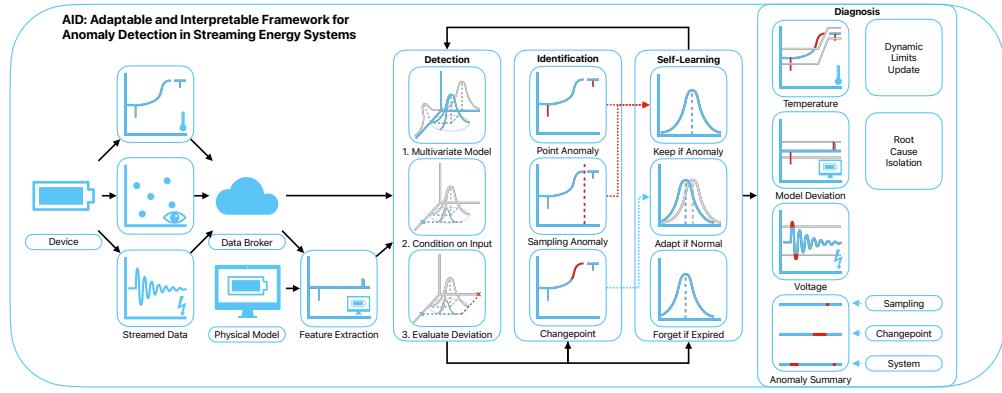


Graphical Abstract

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

- 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
- 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 1. Introduction

² Anomaly detection systems play a critical role in risk-averse systems by
³ identifying abnormal patterns and adapting to novel expected patterns in
⁴ data. These systems are particularly vital in the context of Internet of Things
⁵ (IoT) devices that continuously stream high-fidelity data to control units.

⁶ In this rapidly evolving field with long-spanning roots, Chandola et al.
⁷ conducted an influential review of prior research efforts across diverse appli-
⁸ cation domains (Chandola et al. (2009)). Recent studies have underscored
⁹ the need for holistic and tunable anomaly detection methods accessible to
¹⁰ operators (Laptev et al. (2015); Kejariwal (2015); Cook et al. (2020)).

¹¹ Cook et al. denote substantial aspects that pose challenges to anomaly
¹² detection in IoT, including the temporal, spatial, and external context of
¹³ measurements, multivariate characteristics, noise, and nonstationarity (Cook
¹⁴ et al. (2020)). To address these complexities, Zhang et al. have successfully
¹⁵ employed spatially distributed sensors and time-relative modulation. Their
¹⁶ approach has proven effective, particularly in the context of complex non-
¹⁷ linear systems, offering potential solutions to some of the challenges posed
¹⁸ by IoT data (Zhang et al. (2024)). Huang et al., on the other hand, tackled
¹⁹ the problems of detecting global outliers, local outliers, and outlier clusters
²⁰ simultaneously. Their proposed approach, based on density estimation, relies
²¹ on the notion that density distributions should exhibit minimal variations in
²² local areas. To achieve this, they introduce a novel turning ratio metric,
²³ which reduces reliance on hyperparameters and enhances anomaly detection
²⁴ (Huang et al. (2023)).

²⁵ Additionally, feature engineering techniques play a crucial role in cap-
²⁶ turing contextual properties and enhancing anomaly detection performance
²⁷ (Fan et al. (2019)). However, it's worth noting that feature engineering may
²⁸ introduce categorical variables and significantly increase dimensionality of
²⁹ the data, requiring specific methods for handling large data, sizeable data
³⁰ storage, and substantial computational resources (Talagala et al. (2021)).
³¹ Recently, Li et al. introduced an attribute-weighted outlier detection algo-
³² rithm, designed for high-dimensional datasets with mixtures of categorical
³³ and numerical data. Their approach assigns different weights to individ-
³⁴ ual attributes based on their importance in anomaly detection and uses
³⁵ 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

72 data (Zhang et al. (2023)). Yang et al. recently proposed a Bayesian net-
73 work (BN) for fault detection and diagnosis. In this BN, individual nodes
74 of the network represent normally distributed variables, whereas the multi-
75 ple regression model defines weights and relationships. Using the predefined
76 structure of the BN, the authors propose an offline training with online de-
77 tection and diagnosis (Yang et al. (2022)).

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

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

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

100 *1.2. Novelty of proposed approach*

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

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

114 A limited number of studies have focused on adaptation and interpretabil-
115 ity within the framework of anomaly detection. Two recent contributions
116 in this area are (Steenwinckel (2018)) and (Steenwinckel et al. (2021)). In
117 (Steenwinckel (2018)), the authors emphasize the importance of combining
118 prior knowledge with a data-driven approach to achieve interpretability, par-
119 ticularly concerning root cause isolation. They propose a novel approach
120 that involves extracting features based on knowledge graph pattern extrac-
121 tion and integrating them into the anomaly detection mechanism. This graph
122 is subsequently transformed into a matrix, and adaptive region-of-interest ex-
123 traction is performed using reinforcement learning techniques. To enhance
124 interpretability, a Generative Adversarial Network (GAN) reconstructs a new
125 graphical representation based on selected vectors. However, it's important
126 to note that the validation of this idealized approach is pending further in-
127 vestigation. Lately, (Steenwinckel et al. (2021)) introduced a comprehen-
128 sive framework for adaptive anomaly detection and root cause analysis in
129 data streams. While the adaptation process is driven by user feedback, the
130 specific mechanism remains undisclosed. The authors present an interpreta-
131 tion of their method through a user dashboard, featuring visualizations of
132 raw data. This dashboard is capable of distinguishing between track-related
133 problems and train-related issues, based on whether multiple trains at the
134 same geographical location approach the anomaly. Meanwhile, our attempts
135 aim to develop a self-supervised method capable of learning without human
136 supervision which is often limited in time and poses significant delays in
137 adaptation, while interpretation offers straightforward statistical reasoning
138 and root cause isolation.

139 *1.3. Validation*

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

146 well-established unsupervised anomaly detection methods in streamed data
147 which create a bedrock for many state-of-the-art contributions, such as One-
148 Class SVM (Amer et al. (2013); Liu et al. (2014); Krawczyk and Woźniak
149 (2015); Miao et al. (2019); Gözüaçık and Can (2021)), and Half-Space Trees
150 (Wetzig et al. (2019); Lyu et al. (2020)).

151 *1.4. Broader Impact*

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

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

- 165 • Enriches interpretable anomaly detection with adaptive capabilities
- 166 • Isolates root cause of anomalies while considering interactions
- 167 • Uses self-learning approach on streamed IoT data
- 168 • Demonstrates interpretability by providing process limits for signals
- 169 • Utilizes existing IT infrastructure

170 *1.5. Paper Organization*

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

177 **2. Preliminaries**

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

187 *2.1. Welford's Online Algorithm*

188 Welford introduced a numerically stable online algorithm for calculating
189 mean and variance in a single pass through data. Therefore, the algorithm
190 allows the processing of IoT device measurements without the need to store
191 their values Welford (1962).

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

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

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

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

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

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

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

$$s_n^2 = \frac{S_n}{n-1}. \quad (4)$$

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

202 2.2. Inverting Welford's Algorithm

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

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

209 where the reverted mean is given as

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

210 Finally, the unbiased variance follows the formula:

$$s_{n-1}^2 = \frac{S_{n-1}}{n-2}. \quad (7)$$

211 Notably, inversion allows the algorithm to keep constant rate of adapta-
 212 tion at cost of storing a bounded data buffer.

213 2.3. Statistical Model of Multivariate System

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

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

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

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

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

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

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

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

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

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

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

Initialisation :

1: $f \leftarrow 10; l \leftarrow -f; r \leftarrow f;$

LOOP Process

2: **while** $F(l; \bar{x}_n, s_n^2) > 0$ **do**

3: $r \leftarrow l;$

4: $l \leftarrow lf;$

5: **end while**

6: **while** $F_X(r) - q < 0$ **do**

7: $l \leftarrow r;$

8: $r \leftarrow rf;$

9: **end while**

10: $\tilde{x}_{q,n} = \arg \min_{x_n} \|F(x_n; \bar{x}_n, s_n^2) - q\|$ s.t. $l \leq x_n \leq r$

11: **return** $\tilde{x}_{q,n} \sqrt{s_n^2 + \bar{x}_n}$

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

246 *2.4. Conditional Probability Distribution*

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

$$\mathbf{x}_i = \begin{bmatrix} x_a \\ \mathbf{x}_b \end{bmatrix} \text{ with dimensions } \begin{bmatrix} 1 \times 1 \\ (k-1) \times 1 \end{bmatrix}, \quad (11)$$

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

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

255 and

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

256 where a and \mathbf{b} represent distinct components within the vector.

257 Subsequently, we can derive the conditional distribution of any subset
258 variable x_a , given the complementary vector \mathbf{x}_b . This conditional distribution
259 conforms to a univariate normal distribution, characterized by:

$$X_a | \mathbf{X}_b \sim \mathcal{N}(\mu_{a|\mathbf{b}}, \sigma_{a|\mathbf{b}}^2). \quad (14)$$

260 where $\mu_{a|\mathbf{b}}$ denotes the conditional mean and $\sigma_{a|\mathbf{b}}^2$ represents the condi-
261 tional variance. These crucial parameters can be computed using the Schur
262 complement.

263 For a general matrix M expressed as:

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

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

$$M | D = A - BD^{-1}C. \quad (16)$$

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

$$\sigma_{a|\mathbf{b}}^2 = \sigma_{aa}^2 - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}, \quad (17)$$

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

$$\mu_{a|\mathbf{b}} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b). \quad (18)$$

268 It is important to note that Σ_{bb}^{-1} in Equations (17) and (18) signifies
269 the inverse of the covariance matrix Σ_{bb} . Thus, the conditional variance
270 $\sigma_{a|\mathbf{b}}^2$ essentially represents the Schur complement of Σ_{bb} within the overall
271 covariance matrix Σ .

272 2.5. Gaussian Anomaly Detection

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

278 From a statistical viewpoint, outliers can be denoted as values that sig-
279 nificantly 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 $\mu_{a|\mathbf{b}}$ of 3 conditional variances $\sigma_{a|\mathbf{b}}^2$ and gives an approximate value of q as

$$q = P\{|x_a - \mu_{a|\mathbf{b}}| < 3\sigma_{a|\mathbf{b}}^2\} = 0.99735. \quad (19)$$

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:

$$x_l = F((1 - P\{|x_a - \mu_{a|\mathbf{b}}| < 3\sigma_{a|\mathbf{b}}^2\}); \mu_{a|\mathbf{b}}, \sigma_{a|\mathbf{b}}^2)^{-1}, \quad (20)$$

for the lower limit, and

$$x_u = F((P\{|x_a - \mu_{a|\mathbf{b}}| < 3\sigma_{a|\mathbf{b}}^2\}); \mu_{a|\mathbf{b}}, \sigma_{a|\mathbf{b}}^2)^{-1}, \quad (21)$$

for the upper limit. These lower and upper limits together form vectors \mathbf{x}_l and \mathbf{x}_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 $a = 1, \dots, k$; $\mathbf{b} = \{1, 2, \dots, k\}$ where $a \notin \mathbf{b}$.

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 normality of signals $\mathbf{y}_{s,i}$ 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 $\mu_{a|\mathbf{b}}$ with

310 respect to other features. The calculation of this distance involves the cumulative
 311 distribution function (CDF) of observations and conditional distributions, as follows:
 312

$$F(x_a; \mu_{a|\mathbf{b}}, \sigma_{a|\mathbf{b}}^2) : a = 1, \dots, k; \mathbf{b} = \{1, 2, \dots, k\} \text{ where } a \notin \mathbf{b}. \quad (22)$$

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

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

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

316 For the overall abnormality of the system, any anomaly in signals $\mathbf{y}_{s,i}$ is
 317 considered, resulting in:

$$y_i = \begin{cases} 1 & \text{if } 1 \in \mathbf{y}_{s,i} \\ 0 & \text{otherwise,} \end{cases} \quad (24)$$

318 defining the discrimination boundary between system operation where
 319 $y_i = 0$ indicates normal system operation, and $y_i = 1$ indicates anomalous
 320 operation.

321 2.6. Anomaly Definition

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

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

332 Nevertheless, this conventional method may not be suitable for scenarios
 333 characterized by collective anomalies, where clusters of abnormal data points
 334 coexist. A more pragmatic approach defines anomalies as deviations from
 335 established "normal" patterns, resembling the principles of semi-supervised
 336 learning. Change point detection, in a similar vein, can be regarded as a

337 relative approach that takes into account the varying dynamics of changes,
338 whether they occur gradually or abruptly (Iglesias Vázquez et al. (2023)).

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

345 Notwithstanding the differences in terminology, methods employed for the
346 identification of data points residing in low-probability regions, irrespective
347 of whether they are referred to as "anomaly detection," "outlier detection,"
348 or "novelty detection," share fundamental similarities (Iglesias Vázquez et al.
349 (2023)).

350 **3. Adaptive Anomaly Detection and Interpretation Framework**

351 In this section, we propose an adaptive and interpretable detection frame-
352 work (AID) for multivariate systems with streaming IoT devices. This ap-
353 proach models the system as a dynamic joint normal distribution, enabling
354 it to effectively adapt to pervasive nonstationary effects on processes. Our
355 method handles various factors, including change points, concept drift, and
356 seasonal effects. Our primary contribution lies in the fusion of an adaptable
357 self-supervised system with root cause identification capabilities. This combi-
358 nation empowers the online statistical model to diagnose anomalies through
359 two distinct mechanisms. Firstly, it employs conditional probability calcu-
360 lations to assess the system's operating conditions' normality. Secondly, it
361 identifies outliers within individual signal measurements and features based
362 on dynamic alert-triggering process limits. In the following sections, we de-
363 scribe our proposed methodology across three subsections. The initial sub-
364 section delves into the process of initializing the model's parameters. The
365 subsequent section describes online training and adaptation, while the final
366 subsection expounds upon the model's detection and diagnostic capabilities.
367 For a concise representation of the proposed method, Algorithm 2 is provided.

368 *3.1. Model Parameters Initialization*

369 The model initialization is governed by defining two tunable hyperparam-
370 eters of the model: the expiration period (t_e) and the threshold (T). The
371 expiration period determines the window size for time-rolling computations,

372 impacting the proportion of outliers within a given timeframe, and directly
373 influencing the relaxation (with a longer expiration period) or tightening
374 (with a shorter expiration period) of dynamic signal limits. Additionally, we
375 introduce a grace period, which defaults to $3/4t_e$, allowing for model calibra-
376 tion. During this grace period, system anomalies are not flagged to prevent
377 false positives and speed up self-supervised learning, introduced in Subsec-
378 tion 3.2. The length of the expiration period inversely correlates with the
379 model’s ability to adapt to sudden changes. The adaptation and detection
380 of shifts in the data-generating process, such as changes in mean or variance,
381 is managed through the adaptation period t_a . A longer t_a results in slower
382 adaptation but potentially longer alerts, which can be valuable when colec-
383 tive anomalies are expected to occur. In most cases, $t_a = t_e$ offers optimal
384 performance.

385 As a general rule of thumb, expiration period t_e should be determined
386 based on the slowest observed dynamics within the multivariate system. The
387 threshold T defaults to the three-sigma probability of q in (19). Adjusting
388 this threshold can fine-tune the trade-off between precision and recall. A
389 lower threshold boosts recall but may lower precision, while a higher thresh-
390 old enhances precision at the cost of recall. The presence of one non-default
391 easily interpretable hyperparameter facilitates adaptability to various sce-
392 narios. We recommend starting with the default values of other parameters
393 and making adjustments based on real-time model performance.

394 3.2. Online training

395 Training in AID follows an incremental learning approach, processing
396 each new sample upon arrival. Incremental learning allows online parame-
397 ter updates, albeit with a potential computational delay affecting response
398 latency.

399 In the case of a dynamic joint probability distribution, the parameters are
400 μ_i and Σ_i at time instance i . Update of the mean vector μ_i and covariance
401 matrix Σ_i is governed by Welford’s online algorithm using equation (2) and
402 (4) respectively. Samples beyond the expiration period t_e are disregarded
403 during the second pass. The effect of expired samples is reverted using inverse
404 Welford’s algorithm for mean (6) and variance (7), accessing the data in the
405 buffer. For details, refer to Subsection 2.2.

406 It’s worth noting that adaptation relies on two self-supervised methods.
407 Adaptation routine runs if the observation at time instance i is considered
408 normal. Adaptation period t_a allows the model to update the distribution

409 on outliers as well. Given the predicted system anomaly state from (24) as
 410 y_i over the window of past observations $\mathbf{y}_i = \{y_{i-t_a}, \dots, y_i\}$, the following test
 411 holds when adaptation is performed on outlier:

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

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

415 *3.3. Online prediction*

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

418 Firstly, we calculate the parameters of the conditional distribution con-
 419 cerning the dynamic multivariate Gaussian distribution. These calculations
 420 are performed for the process observation vector \mathbf{x}_i at time instance i . Specif-
 421 ically, we compute the conditional mean using (18) and the conditional vari-
 422 ance using (17). These computations yield univariate conditional distri-
 423 butions for individual signals and features. These conditional distributions play
 424 a crucial role in assessing the abnormality of signals and features concerning
 425 other observed values. This assessment relies on the strength of relation-
 426 ships defined by the covariance matrix of the dynamic multivariate Gaussian
 427 distribution. Consequently, our approach inherently considers the interac-
 428 tions between input signals and features. The determination of anomalous
 429 behavior is governed by (23).

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

434 To assist operators in their assessments, we establish a hypercube defined
 435 by lower and upper threshold values, denoted as \mathbf{x}_l and \mathbf{x}_u , respectively.
 436 These thresholds are derived from (20) and (21), incorporating updated
 437 model parameters. Lower and upper thresholds play a pivotal role as dy-
 438 namic process limits. They replace the conservative process limits provided
 439 in sensor documentation, accounting for spatial factors, such as multipoint
 440 measurements and temporal factors such as aging, and actual environmental
 441 conditions that influence sensor operation.

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

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

Algorithm 2 Online Detection and Identification Workflow

Input: expiration period t_e

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

Initialisation :

- 1: $i \leftarrow 1; n \leftarrow 1; T \leftarrow (19); \boldsymbol{\mu} \leftarrow \mathbf{x}_0; \boldsymbol{\Sigma} \leftarrow \mathbf{1}_{k \times k}; \mu_t \leftarrow 0; \sigma_t^2 \leftarrow 1;$
 - 2: compute $F(\mathbf{x}_0; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ using algorithm in Genz (2000);
LOOP Process
 - 3: **loop**
 - 4: $\mathbf{x}_i, t_i \leftarrow \text{RECEIVE}();$
 - 5: $\mathbf{y}_{s,i} \leftarrow \text{PREDICT}(\mathbf{x}_i, T)$ using (23);
 - 6: $y_i \leftarrow \text{PREDICT}(\mathbf{y}_{s,i})$ using (24);
 - 7: $\mathbf{x}_{l,i}, \mathbf{x}_{u,i} \leftarrow \text{GET}(T, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ using Algorithm 1;
 - 8: $y_{t,i} \leftarrow \text{PREDICT}(t_i - t_{i-1})$ using (23);
 - 9: $\mu_t, \sigma_t^2 \leftarrow \text{UPDATE}(t_i - t_{i-1}, \mu_t, \sigma_t^2)$ using (2), (4);
 - 10: **if** not (24) **or** (25) **then**
 - 11: $\boldsymbol{\mu}, \boldsymbol{\Sigma} \leftarrow \text{UPDATE}(\mathbf{x}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}, n)$ using (2), (4);
 - 12: **if** (25) **then**
 - 13: $y_{c,i} \leftarrow 1;$
 - 14: **else**
 - 15: $y_{c,i} \leftarrow 0;$
 - 16: **end if**
 - 17: $n \leftarrow n + 1;$
 - 18: **for** \mathbf{x}_{i-t_e} **do**
 - 19: $\boldsymbol{\mu}, \boldsymbol{\Sigma} \leftarrow \text{REVERT}(\mathbf{x}_{i-t_e}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, n)$ using (6), (7);
 - 20: $n \leftarrow n - 1;$
 - 21: **end for**
 - 22: **end if**
 - 23: $i \leftarrow i + 1;$
 - 24: **end loop**
-

455 **4. Case Study**

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

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

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

464 *4.1. Benchmark*

465 In this subsection, we compare the proposed method with adaptive un-
466 supervised detection methods without an interpretability layer. Two of the
467 well-established methods, providing iterative learning capabilities over mul-
468 tivariate time-series data are One-Class Support Vector Machine (OC-SVM)
469 and Half Spaced Trees (HS-Trees). Both methods represent the backbone
470 of multiple state-of-the-art methods for cases of anomaly detection on dy-
471 namic system data, with a brief list of recent applications in Introduction
472 1.3. Comparison is conducted on real benchmarking data, annotated with
473 labels of whether the observation was anomalous or normal. The dataset of
474 Skoltech Anomaly Benchmark (SKAB) Katser and Kozitsin (2020) is used for
475 this purpose, as no established benchmarking multivariate data were found
476 regarding energy storage systems. It represents a combination of experiments
477 with the behavior of rotor imbalance as a subject to various functions intro-
478 duced to control action as well as slow and sudden changes in the amount
479 of water in the circuit. The system is described by 8 features. The data
480 were preprocessed according to best practices for the given method, namely:
481 standard scaling for OC-SVM, normalization for HS-Trees, while no scaling
482 was required by our proposed method. The optimal quantile threshold value
483 for both reference methods is found using Bayesian Optimization. Due to no
484 further knowledge about the process, the parameters of the proposed method
485 were optimized using Bayesian Optimization as well. Results are provided
486 within Table 1, evaluating F1 score, Recall and Precision. A value of 100%
487 at each metric represents a perfect detection. The latency represents the
488 average computation time per sample of the pipeline including training and
489 data preprocessing.

490 The results in Table 1 suggest, that our algorithm provides slightly better
491 performance than reference methods. Based on the Scoreboard for various
492 algorithms on SKAB’s Kaggle page, our iterative approach performs com-
493 parably to the evaluated batch-trained model. Such a model has all the
494 training data available before prediction unlike ours, evaluating the metrics
495 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

496 *4.2. Battery Energy Storage System (BESS)*

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

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

511 The industrial partner provided a physical model of the battery cell tem-
 512 perature, defined as follows:

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

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

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

523 Fig. 1 depicts the operation of the BESS over March 2022. Multiple
524 events of anomalous behavior happened within this period, confirmed by the
525 operators, that are observable through a sudden or significant shift in mea-
526 surements in a given period. As the first step, the detection mechanism was
527 initialized, following the provided guidelines for parameter selection in Sub-
528 section 3.1. The expiration period was set to $t_e = 7$ days, due to the weekly
529 seasonality of human behavior impacting battery usage. The threshold was
530 kept at default value $T = 0.99735$. A grace period, during which the model
531 learns from both normal and anomalous data (though normal are expected,
532 yet not required here), is shortened to 2.5 days to observe the effect of BESS
533 calibration happening on 3rd day from deployment.

534 The deployment and operation of the anomaly detection system were suc-
535 cessful as shown by its adaptation of changepoint on 7th March 2022 that
536 appeared due to the relocation of the battery storage system outdoors. The
537 model was adapted online based on Subsection 3.2. The sudden shift in envi-
538 ronmental conditions, due to the transfer of the system to outside changed the
539 dynamics of the system’s temperature. However, new behavior was adopted
540 by the top-level anomaly isolation system within five days, reducing potential
541 false alerts afterward, by observably shifting the conditional mean to lower
542 temperatures. Perhaps more interesting are the alerted changepoints.

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

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

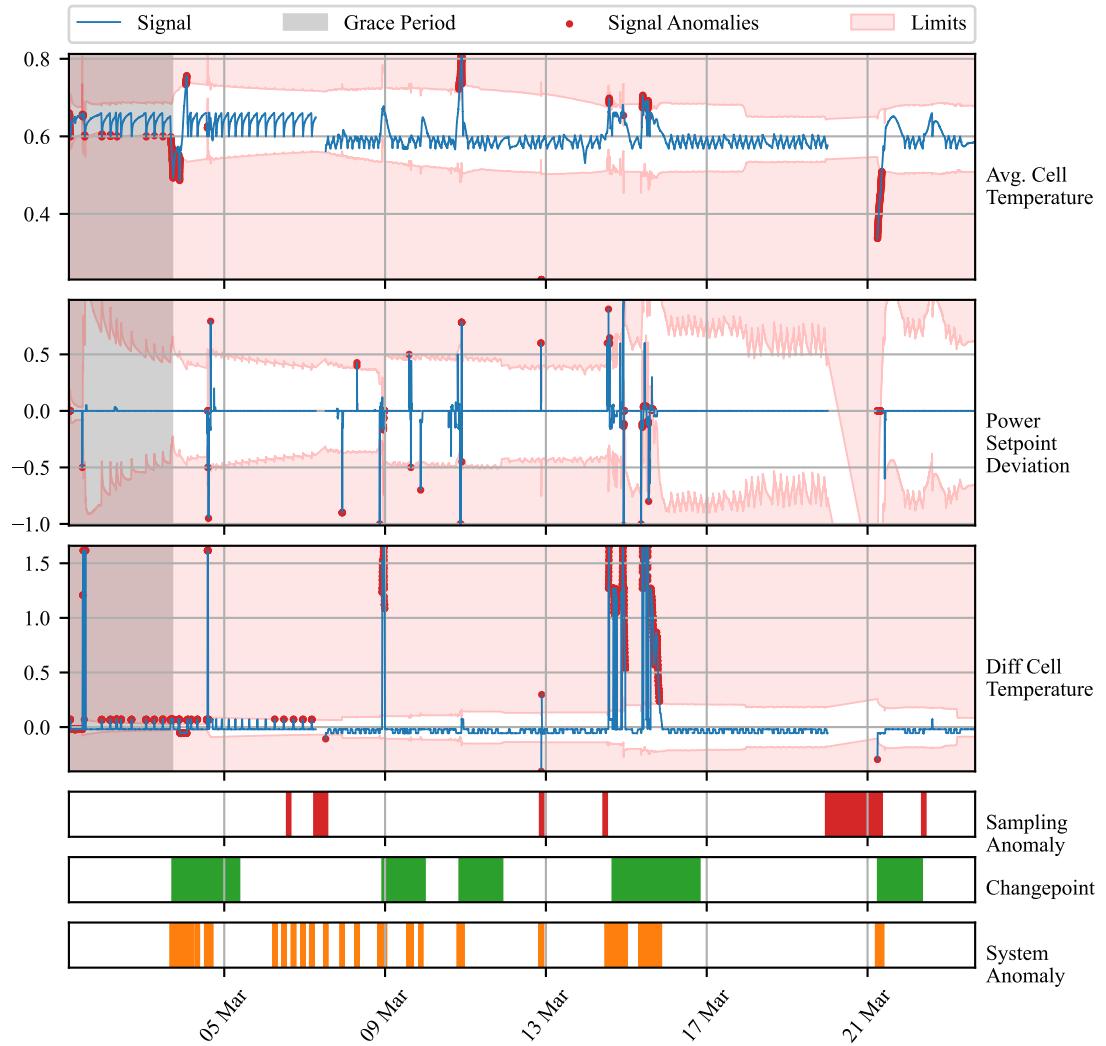


Figure 1: Time Series of BESS measurements (blue line) of process variables. The y-axis renders the values after the normalization of raw inputs. Root causes of anomalies are marked within specific signals as red dots. The light red area represents out-of-limits values for individual signals. Non-uniform 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.

557 4.3. Kokam Battery Temperature Module

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

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

571 In Figure 2 we observe 5 days of deviations between the observed tem-
572 perature measured by channels of module 9 and the average temperature
573 of all modules. After the grace period, we observe multiple system alarms
574 raised by various channels. Until the noon of 22nd August, they seem to be
575 spread out randomly between individual channels. During the late evening
576 of 22nd, anomalies were reported by both channels 4 and 5 for a prolonged
577 period, followed by an anomalous rise in temperature measured by channel
578 6 early in the morning on 23rd August. The fan fault was observed approx-
579 imately at 5 pm on 23rd August. Our anomaly detection system instantly
580 raised an alarm, notifying us of anomalous behavior reported by channels 1
581 - 3. The prolonged duration of the alarm triggered the changepoint alarm
582 approximately 2 hours later. This resulted in a slightly faster adaptation of
583 the system to the new operation under increased temperature. Surprisingly,
584 the temperature decreased during the next day, notifying us of the fan be-
585 ing in operation for a brief period, to fail again 30 minutes later after the
586 battery modules were cooled down to the previous setpoint. The anomaly
587 detection system was triggered once again, although adaptation loosened the
588 region of normal operation to allow itself to adapt. No significant anomalies
589 in sampling were observed during the period.

590 **5. Conclusion**

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

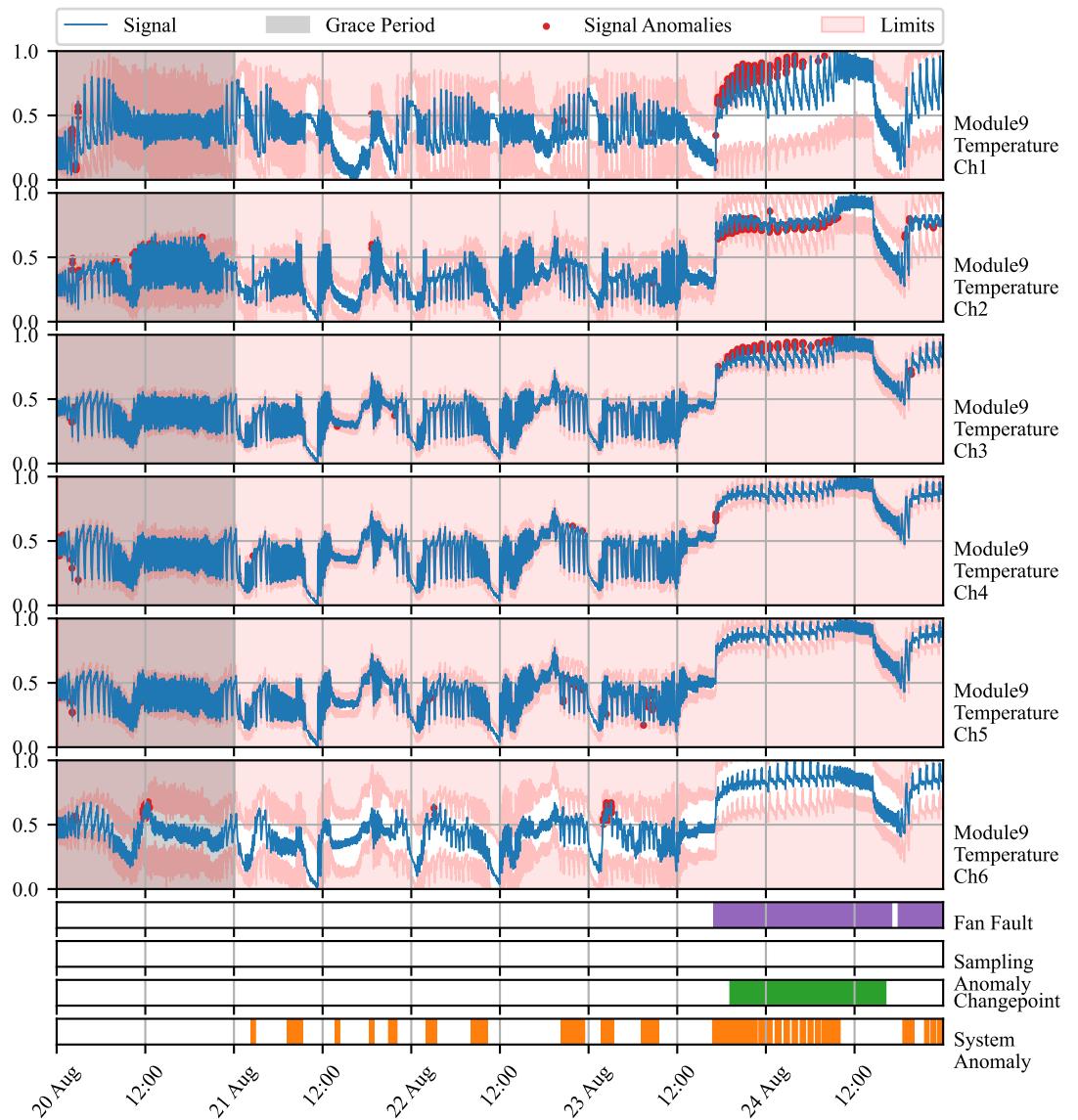


Figure 2: Time Series of battery module 9 measurements (blue line) of process variables. The y-axis renders the normalized deviations of temperature from average of all 12 modules. Signal anomalies are marked as red dots. The light red area represents out-of-limits values for individual signals. True fan faults are marked by purple bars. Green bars represent the times, at which changepoint was detected. All the signal anomalies are depicted as orange bars below the graph.

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

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

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

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

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

631 demonstrated the ability to capture system anomalies and provide less con-
632 servative limits to signals. The physical model aided decisions about the
633 normality of the measured temperature of BESS.

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

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

645 References

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

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

655 A. Kejariwal, Introducing practical and robust anomaly
656 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.

659 A. A. Cook, G. Misirli, Z. Fan, Anomaly detection for iot time-series data:
660 A survey, IEEE Internet of Things Journal 7 (2020) 6481–6494. doi:10.
661 1109/JIOT.2019.2958185.

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

- 665 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>.
- 666
- 667
- 668
- 669 J. Huang, D. Cheng, S. Zhang, A novel outlier detecting algo-
670 rithm based on the outlier turning points, Expert Systems with
671 Applications 231 (2023) 120799. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423013015>. doi:<https://doi.org/10.1016/j.eswa.2023.120799>.
- 672
- 673
- 674 C. Fan, Y. Sun, Y. Zhao, M. Song, J. Wang, Deep learning-based fea-
675 ture 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>.
- 676
- 677
- 678
- 679 P. D. Talagala, R. J. Hyndman, K. Smith-Miles, Anomaly detection
680 in high-dimensional data, Journal of Computational and Graphi-
681 cal Statistics 30 (2021) 360–374. URL: <https://doi.org/10.1080/10618600.2020.1807997>.
682 doi:<https://doi.org/10.1080/10618600.2020.1807997>.
683 arXiv:<https://doi.org/10.1080/10618600.2020.1807997>.
- 684
- 685 J. Li, Z. Liu, Attribute-weighted outlier detection for mixed data
686 based on parallel mutual information, Expert Systems with Appli-
687 cations 236 (2024) 121304. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423018067>. doi:<https://doi.org/10.1016/j.eswa.2023.121304>.
- 688
- 689 X. Du, J. Chen, J. Yu, S. Li, Q. Tan, Generative adversar-
690 ial nets for unsupervised outlier detection, Expert Systems with
691 Applications 236 (2024) 121161. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423016639>. doi:<https://doi.org/10.1016/j.eswa.2023.121161>.
- 692
- 693
- 694 M. Salehi, L. Rashidi, A survey on anomaly detection in evolving data:
695 [with application to forest fire risk prediction], SIGKDD Explor. Newsl.
696 20 (2018) 13–23. URL: <https://doi.org/10.1145/3229329.3229332>.
697 doi:<https://doi.org/10.1145/3229329.3229332>.

- 698 N. Barbosa Roa, L. Travé-Massuyès, V. H. Grisales-Palacio, Dy-
699 clee: Dynamic clustering for tracking evolving environments, Pat-
700 tern 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>.
- 703 A. G. Tartakovsky, A. S. Polunchenko, G. Sokolov, Efficient computer net-
704 work anomaly detection by changepoint detection methods, IEEE Journal
705 of Selected Topics in Signal Processing 7 (2013) 4–11. doi:[10.1109/JSTSP.2012.2233713](https://doi.org/10.1109/JSTSP.2012.2233713).
- 707 H. Wu, J. He, M. Tömösközi, Z. Xiang, F. H. Fitzek, In-network processing
708 for low-latency industrial anomaly detection in softwarized networks, in:
709 2021 IEEE Global Communications Conference (GLOBECOM), 2021, pp.
710 01–07. doi:[10.1109/GLOBECOM46510.2021.9685489](https://doi.org/10.1109/GLOBECOM46510.2021.9685489).
- 711 H. S. Pannu, J. Liu, S. Fu, Aad: Adaptive anomaly detection system for cloud
712 computing infrastructures, in: 2012 IEEE 31st Symposium on Reliable
713 Distributed Systems, 2012, pp. 396–397. doi:[10.1109/SRDS.2012.3](https://doi.org/10.1109/SRDS.2012.3).
- 714 S. Ahmad, A. Lavin, S. Purdy, Z. Agha, Unsupervised real-time anomaly
715 detection for streaming data, Neurocomputing 262 (2017) 134–
716 147. 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.
- 719 H. H. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, A. Liotta, Ensembles
720 of incremental learners to detect anomalies in ad hoc sensor networks,
721 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
723 Sensing, Processing and Networking Technologies.
- 725 M. Carletti, C. Masiero, A. Beghi, G. A. Susto, Explainable machine learning
726 in industry 4.0: Evaluating feature importance in anomaly detection to
727 enable root cause analysis, in: 2019 IEEE International Conference on
728 Systems, Man and Cybernetics (SMC), 2019, pp. 21–26. doi:[10.1109/SMC.2019.8913901](https://doi.org/10.1109/SMC.2019.8913901).

- 730 Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, M. C. Chan, Gee: A
731 gradient-based explainable variational autoencoder for network anomaly
732 detection, in: 2019 IEEE Conference on Communications and Network
733 Security (CNS), 2019, pp. 91–99. doi:10.1109/CNS.2019.8802833.
- 734 K. Amarasinghe, K. Kenney, M. Manic, Toward explainable deep neural
735 network based anomaly detection, in: 2018 11th International Conference
736 on Human System Interaction (HSI), 2018, pp. 311–317. doi:10.1109/HSI.
737 2018.8430788.
- 738 X. Zhang, J. Shi, X. Huang, F. Xiao, M. Yang, J. Huang,
739 X. Yin, A. Sohail Usmani, G. Chen, Towards deep probabilistic
740 graph neural network for natural gas leak detection and localiza-
741 tion without labeled anomaly data, Expert Systems with Appli-
742 cations 231 (2023) 120542. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423010448>. doi:<https://doi.org/10.1016/j.eswa.2023.120542>.
- 745 W.-T. Yang, M. S. Reis, V. Borodin, M. Juge, A. Roussy, An interpretable
746 unsupervised bayesian network model for fault detection and diagno-
747 sis, Control Engineering Practice 127 (2022) 105304. URL: <https://www.sciencedirect.com/science/article/pii/S0967066122001502>.
748 doi:<https://doi.org/10.1016/j.conengprac.2022.105304>.
- 750 L. C. Brito, G. A. Susto, J. N. Brito, M. A. V. Duarte, Fault diagno-
751 sis using explainable ai: A transfer learning-based approach for rotating
752 machinery exploiting augmented synthetic data, Expert Systems with
753 Applications 232 (2023) 120860. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423013623>. doi:<https://doi.org/10.1016/j.eswa.2023.120860>.
- 756 Z. Wu, X. Yang, X. Wei, P. Yuan, Y. Zhang, J. Bai, A self-supervised
757 anomaly detection algorithm with interpretability, Expert Systems with
758 Applications 237 (2024) 121539. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423020419>. doi:<https://doi.org/10.1016/j.eswa.2023.121539>.
- 761 M. Wadinger, M. Kvasnica, Real-time outlier detection with dynamic pro-
762 cess limits, in: Proceedings of the 2023 24th International Conference on
763 Process Control (PC), 2023. In press.

- 764 K. Yamanishi, J.-i. Takeuchi, A unifying framework for detecting outliers and
765 change points from non-stationary time series data, in: Proceedings of the
766 Eighth ACM SIGKDD International Conference on Knowledge Discovery
767 and Data Mining, KDD '02, Association for Computing Machinery, New
768 York, NY, USA, 2002, pp. 676–681. URL: <https://doi.org/10.1145/775047.775148>. doi:10.1145/775047.775148.
- 770 K. Yamanishi, J.-i. Takeuchi, G. Williams, P. Milne, On-line unsu-
771 pervised outlier detection using finite mixtures with discounting learn-
772 ing algorithms, *Data Mining and Knowledge Discovery* 8 (2004) 275–
773 300. URL: <https://doi.org/10.1023/B:DAMI.0000023676.72185.7c>.
774 doi:10.1023/B:DAMI.0000023676.72185.7c.
- 775 B. Steenwinckel, Adaptive anomaly detection and root cause analysis by fus-
776 ing semantics and machine learning, in: A. Gangemi, A. L. Gentile, A. G.
777 Nuzzolese, S. Rudolph, M. Maleshkova, H. Paulheim, J. Z. Pan, M. Alam
778 (Eds.), *The Semantic Web: ESWC 2018 Satellite Events*, Springer Inter-
779 national Publishing, Cham, 2018, pp. 272–282.
- 780 B. Steenwinckel, D. De Paepe, S. Vanden Hautte, P. Heyvaert, M. Ben-
781 tefrit, P. Moens, A. Dimou, B. Van Den Bossche, F. De Turck, S. Van
782 Hoecke, F. Ongena, Flags: A methodology for adaptive anomaly
783 detection and root cause analysis on sensor data streams by fusing
784 expert knowledge with machine learning, *Future Generation Com-
785 puter 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>.
- 788 M. Amer, M. Goldstein, S. Abdennadher, Enhancing one-class support vec-
789 tor machines for unsupervised anomaly detection, in: Proceedings of the
790 ACM SIGKDD Workshop on Outlier Detection and Description, ODD '13,
791 Association for Computing Machinery, New York, NY, USA, 2013, pp.
792 8–15. URL: <https://doi.org/10.1145/2500853.2500857>. doi:10.1145/2500853.2500857.
- 794 B. Liu, Y. Xiao, P. S. Yu, L. Cao, Y. Zhang, Z. Hao, Uncertain one-class
795 learning and concept summarization learning on uncertain data streams,
796 *IEEE Transactions on Knowledge and Data Engineering* 26 (2014) 468–
797 484. doi:10.1109/TKDE.2012.235.

- 798 B. Krawczyk, M. Woźniak, One-class classifiers with incremental
799 learning and forgetting for data streams with concept drift, Soft
800 Computing 19 (2015) 3387–3400. URL: <https://doi.org/10.1007/s00500-014-1492-5>. doi:10.1007/s00500-014-1492-5.
- 802 X. Miao, Y. Liu, H. Zhao, C. Li, Distributed online one-class support vec-
803 tor machine for anomaly detection over networks, IEEE Transactions on
804 Cybernetics 49 (2019) 1475–1488. doi:10.1109/TCYB.2018.2804940.
- 805 Ö. Gözüaçık, F. Can, Concept learning using one-class classifiers for
806 implicit drift detection in evolving data streams, Artificial Intelli-
807 gence Review 54 (2021) 3725–3747. URL: <https://doi.org/10.1007/s10462-020-09939-x>. doi:10.1007/s10462-020-09939-x.
- 809 R. Wetzig, A. Gulenko, F. Schmidt, Unsupervised anomaly alerting for
810 iot-gateway monitoring using adaptive thresholds and half-space trees,
811 in: 2019 Sixth International Conference on Internet of Things: Systems,
812 Management and Security (IOTSMS), 2019, pp. 161–168. doi:10.1109/IOTSMS48152.2019.8939201.
- 814 Y. Lyu, W. Li, Y. Wang, S. Sun, C. Wang, Rmhsforest: Rel-
815 ative mass and half-space tree based forest for anomaly detec-
816 tion, Chinese Journal of Electronics 29 (2020) 1093–1101. URL:
817 <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cje.2020.09.010>.
818 doi:<https://doi.org/10.1049/cje.2020.09.010>.
819 arXiv:<https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/cje.2020.09.010>
- 820 B. P. Welford, Note on a method for calculating corrected
821 sums of squares and products, Technometrics 4 (1962) 419–
822 420. URL: <https://www.tandfonline.com/doi/abs/10.1080/00401706.1962.10490022>.
823 doi:10.1080/00401706.1962.10490022.
824 arXiv:<https://www.tandfonline.com/doi/pdf/10.1080/00401706.1962.10490022>.
- 825 A. Genz, Numerical computation of multivariate normal probabilities, Jour-
826 nal of Computational and Graphical Statistics 1 (2000). doi:10.1080/
827 10618600.1992.10477010.
- 828 R. P. Brent, Algorithms for minimization without derivatives, Prentice-Hall,
829 Englewood Cliffs, N.J, 1972. URL: https://openlibrary.org/books/OL4739237M/Algorithms_for_minimization_without_derivatives.

- 831 F. Iglesias Vázquez, A. Hartl, T. Zseby, A. Zimek, Anomaly detection in
832 streaming data: A comparison and evaluation study, Expert Systems with
833 Applications 233 (2023) 120994. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423014963>. doi:<https://doi.org/10.1016/j.eswa.2023.120994>.
- 836 L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek,
837 M. Kloft, T. G. Dietterich, K.-R. Müller, A unifying review of deep and
838 shallow anomaly detection, Proceedings of the IEEE 109 (2021) 756–795.
839 doi:[10.1109/JPROC.2021.3052449](https://doi.org/10.1109/JPROC.2021.3052449).
- 840 I. D. Katser, V. O. Kozitsin, Skoltech anomaly benchmark (skab),
841 <https://www.kaggle.com/dsv/1693952>, 2020. doi:[10.34740/KAGGLE/DSV/1693952](https://doi.org/10.34740/KAGGLE/DSV/1693952).