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: 2023 24th International Conference on Process Control (PC), 2023, pp. 138–143. doi:10.1109/PC58330.2023.10217717.

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.

T. Stauffer, D. Chastain-Knight, Do not let your safe operating limits leave you s-o-l (out of luck), Process Safety Progress 40 (2021) e12163. URL: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/prs.12163. doi:https://doi.org/10.1002/prs.12163. arXiv:https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/prs.12163.

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. doi:https://doi.org/10.1049/cje.2020.09.010.

B. P. Welford, Note on a method for calculating corrected sums of squares and products, Technometrics 4 (1962) 419–420. doi:10.1080/00401706.1962.10490022.

S. Mishra, A. Datta-Gupta, Chapter 3 - distributions and models thereof, in: S. Mishra, A. Datta-Gupta (Eds.), Applied Statistical Modeling and Data Analytics, Elsevier, 2018, pp. 31–67. URL: https://www.sciencedirect.com/science/article/pii/B9780128032794000031. doi:https://doi.org/10.1016/B978-0-12-803279-4.00003-1.

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

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.

J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H. M. Gomes, J. Read, T. Abdessalem, A. Bifet, River: machine learning for streaming data in python, Journal of Machine Learning Research 22 (2021) 1–8. URL: http://jmlr.org/papers/v22/20-1380.html.

F. Nogueira, Bayesian Optimization: Open source constrained global optimization tool for Python, 2014. URL: https://github.com/fmfn/BayesianOptimization.