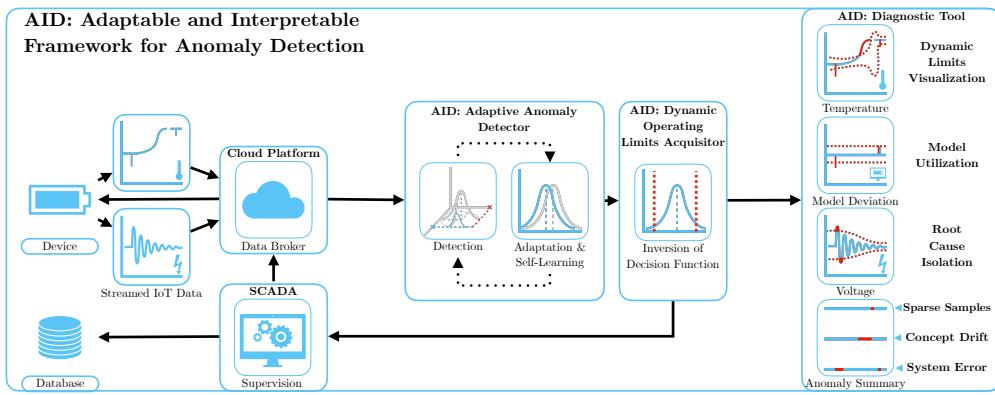


Graphical Abstract

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

Marek Wadinger, Michal Kvasnica



Highlights

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

Marek Wadinger, Michal Kvasnica

- Interpretable anomaly detector with self-supervised adaptation
- Demonstrates interpretability by providing dynamic operating limits
- Leverages self-learning approach on streamed IoT data
- Utilizes existing SCADA-based industrial infrastructure
- Offers faster response time to incidents due to root cause isolation

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

Marek Wadinger^{a,*}, Michal Kvasnica^a

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

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 at the level of individual inputs. 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 operating limits to integrate with existing alarm handling mechanisms in SCADA-based IoT systems. Two industrial-scale case studies demonstrate AID's capabilities. The first study showcases AID's effectiveness on energy storage system, adapting to changes, setting context-aware limits for SCADA, and ability to leverage a physics-based 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

*Phone numbers: +421 902 810 324 (Marek Wadinger)

Email addresses: marek.wadinger@stuba.sk (Marek Wadinger), michal.kvasnica@stuba.sk (Michal Kvasnica)

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 (2009) conducted an influential review of prior research efforts across diverse
⁸ application domains. Recent studies have underscored the need for holis-
⁹ tic and tunable anomaly detection methods accessible to operators (Laptev
¹⁰ 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 complexity issues, Zhang et al. (2024) have
¹⁵ successfully employed spatially distributed sensors and time-relative modu-
¹⁶ lation. Their approach has proven effective, particularly in the context of
¹⁷ complex non-linear systems, offering potential solutions to some of the chal-
¹⁸ lenges posed by IoT data. Huang et al., on the other hand, tackled the
¹⁹ problems of detecting global outliers, local outliers, and outlier clusters si-
²⁰ multaneously. Their proposed approach, based on density estimation, relies
²¹ on the notion that density distributions should exhibit minimal variations
²² in 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 is worth noting that feature engineering
²⁸ may introduce categorical variables and significantly increase the dimen-
²⁹ sionality of the data, requiring specific methods for handling large data, size-
³⁰ able data storage, and substantial computational resources (Talagala et al.,
³¹ 2021). Recently, Li et al. introduced an attribute-weighted outlier detection
³² algorithm, designed for high-dimensional datasets with mixtures of categor-
³³ ical and numerical data. Their approach assigns different weights to indi-
³⁴ vidual 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, immediate detection is not a feasible option unless there is a high tolerance for false alarms. **Promissing balance between early transition detection and low false alarm rate could be achieved by contrastive learning approach.** Deldari et al. (2021) have shown that by evaluating cosine similarity between predicted future representation and anticipated representation of time windows, it is possible to detect evolution in data with high accuracy.

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. (2012) showed the detector’s adaptation to data labeled on the flight. 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

72 second group uses statistical learning creating models explainable via prob-
73 ability. For instance, the integration of variational Bayesian inference prob-
74 abilistic graph neural network allowed Zhang et al. to model the posterior
75 distribution of sensor dependency for gas leakage localization on unlabeled
76 data (Zhang et al., 2023b). Yang et al. recently proposed a Bayesian net-
77 work (BN) for fault detection and diagnosis. In this BN, individual nodes
78 of the network represent normally distributed variables, whereas the multi-
79 ple regression model defines weights and relationships. Using the predefined
80 structure of the BN, the authors propose offline training with online detection
81 and diagnosis (Yang et al., 2022).

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

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

95 This paper emphasizes the importance of combining adaptability in in-
96 terpretable anomaly detection and proposes a method that addresses this
97 challenge in real industrial systems. Here we report the discovery and char-
98 acterization of an adaptive anomaly detection method for existing supervi-
99 sory control and data acquisition (SCADA) systems, employing streaming
100 IoT data. The ability to diagnose multivariate data while providing root
101 cause isolation via statistical learning, extends our previous contribution to
102 the field as presented in (Wadinger and Kvasnica, 2023). The proposed algo-
103 rithm aims to represent a general method that aids a range of existing safety-
104 critical systems where anomaly diagnosis and identification are paramount.
105 The schematic overview of the proposed method’s integration is presented in
106 Figure 1.

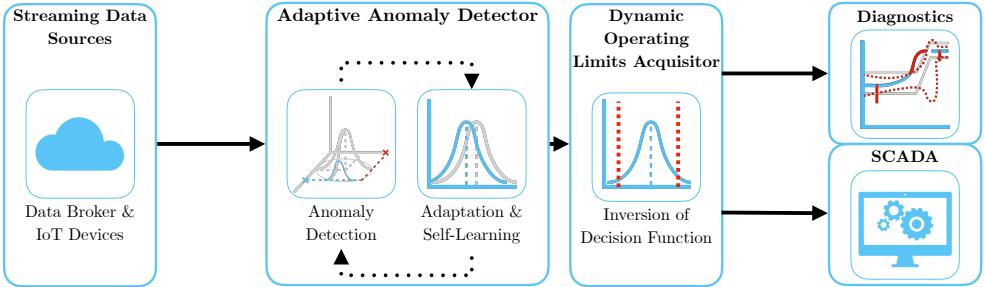


Figure 1: Schematic representation of the proposed method AID.

107 *1.2. Novelty of proposed approach*

108 The idea of using statistical outlier detection is well-established. We
109 highlight the impactful contributions of Yamanishi et al. in (Yamanishi and
110 Takeuchi, 2002; Yamanishi et al., 2004). The authors propose a method
111 for detecting anomalies in a time series. The method is based on the as-
112 sumption that the continuous data is generated by a mixture of Gaussian
113 distributions, while discrete data is modeled as histogram density. The au-
114 thors solve the problem of change point detection as well. However, the
115 adaptation system is unaware of such changes, making the moving window
116 the only source of adaptation. 117 [Online vectorized forecasting methods based](#)
118 [on well-established autoregression and moving averages have recently shown](#)
119 [the capability of adapting to non-stationarity in multivariate systems with-](#)
120 [out supervision Melnyk et al. \(2016\); Zhang et al. \(2023a\).](#) Their extension
121 [to diagnostic tasks is yet to be explored.](#) Our self-supervised approach facil-
122 itates intelligent adaptation concerning detected change points, to increase
123 the speed of adaptation where the probability of concept drift is high. By
124 leveraging its ability to adapt to changes in operational states, our proposed
125 method operates autonomously when such changes occur. Moreover, Yaman-
126 ishi et al. (2004) does not attempt to isolate the root cause of the anomaly.
127 Our approach extends statistical outlier detection by incorporating inter-
128 pretability. This is achieved by evaluating the inverse cumulative distribu-
129 tion function of the latest conditional probabilities for each measurement,
130 considering the remainder of the measurements, and establishing limits that
 define the threshold for normal event probabilities.

131 A limited number of studies have focused on adaptation and interpretabil-
132 ity within the framework of anomaly detection. Two recent contributions in

133 this area are made by Steenwinckel et al. as reported in (Steenwinckel, 2018;
134 Steenwinckel et al., 2021). In Steenwinckel (2018), the authors emphasize
135 the importance of combining prior knowledge with a data-driven approach
136 to achieve interpretability, particularly concerning root cause isolation. They
137 propose a novel approach that involves extracting features based on knowl-
138 edge graph pattern extraction and integrating them into the anomaly de-
139 tection mechanism. This graph is subsequently transformed into a matrix,
140 and adaptive region-of-interest extraction is performed using reinforcement
141 learning techniques. To enhance interpretability, a Generative Adversarial
142 Network (GAN) reconstructs a new graphical representation based on se-
143 lected vectors. However, it is important to note that the validation of this
144 idealized approach is pending further investigation. Lately, Steenwinckel
145 et al. (2021) introduced a comprehensive framework for adaptive anomaly
146 detection and root cause analysis in data streams. While the adaptation
147 process is driven by user feedback, the specific mechanism remains undis-
148 closed. The authors present an interpretation of their method through a user
149 dashboard, featuring visualizations of raw data. This dashboard is capable of
150 distinguishing between track-related problems and train-related issues, based
151 on whether multiple trains at the same geographical location approach the
152 anomaly. Meanwhile, our efforts are directed towards the development of a
153 self-supervised method that can learn autonomously, reducing the reliance
154 on human supervision, which is often constrained by time limitations and can
155 lead to significant delays in adaptation. Our method is distinguished by its
156 straightforward statistical reasoning and the ability to isolate the root cause
157 of anomalies. The interpretability of our method is demonstrated through
158 the establishment of dynamic operating limits for each signal, leveraging con-
159 ditional probabilities derived from the signal and other system measurements
160 and features. This provides operators with a clear understanding of the sys-
161 tem’s state and the underlying causes of anomalies and offers interoperability
162 with existing alarm handling mechanisms in SCADA which utilize operating
163 limits. To the best of our knowledge, this study appears to be one of the ini-
164 tial attempts to introduce a self-supervised approach for adaptive anomaly
165 detection and root cause isolation in SCADA-based systems utilizing IoT
166 data streams.

167 *1.3. Validation*

168 Two real-world industrial-scale case studies showcase that our proposed
169 method has the capacity to explain anomalies, isolate the root cause, and

allow adaptation to change points, allowing long-term deployment at the end users of energy storage systems. 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).

1.4. Practical 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 operating 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 operating limits, based on the current state of the system and its environment, with direct utilization in SCADA systems expecting minimal intervention to existing infrastructure. This allows the system to operate closer or further from its design limits, increasing its safety and profitability. The dynamic operating limits allow operational metrics monitoring, making potential early detection and prevention easier. Using general adaptable methods without interpretability, on the other hand, may pose safety risks and lower total financial benefits, as the triggered false alarms may need to be thoroughly analyzed, resulting in prolonged downtimes.

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

- Interpretable anomaly detector with self-supervised adaptation

- Demonstrates interpretability by providing dynamic operating limits
- Leverages self-learning approach on streamed IoT data
- Utilizes existing SCADA-based industrial infrastructure
- Offers faster response time to incidents due to root cause isolation

210 1.5. Paper Organization

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

217 2. Preliminaries

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

227 2.1. Welford's Online Algorithm

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

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

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

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

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

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

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

238 as it is less prone to numerical instability due to catastrophic cancellation,
239 significant loss of precision due to subtracting two nearly equal numbers.
240 Finally, the corresponding unbiased variance is

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

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

243 2.2. Inverting Welford's Algorithm

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

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

250 where the reverted mean is given as

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

251 Finally, the unbiased variance follows the formula:

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

252 Notably, inversion allows the algorithm to keep a constant rate of adap-
253 tation at the cost of storing a bounded data buffer.

254 2.3. Statistical Model of Multivariate System

255 Multivariate normal distribution generalizes the multivariate systems to
 256 the model where the degree to which variables are related is represented by
 257 the covariance matrix. Gaussian normal distribution of variables is a reason-
 258 able assumption for process measurements, as it is a common distribution
 259 that arises from stable physical processes measured with noise (Mishra and
 260 Datta-Gupta, 2018). The general notation of multivariate normal distribu-
 261 tion is:

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

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

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

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

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

269 The cumulative distribution function (CDF) of a multivariate Gaussian
 270 distribution describes the probability that all components of the random
 271 vector \mathbf{X} take on a value less than or equal to a particular point q in space,
 272 and can be used to evaluate the likelihood of observing a particular set of
 273 measurements or data points. In other words, it gives the probability of
 274 observing a random vector that falls within a certain region of space. The
 275 standard notation of CDF is as follows:

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

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

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

279 Given the PDF, we can also determine the value of \mathbf{x} that corresponds to a
 280 given quantile q using a numerical method for inversion of CDF (ICDF) often
 281 denoted as percent point function (PPF) or $F(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})^{-1}$. An algorithm that
 282 calculates the value of the PPF is part of standard statistical software tools.

283 2.4. Conditional Probability Distribution

284 Considering that we observe particular vector \mathbf{x}_i , we can update probability
 285 distributions, calculated according to the rules of conditional probability,
 286 of individual measurements within the vector given the rest of the measure-
 287 ments in \mathbf{x}_i . Let's assume multivariate normal distribution (8) and without
 288 loss of generality, that the vector \mathbf{x}_i can be partitioned into subset variable
 289 x_a , and complement vector \mathbf{x}_b as follows

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

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

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

292 and

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

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

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

296 where $\mu_{a|b}$ denotes the conditional mean and $\sigma_{a|b}^2$ represents the condi-
 297 tional variance. These crucial parameters can be computed by applying the
 298 Schur complement as follows:

$$\sigma_{a|b}^2 = \sigma_{aa}^2 - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}, \quad (15)$$

299 for the conditional variance $\sigma_{a|b}^2$, while the conditional mean, denoted as
 300 $\mu_{a|b}$, is determined by:

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

301 The conditional variance $\sigma_{a|b}^2$ essentially represents the Schur complement
 302 of $\boldsymbol{\Sigma}_{bb}$ within the overall covariance matrix $\boldsymbol{\Sigma}$.

303 2.5. Gaussian Anomaly Detection

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

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

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

$$q = P\{|x_a - \mu_{a|\mathbf{b}}| < 3\sigma_{a|\mathbf{b}}^2\} = 0.99735. \quad (17)$$

317 Utilizing a probabilistic model of normal behavior, we can determine
 318 threshold values x_l and x_u corresponding to the closed interval of the CDF
 319 where this probability is established. The inversion of Equation (10) facil-
 320 itates this calculation, yielding:

$$x_l = F((1 - P\{|x_a - \mu_{a|\mathbf{b}}| < 3\sigma_{a|\mathbf{b}}^2\}); \mu_{a|\mathbf{b}}, \sigma_{a|\mathbf{b}}^2)^{-1}, \quad (18)$$

321 for the lower limit, and

$$x_u = F((P\{|x_a - \mu_{a|\mathbf{b}}| < 3\sigma_{a|\mathbf{b}}^2\}); \mu_{a|\mathbf{b}}, \sigma_{a|\mathbf{b}}^2)^{-1}, \quad (19)$$

322 for the upper limit. These lower and upper limits together form vectors
 323 \mathbf{x}_l and \mathbf{x}_u , respectively, defining the region of normal system operation. This
 324 region is conceptualized as a hypercube in the feature space, with each di-
 325 mension bounded by the corresponding feature limits, as computed using
 326 Equations (18) and (19) for all $a = 1, \dots, k$; $\mathbf{b} = \{1, 2, \dots, k\}$ where $a \notin \mathbf{b}$.
 327 The approximation of a confidence ellipse as a hypercube can be employed
 328 to represent the region of normal system operation for individual variables
 329 of a multivariate system, rendering it as an aid for visual representation.

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

333 probabilistic density corresponds to the vector of conditional means $\mu_{a|\mathbf{b}}$ with
 334 respect to other features. The calculation of this distance involves the cumu-
 335 lative distribution function (CDF) of observations and conditional distribu-
 336 tions, as follows:

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

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

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

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

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

$$y_i = \begin{cases} 1 & \text{if } 1 \in \mathbf{y}_{s,i} \\ 0 & \text{otherwise,} \end{cases} \quad (22)$$

342 defining the discrimination boundary between system operation where
 343 $y_i = 0$ indicates normal system operation, and $y_i = 1$ indicates anomalous
 344 operation.

345 2.6. Anomaly Definition

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

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

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

358 Nevertheless, this conventional method may not be suitable for scenarios
 359 characterized by collective anomalies, where clusters of abnormal data points

360 coexist. A more pragmatic approach defines anomalies as deviations from
361 established "normal" patterns, resembling the principles of semi-supervised
362 learning. Change point detection, in a similar vein, can be regarded as a
363 relative approach that takes into account the varying dynamics of changes,
364 whether they occur gradually or abruptly (Iglesias Vázquez et al., 2023).

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

371 Notwithstanding the differences in terminology, methods employed for the
372 identification of data points residing in low-probability regions, irrespective of
373 whether they are referred to as "anomaly detection," "outlier detection," or
374 "novelty detection," share fundamental similarities (Iglesias Vázquez et al.,
375 2023).

376 For visual clarity, Figure 2 illustrates the differences between point anom-
377 lies, collective anomalies, and change points.

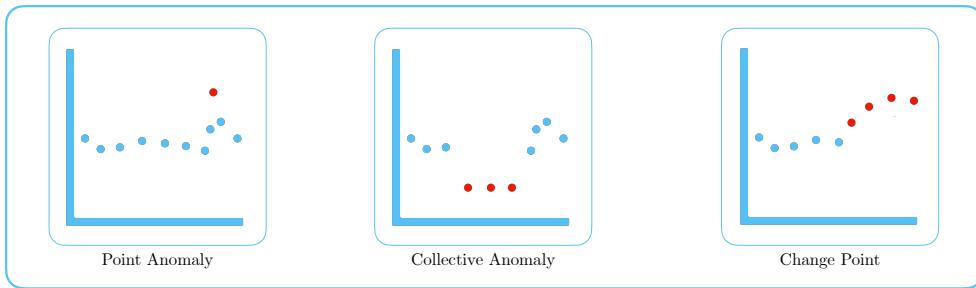


Figure 2: Illustration of sample scenarios of point anomaly (measurement with significant dissimilarity), collective anomaly (cluster of abnormal points), and change point (initial sequence of changed operation) detection.

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

379 In this section, we present an adaptive and interpretable detection frame-
380 work (AID) designed for SCADA-based industrial systems with streaming
381 IoT devices. Our approach is rooted in the foundational concepts discussed
382 in Preliminaries 2. We systematically leverage these theoretical building
383 blocks to introduce our method in a coherent manner.

384 Our approach begins by modeling the system as a dynamic multivariate
385 normal distribution, allowing it to effectively handle pervasive nonstationary
386 effects and interactions that impact industrial processes. We address several
387 critical factors, such as change points, concept drift, and seasonal effects.
388 Our primary contribution is the integration of an adaptable self-supervised
389 system with root cause identification and dynamic operating limits setting.
390 This unique combination empowers our online statistical model to diagnose
391 anomalies through three distinct mechanisms.

392 Firstly, we employ conditional probability calculations to assess the nor-
393 mality of the system’s operating conditions. This step ensures that our
394 method identifies outliers within individual signal measurements and inter-
395 prets the root causes of anomalies, facilitating faster and more precise diag-
396 noses. Secondly, we detect abrupt changes due to concept drift, serving for
397 faster adaptation to new operating conditions without human intervention.
398 Thirdly, we harness interpretability as a tool to establish dynamic operating
399 limits. These adaptive limits enable our framework to seamlessly integrate
400 with existing SCADA-based infrastructure, a substantial advantage over ex-
401 isting solutions.

402 We have structured the subsequent sections to delve into the details of our
403 proposed methodology by the logical flow of data. The upcoming subsection
404 will cover the anomaly detection mechanism, followed by sections on online
405 training and adaptation. The next subsection will describe dynamic operat-
406 ing limits setting, followed by diagnostic capabilities. Lastly, we describe how
407 those parts converge into a diagnostic tool. For a schematic representation of
408 our proposed method, with a highlighted subsection attribution, please refer
409 to Figure 3. For a concise technical representation of our proposed method,
410 please refer to Algorithm 1.

411 *3.1. Online detection*

412 In the online detection phase, AID distinguishes between normal and
413 anomalous observations based on the model of the system’s normal behavior.

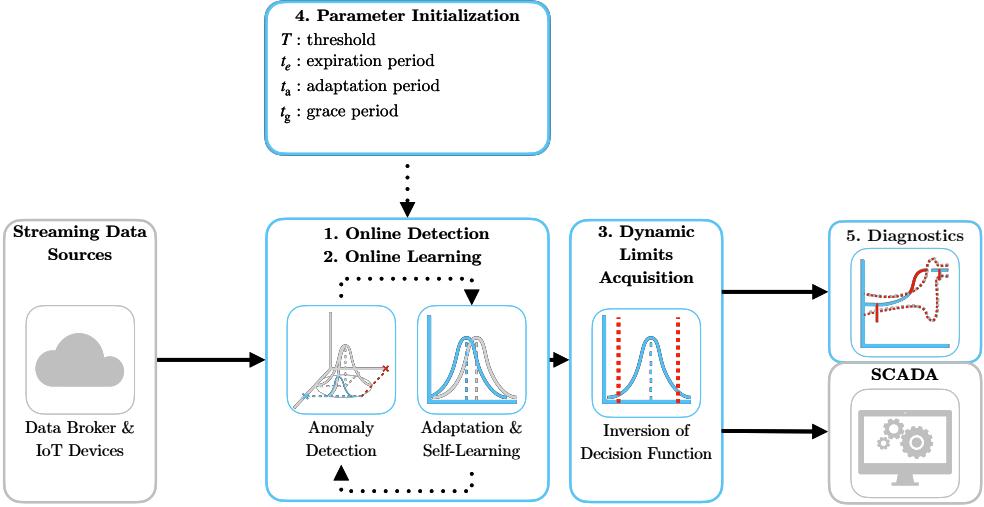


Figure 3: Schematic representation of the proposed method AID with parameter initialization. Colored boxes represent steps described within the subsection.

414 The detection pipeline is event-triggered upon the arrival of a new set of
 415 measurements.

416 To initiate the process, AID computes the properties of the conditional
 417 distribution based on the current observations given the dynamic joint nor-
 418 mal distribution. These calculations are performed for each element of the
 419 process observation vector \mathbf{x}_i at time instance i . Specifically, we calculate the
 420 conditional mean using (16) and the conditional variance using (15) for ele-
 421 ments of \mathbf{x}_i . These computations yield univariate conditional distributions
 422 for individual signals and features. These conditional distributions play a
 423 crucial role in assessing the abnormality of signals and features concerning
 424 their relationships with other elements of \mathbf{x}_i . Consequently, AID inherently
 425 considers the interactions between input signals and features.

426 The determination of anomalous behavior is influenced by the parame-
 427 ter T , which is a user-defined hyperparameter representing a probabilistic
 428 threshold that sets the boundary between normal and anomalous behavior.
 429 Details regarding the selection of an appropriate value for T are discussed
 430 in Subsection 3.5. Whenever an anomaly is detected within one of the sig-
 431 nals or features, it triggers an alert regarding the overall system's anomalous
 432 behavior, as described in (22). Nevertheless, individual determinations of

433 anomalies serve as a diagnostic tool for isolating the root causes of anomalies,
434 as further discussed in Subsection 3.4.

435 The proposed mechanism is applicable to both point anomalies and col-
436 lective anomalies. In the case of collective anomalies, their duration and
437 deviation may serve as precursors to concept drift in the system. To iden-
438 tify concept drift, we introduce a parameter adaptation period t_a . Given
439 the predicted system anomaly state from (22) as y_i over a window of past
440 observations $\mathbf{y}_i = \{y_{i-t_a}, \dots, y_i\}$ bounded by t_a , the following test determines
441 anticipated change points:

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

442 Here, $n(\mathbf{y}_i)$ denotes the dimensionality of \mathbf{y}_i . The logic behind (23) is
443 that over an adaptation period t_a , change points can be distinguished from
444 collective anomalies and point anomalies due to their minimum duration,
445 while T allows for some overlap with previous normal conditions.

446 Our framework anticipates unexpected novel behavior, including non-
447 uniformities in sampling. Assuming that the distribution of sampling times
448 remains stable over the long term, we can employ equivalent steps on the
449 observed time between samples to discriminate signal loss from long-term
450 anomalous network events.

451 3.2. Online learning

452 AID's training process follows an incremental self-learning approach, al-
453 lowing for online model updates as new samples arrive. Self-learning, in this
454 context, focuses on selecting only relevant data for training to maintain the
455 model's long-term relevancy and stability. This approach proves particularly
456 valuable in handling streaming data, where human supervision can intro-
457 duce significant computational delays, affecting response time in a sequential
458 setting.

459 In online anomaly detector training, regardless of the type of supervi-
460 sion, the learning is typically built upon observations of the normal state.
461 We introduce a grace period denoted as t_g to enable model calibration in
462 the initial stages after deployment. During this period, when normality in
463 samples is expected, the model learns from all observations. Subsequently,
464 self-supervised and unsupervised detectors are expected to make autonomous
465 decisions.

466 However, in the case of industrial systems, the drifts in the concept might
467 often render the normal state anomalous, slowing down or preventing adap-
468 tation completely. This is particularly true for the case of seasonal effects,
469 where the system is expected to operate in a different mode for a certain
470 period of time. To address this issue, AID’s adaptation incorporates two
471 self-supervised mechanisms.

472 Firstly, the model is updated if the observation at time instance i is
473 marked normal in the detection phase. In the case of a dynamic multivari-
474 ate probability distribution, the updated parameters are μ_i and Σ_i at time
475 instance i . Update of the mean vector μ_i and covariance matrix Σ_i is gov-
476 erned by Welford’s online algorithm using equation (2) and (4) respectively.
477 Samples beyond the expiration period t_e , discussed further in Subsection 3.5,
478 are disregarded during the second pass. The effect of expired samples is
479 reverted using inverse Welford’s algorithm for mean (6) and variance (7),
480 accessing the data in the bounded internal buffer. For more details, refer to
481 Subsection 2.2.

482 The second mechanism, which enables adaptation to anomalous samples,
483 relies on changepoint detection. This mechanism operates under the as-
484 sumption that detected changepoints represent new operational states with
485 limited overlap with the previous ones, as specified in Equation 23. It facili-
486 tates rapid adaptation to evolving data patterns without the need for human
487 intervention. The selection of the adaptation period t_a , as discussed further
488 in Subsection 3.5, is thus crucial for determining the speed of adaptation or
489 the potential mitigation of the second adaptation mechanism.

490 To anticipate potential deviations from sampling uniformity, we calculate
491 the cumulative distribution function (CDF) over the univariate normal dis-
492 tribution of sampling. We operate under the assumption that, over the long
493 term, the distribution of sampling times remains stable, employing a one-
494 pass update mechanism of (2) and (4), for efficiency. To proactively detect
495 subtle changes in sampling patterns, self-supervised learning is employed,
496 leveraging anomalies weighted by the deviation from $(1 - F(x_i; \mu, \sigma^2))$ for
497 training.

498 3.3. *Dynamic limits acquisition*

499 As we wrote in the subsection 1.4 Practical Impact, the monitoring mech-
500 anisms of SCADA readily depend on the upper and lower operating limits
501 of individual parameters of the system. In the case of industrial systems,

502 these limits are often defined by the sensor's designed limits and the sys-
503 tem dynamics. These limits are typically static and do not account for the
504 dynamically changing conditions. Our proposed method AID is capable of
505 setting dynamic operating limits, thus allowing integration into the existing
506 SCADA-based infrastructure.

507 The threshold T applied on the dynamic multivariate normal distribution
508 creates a confidence hyperellipse at T probability level. Such a hyperellipse
509 would not allow to effectively bound individual signals as it depends on val-
510 ues that other jointly distributed variables take. Nevertheless, by computing
511 the conditional for process observation vector \mathbf{x}_i at time instance i , we can
512 compute the conditional density function for individual signals. By applying
513 threshold T on individual conditional probabilities, we establish a hyper-
514 cube defined by lower and upper threshold values, denoted as \mathbf{x}_l and \mathbf{x}_u ,
515 respectively. These thresholds are derived from (18) and (19), incorporating
516 updated model parameters. Lower and upper thresholds play a pivotal role
517 as dynamic operating limits. They may be used as an addition to static op-
518 erating limits used by monitoring systems in SCADA, accounting for spatial
519 factors, such as multipoint measurements, temporal factors, such as aging,
520 and actual environmental conditions that influence sensor operation. More-
521 over, any violation of the limits is also detected as an anomaly.

522 3.4. Diagnostics

523 One of the crucial aspects of diagnostics is root cause isolation. Using the
524 ability to detect anomalies in individual signals and features, AID is capable
525 of isolating the root cause of anomalies with consideration of their mutual
526 relationships. This is achieved by computing the conditional probability of
527 individual signals and features given the rest of the process observation vec-
528 tor \mathbf{x}_i at time instance i . The dynamic process limits further enhance the
529 diagnosis by providing the context of the anomaly, including the extent of
530 deviation from normal operation and the direction of the deviation. The
531 proposed diagnostic mechanism is particularly useful in the case of collec-
532 tive anomalies, where the unified direction of deviations is expected. AID's
533 interpretability is an asset for domain experts to understand why certain
534 anomalies are flagged and enables operators to assess the system's state by
535 visualizing limits and deviations, thus detecting the speed at which the pro-
536 cess variable approaches the limits before an anomaly occurs.

537 3.5. Model Parameters Initialization

538 The model initialization is governed by defining two required hyperparameters of the model: the expiration period (t_e) and the threshold (T). The
539 expiration period determines the window size for time-rolling computations,
540 impacting the proportion of outliers within a given timeframe and directly in-
541 fluencing the relaxation (with a longer expiration period) or tightening (with
542 a shorter expiration period) of dynamic signal limits. Additionally, we intro-
543 duce a grace period t_g , which defaults to $uite$, allowing for model calibration.
544 During this grace period, system anomalies are not flagged to prevent false
545 positives and speed up self-supervised learning, introduced in Subsection 3.2.
546 t_g can take any value smaller than $uite$, if the detection must be delivered fast
547 after intergration. The length of the expiration period inversely correlates
548 with the model’s ability to adapt to sudden changes. The adaptation and
549 detection of significant drifts in the data-generating process, such as changes
550 in central tendency, is managed through the adaptation period t_a . A shorter
551 t_a results in faster adaptation to new operating conditions, while making the
552 system vulnerable to prolonged collective anomalies. A longer t_a results in
553 slower adaptation to significantly deviating new operations, but allows longer
554 alerts regarding collective anomalies. In most cases, $t_a = 1/4t_e$ offers optimal
555 performance.

557 As a general rule of thumb, expiration period t_e should be determined
558 based on the slowest observed dynamics within the multivariate system. The
559 threshold T defaults to the three-sigma probability of q in (17). Adjusting
560 this threshold can fine-tune the trade-off between precision and recall. A
561 lower threshold boosts recall but may lower precision, while a higher thresh-
562 old enhances precision at the cost of recall. We recommend starting with
563 the default values of other parameters and making adjustments based on
564 real-time model performance, as the model’s interpretability can reduce the
565 time and effort required for fine-tuning. The presence of one non-default
566 interpretable hyperparameter facilitates quick adaptation of AID in a broad
567 range of use cases.

Algorithm 1 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: $i \leftarrow 1; n \leftarrow 1; T \leftarrow (17); \boldsymbol{\mu} \leftarrow \mathbf{x}_0; \boldsymbol{\Sigma} \leftarrow \mathbf{1}_{k \times k}; \mu_t \leftarrow 0; \sigma_t^2 \leftarrow 1;$
 - 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 (21);
 - 6: $y_i \leftarrow \text{PREDICT}(\mathbf{y}_{s,i})$ using (22);
 - 7: $\mathbf{x}_{l,i}, \mathbf{x}_{u,i} \leftarrow \text{GET}(T, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ using (18), (19);
 - 8: $y_{t,i} \leftarrow \text{PREDICT}(t_i - t_{i-1})$ using (21);
 - 9: $\mu_t, \sigma_t^2 \leftarrow \text{UPDATE}(t_i - t_{i-1}, \mu_t, \sigma_t^2)$ using (2), (4);
 - 10: **if** (22) = 0 **or** (23) **then**
 - 11: $\boldsymbol{\mu}, \boldsymbol{\Sigma} \leftarrow \text{UPDATE}(\mathbf{x}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}, n)$ using (2), (4);
 - 12: **if** (23) **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**
-

568 **4. Case Study**

569 This section presents two case studies on real industrial-scale energy stor-
570 ages and a real data benchmark to demonstrate the effectiveness and appli-
571 cability of our proposed approach. We investigate the properties and perfor-
572 mance of the approach using signals from IoT devices in an energy system
573 and streamed benchmark system data. The successful deployment demon-
574 strates that this approach is suitable for existing industrial systems utilizing
575 IoT data streams on top of well-established SCADA systems.

576 *4.1. Battery Energy Storage System TERRA*

577 In the first case study, we demonstrate our proposed method on real
578 industrial-scale battery energy storage system (BESS) TERRA, depicted in
579 Fig 4. TERRA has an installed capacity of 151 kWh distributed among 10
580 modules with 20 Li-ion NMC cells. The Inverter’s nominal power is 100 kW.
581 The TERRA reports measurements of State of Charge (SoC), supply/draw
582 energy set-points, and inner temperature, at 6 positions (channels) of each
583 battery module. A substantial size of the system, which is 2.4x2.4x1.2m
584 (HxWxD), requires a proper cooling mechanism. The cooling is handled by
585 forced air from the HVAC system and inner fans, while the fire safety system
586 is passive. Tight battery temperature control is needed to optimize perfor-
587 mance and maximize the safety and battery’s lifespan. Identifying anomalous
588 events and removal of corrupted data might yield significant improvement in
589 the process control level and increase the reliability and stability of the sys-
590 tem.

591 The AID is integrated into the existing software infrastructure of the
592 system, allowing detection and diagnosis of the system using streamed IoT
593 data. Here we replay a 9-day stream of historical measurements of the device,
594 to demonstrate key features of AID.

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



Figure 4: Photograph of the actual studied TERRA energy storage unit with open doors (left), and closed doors (right).

603 Figure 5 depicts the average cell temperature measurement of the TERRA
 604 for all 10 modules. The data are normalized to the range $[0, 1]$ to protect
 605 the sensitive business value. The light red area represents the region out of
 606 dynamic operating limits as provided by AID. On 7th March 2022, the system
 607 was relocated from the inside of the building to the outside power socket. The
 608 system was expected to adapt to the new behavior within 4 days as specified
 609 by t_e . Nevertheless, due to the protection of the model from learning the
 610 anomalous data, the new behavior could not be captured as the system was
 611 not operating within the safe limits. The adaptation started three days later,
 612 as only some of the measurements within the safe region after transfer were
 613 learned. Therefore, the importance of self-supervised adaptation to changes
 614 in data is crucial. As we can see, the change points detection according to
 615 (23) alerted such change shortly after the TERRA was connected to a data
 616 broker, while the length of the adaptation period enabled discrimination from
 617 collective anomaly.

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

623 The default sampling rate of the incoming signal measurements is 1
 624 minute. However, network communication of the IoT devices is prone to

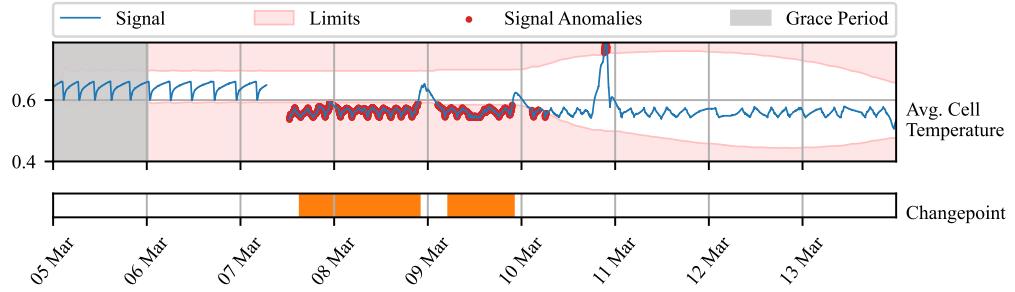


Figure 5: Detection of anomalies using model **without** adaptation to change points in normalized average cell temperature of TERRA observed over nine days (blue line). The model alerts anomalies (red dots) for approximately three days. The dynamic operating limits (light red area), given by the model without adaptation to change points, are stagnant during the period of detected change point (orange bars), which is triggered t_a hours after anomaly is alerted. The adaptation of the model to novel behavior starts as the t_e is approached.

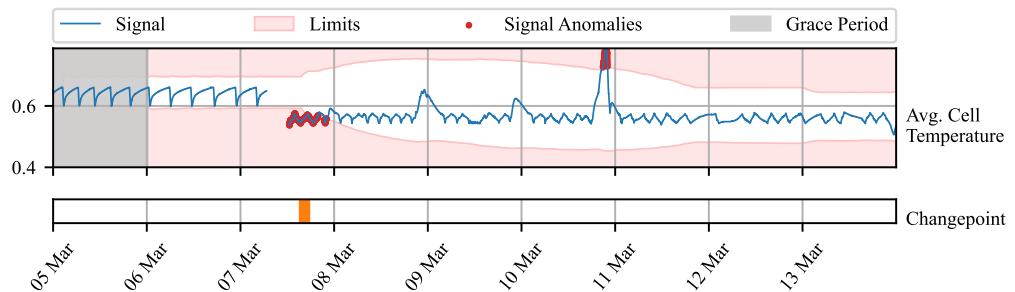


Figure 6: Detection of anomalies using model **with** adaptation to change points over the same historical measurement as in Figure 5. The change point is detected t_a hours after the anomaly is alerted, triggering adaptation to changed behavior more than two days sooner, compared to model **without** adaptation to change points in Figure 5. The adaptation is reflected in changes in dynamic operating limits. The system alerts anomalies for approximately 10 hours.

625 packet dropout, which results in unexpected non-uniformities in sampling
 626 from the perspective of the SCADA system. The transfer of TERRA was
 627 accompanied by the disconnection of IoT sensors from the data broker which
 628 might be considered an anomaly. The system can detect such anomalies as
 629 well, as depicted in Figure 7. Along with known disconnection, the system
 630 alerted two more non-uniformities of shorter extend, scaled in the figure for
 631 better visibility. The short loss of signal was caused by the packet drop, as
 632 it impacted only a few consecutive measurements. Various confidence levels
 633 could be used to further analyze and map potential causes to the duration
 634 of the outage.

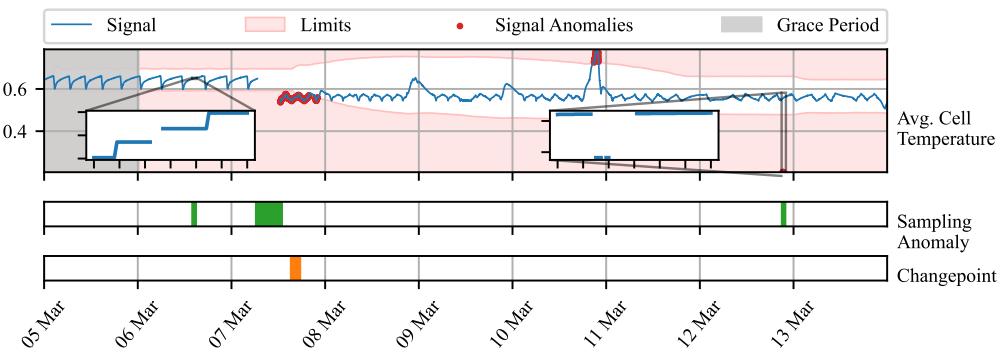


Figure 7: Detection of anomalies using model with adaptation to change points from Figure 6 with added depiction of accompanying task of sampling anomaly detection (green bars). Zoomed areas focus on short events of abnormal sampling detected by AID. The second zoomed area also highlights faulty measurement, which the system marked as a point anomaly.

635 Lastly, we want to acknowledge the outlier, left uncaptured due to in-
 636 creased variance of the distribution in a period of adaptation. Observing
 637 multiple variables, where some might be influenced less by the change in be-
 638 havior, might be beneficial in such cases. The industrial partner provided a
 639 physics-based model of the battery module temperature, defined as follows:

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

640 When combined with an averaged measurement of battery module tem-
 641 perature, we could compute the difference between real and predicted tem-

642 perature. Such deviation can be useful in detecting unexpected patterns in
643 temperature due to the impact of external disturbance and aging. Neverthe-
644 less, it may be inaccurate as the physics-based model is simplified and does
645 not account for spatial aspects, like temperature gradients as well as different
646 dynamic effects of charging and discharging on temperature. For instance,
647 in Fig. 4 during the first two days we see, that the cooling dynamic is not
648 captured well, resulting in a subtle positive difference between average cell
649 temperature and the temperature predicted by the model. In combination
650 with the raw measured average of the temperature, the AID captures the
651 outlier on 9th March which could not be captured in a univariate setting.
652 The physics-based model exposes temporal aspects of the behavior as it con-
653 siders the dynamics of its inputs. The rapid increase in temperature w.r.t
654 the modeled dynamics due to environmental conditions will draw a sharp
655 positive peak in the difference between the real and predicted temperature,
656 which will slowly vanish. Based on the significance of the deviation, the peak
657 will be notified as a single-point anomaly or collective anomaly.

658 This case study demonstrated AID’s effectiveness within the context of
659 the energy storage system, specifically the TERRA system. The AID system
660 exhibited adaptability to changes in the operational environment, contribut-
661 ing to its versatility and robustness. Additionally, it facilitated the establish-
662 ment of dynamic operating limits for SCADA systems, considering context
663 of the device such as environmental conditions or aging. Furthermore, the
664 AID system showcased its capability to operate with a physics-based model,
665 enhancing the precision of anomaly detection processes. This highlights the
666 potential of AID as a valuable tool within complex industrial systems. The
667 validity of our proposed approach was verified by our industrial partner, who
668 confirmed that the detected anomalies were indeed caused by the aforemen-
669 tioned events.

670 *4.2. Kokam Battery Module*

671 A second case study presents temperature profile monitoring of individual
672 modules of battery pack TERRA deployed at the premises of the end user.
673 During the operation, a hardware fault of module’s 9 cooling fan occurred on
674 23rd August 2023 at 17:12:30. Our industrial partner was interested in find-
675 ing out, whether such an event could be captured by an anomaly detection
676 system. Each of the 10 modules, embodies 20 cells measured by 6 spatially
677 distributed sensors as shown in Figure 9. The measurements are sent in 30-
678 second intervals and processed in a streamed manner by SCADA. With the

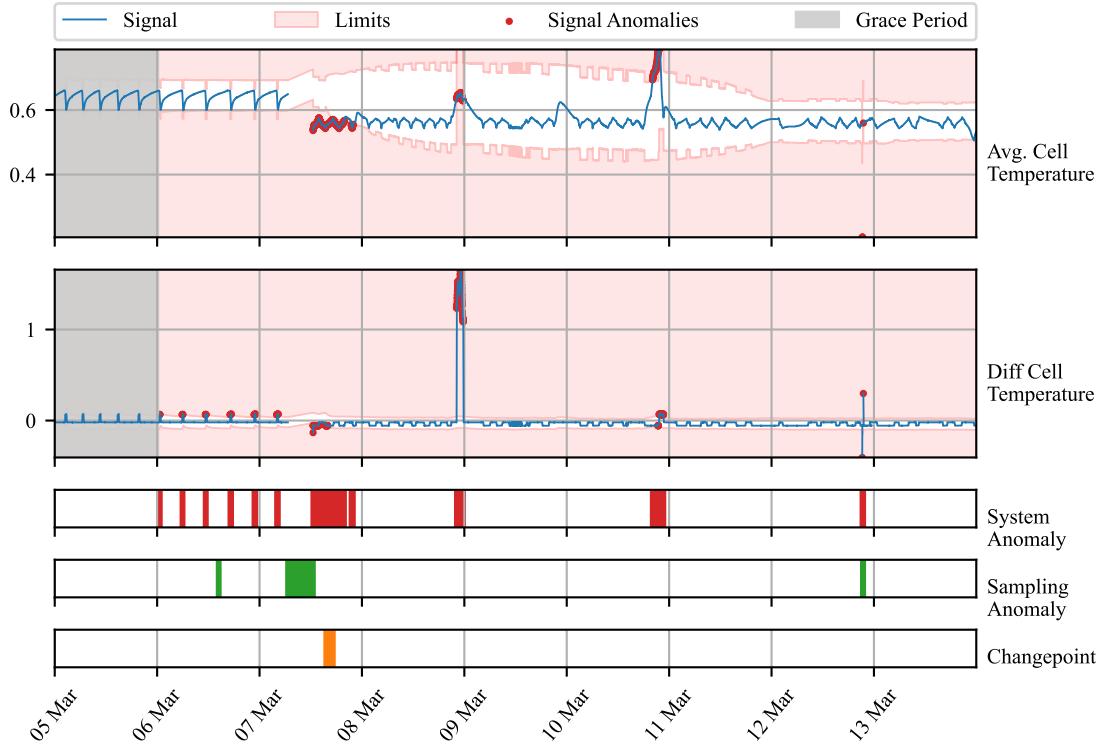


Figure 8: Time series of TERRA measurements observed over 9 days (blue line). The y-axis renders the average temperature of all cells and modules after the normalization to the range of [0, 1]. The light red area represents an area out of dynamic operating limits for individual signals. Observations 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.

679 availability of the temperature profiles for all the modules, we computed the
 680 deviation of the observed value from the average of all the modules' temper-
 681 ature measurements. The ground truth information about the fan fault was
 682 provided to the best of the operator's knowledge. However, this information
 683 serves for evaluation only, as the system operates in a self-supervised manner.

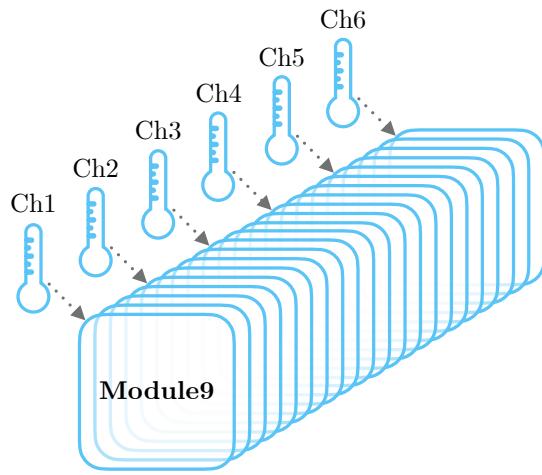


Figure 9: Module 9 with 20 cells and 6 sensors measuring the temperature at each 4th cell.

684 Our anomaly detection system was, in this case, initialized for the op-
 685 eration in production. The expiration period of 7 days, allowed the system
 686 to adapt to weekly seasonality due to the usage of the battery following
 687 work week. The grace period was kept at the default value, equal to t_e . The
 688 threshold value was shifted to a 4 sigma value of 99.977% which makes the the
 689 frequency of anomalous events approximately once a week given 30-second
 690 sampling. The adaptation period was held constant as the deployed system
 691 is not expected to change its behavior dramatically on a daily basis.

692 In Figure 10 we observe 4 days of operation around the period of fan
 693 fault occurrence. The deviations between the observed temperature mea-
 694 sured by channels of module 9 and the average temperature of all modules
 695 are displayed. The dynamic operating limits tightly envelop temperatures
 696 measured by the sensors in the middle of the module (refer to Figure 9),
 697 while measurements at both sides deviate more due to the proximity to the
 698 walls and sources of disturbance. We observed multiple alarms raised by var-

699 ious channels individually before the fan fault. These anomalies, while not
700 addressed here further, could be subjects of interest for further investigation
701 by system operators. Meanwhile, the fan fault at the center of our focus is
702 alarmed based on three measurements, namely channels 1, 2, and 3. From
703 the zoomed views, we can observe a sharp increase in the temperature devia-
704 tion. The alarm is on until 24th August at noon, when significant fluctuations
705 vanish followed by temporary settling of the temperature. On 25th August
706 at 11:21, increased temperature fluctuations are followed by an increase of
707 temperature similar to the initial one. AID alerts this fault again based on
708 measurements by channels 1, 2, and 3.

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

717 Interestingly, during the presence of a fault in the fan, two more peri-
718 ods where the fan started operating again followed as depicted in Figure 11.
719 Periods of operation were interrupted again on 27th and 28th August respec-
720 tively in the early morning hours. In both of the cases, AID detected the
721 presence of the fault at the moment of occurrence. In the first case, channel
722 3 reported an anomaly slightly before the increase in temperature, due to
723 abnormal fluctuation happening prior to faults.

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

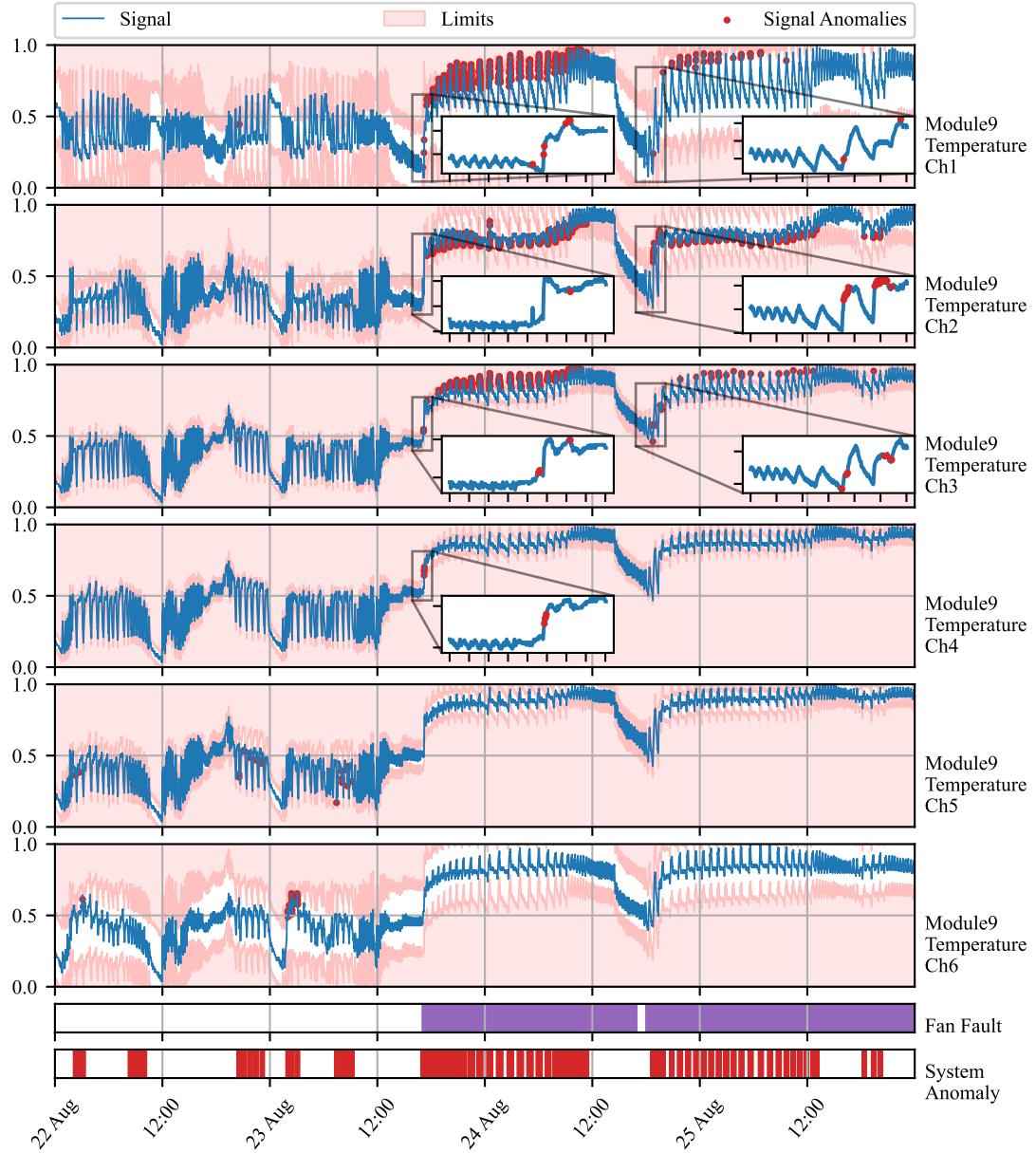


Figure 10: Time series of battery module 9 measurements (blue line). The y-axis renders the normalized deviations of temperature from the average of all modules. Signal anomalies are marked as red dots. The light red area represents an area out of dynamic operating limits. True fan faults are marked by purple bars. Green bars represent the times of changepoint detection. Red bars denote the period where any signal anomaly.

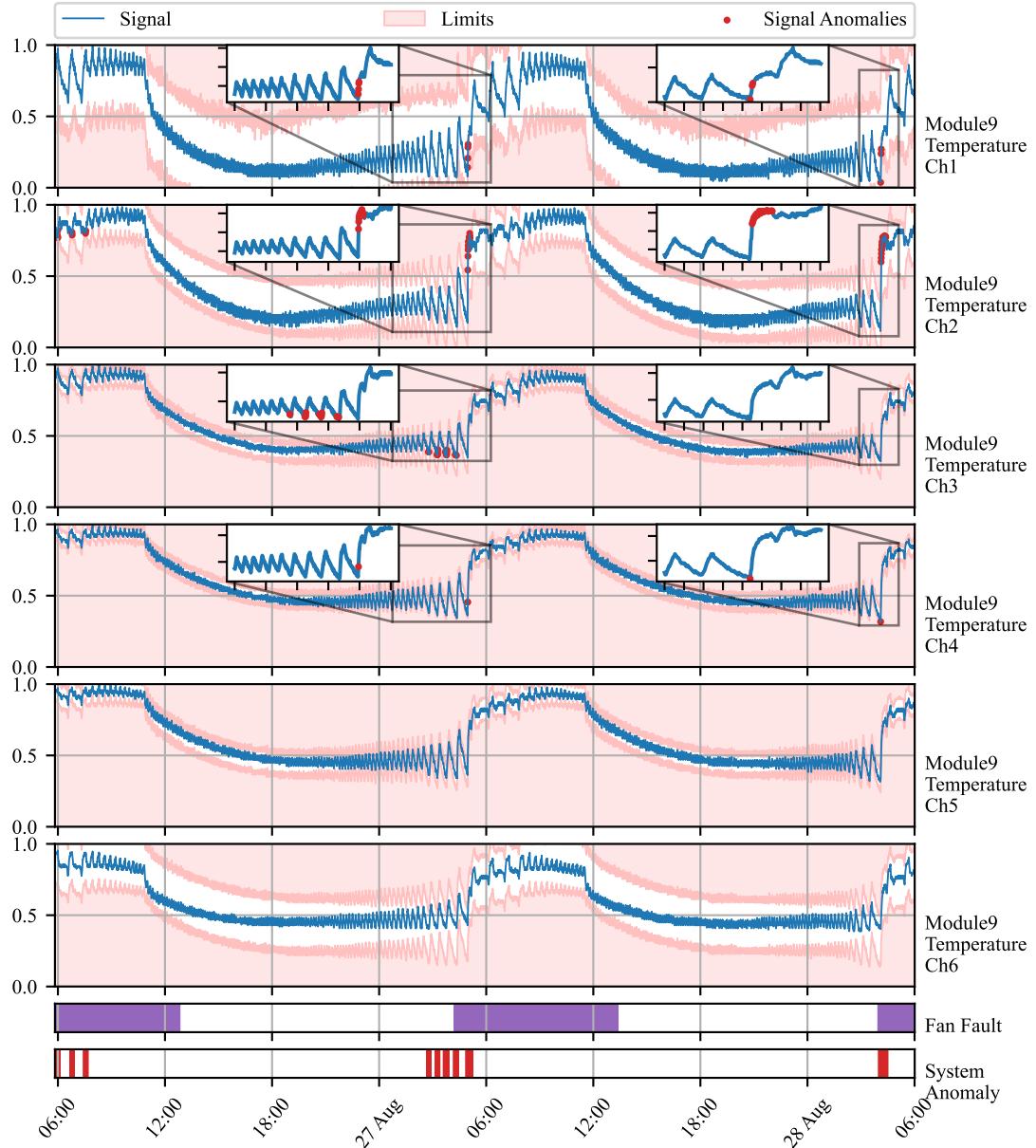


Figure 11: Time series of battery module 9 measurements (blue line). The y-axis renders the normalized deviations of temperature from an average of all modules. Signal anomalies are marked as red dots. The light red area represents an area out of dynamic operating limits. True fan faults are marked by purple bars. Green bars represent the times of changepoint detection. Red bars denote the period where any of the signals contained anomaly.

737 *4.3. Real Data Benchmark*

738 The benchmarking comparison in this subsection evaluates the AID frame-
739 work against adaptive unsupervised detection methods, specifically One-
740 Class Support Vector Machine (OC-SVM) and Half-Space Trees (HS-Trees).
741 These methods are widely recognized for their iterative learning capabilities
742 on multivariate time-series data, making them suitable for anomaly detection
743 in dynamic systems, as previously discussed in the Introduction 1.3.

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

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

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

770 As adaptation is required and anticipated within benchmark datasets,
771 the performance is evaluated iteratively, similarly to the operation after de-
772 ployment. The metric is updated with each new sample and its final value is
773 used to drive Bayesian Optimization. The performance is evaluated using the
774 best-performing model, found by Bayesian Optimization. The performance

775 of the proposed method is evaluated on the same data as the models are
776 optimized for.

777 Hyperparameter search ranges are specified, with values centered around
778 default library values for OC-SVM and HS-Trees. The ranges are inten-
779 tionally set wide to facilitate comprehensive exploration. The quantile filter
780 threshold used in OC-SVM and HS-Trees aligns with the threshold used in
781 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)
	t_g	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)

782 The results for models optimized for the F1 score are summarized in Ta-
783 ble 2, which includes precision, recall, F1 score, and average latency. Macro
784 values are enclosed in brackets, representing the mean of the metric for both
785 anomalies and normal data. A perfect detection achieves 100% in each met-
786 ric **except for the false positive rate (FAR), where a perfect detection attains**
787 **0%**. According to the Scoreboard for various algorithms on SKAB’s Kaggle
788 page, all iterative approaches perform comparably to the batch-trained iso-
789 lation forest and autoencoder, validating the optimization process. Notably,
790 the proposed AID method outperforms both reference methods in terms of
791 **precision, recall, F1 score, area under curve, and false positive rate**, despite
792 having a 30-fold higher latency per sample. This highlights the scalability
793 as a candidate for further development. Nevertheless, in this case, sampling
794 of the benchmark data still offers enough time to deliver predictions with
795 sufficient frequency. Scalability analysis to number of features is presented
796 in Subsection 4.4.

797 Optimal hyperparameters found during Bayesian Optimization are de-
798 tailed in Table 3. None of the parameters are at the edge of the provided

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	AID	HS-Trees	OC-SVM
Precision [%]	41 (59)	36 (51)	39 (54)
Recall [%]	80 (59)	74 (51)	63 (54)
F1 [%]	54 (53)	48 (44)	48 (52)
AUC [%]	59	51	54
Mean Rolling AUC [%]	57	50	53
FPR [%]	47	56	48
Avg. Latency [ms]	1.45	0.05	0.05

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.

Table 3: Optimal hyperparameters of methods optimized for F1 score

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

4.4. Scalability Analysis

We evaluate the scalability of the proposed method using temperature data from 10 battery modules with six temperature measurement points in the TERRA system. The data, sampled at 30-second intervals, are streamed to the AID system, which is initialized with parameters identical to those

807 in Subsection 4.2. Processing the data in a streamed manner simulates a
 808 real production environment. Latency is measured as the time between the
 809 sample’s arrival and the prediction’s delivery, including model updates. This
 810 evaluation occurs in a containerized environment with a single core and 8 GB
 811 RAM. Latency measurement spans 20160 samples from 2023-08-21 to 2023-
 812 08-27, excluding all but one measurement made during the grace period of 1
 813 day. We analyze latency for both the detection task alone and the combined
 814 task of detection and establishing dynamic process limits. Table 4 presents
 815 statistical indicators of the results, while the accompanying violin plots in
 816 the Figure offer visual insights into latency distribution for varying numbers
 817 of features. The significantly smaller minimum latency is attributed to eval-
 818 uation during the grace period, where fewer computations are performed.
 819 The significantly higher maximum latency could be attributed to reverting
 820 the effect of multiple points after signal loss in time-series data occurs.

Table 4: Latency analysis of the proposed method AID with varying number of features.

Number of Features	Detection $\mu \pm \sigma$ (min, max) [ms]	Detection + Limits $\mu \pm \sigma$ (min, max) [ms]
1	0.37 ± 0.26 (0.05, 31.7)	0.63 ± 0.38 (0.23, 35.9)
10	2.25 ± 0.92 (0.10, 13.6)	5.24 ± 0.98 (0.80, 15.1)
20	5.46 ± 2.16 (0.26, 30.6)	14.7 ± 2.27 (1.10, 47.5)
30	10.9 ± 4.31 (0.52, 42.4)	34.3 ± 4.50 (2.59, 72.4)
40	20.7 ± 8.15 (0.89, 52.7)	69.5 ± 8.57 (2.84, 140)
50	97.3 ± 47.4 (1.36, 1010)	297 ± 59.4 (3.94, 1330)
60	142 ± 71.2 (1.95, 1640)	468 ± 111 (7.08, 3710)

821 5. Conclusion

822 In this paper, we demonstrate the capacity of adaptive conditional prob-
 823 ability distribution to model the normal operation of dynamic systems em-
 824 ploying streaming IoT data and isolate the root cause of anomalies. AID
 825 dynamically adapts to non-stationarity by updating multivariate Gaussian
 826 distribution parameters over time. Additionally, self-supervision enhances
 827 the model by protecting it from the effects of outliers and increasing the

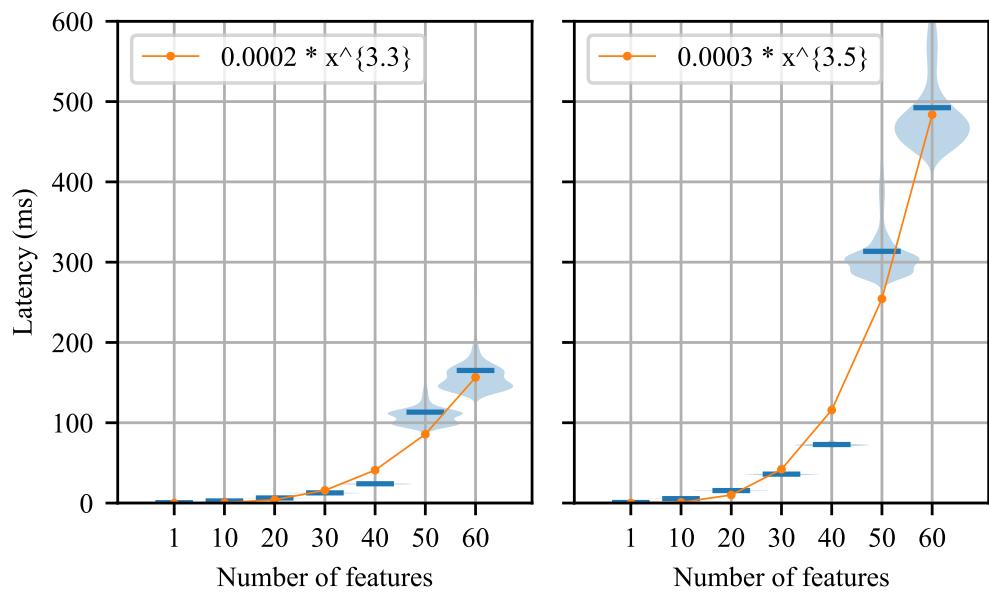


Figure 12: Analysis of latency distribution of the proposed method AID. Violin plots depict the distribution of the latency for varying number of features, while horizontal bars show mean latency.

828 speed of adaptation in response to autonomously detected changes in oper-
829 ation.

830 Our statistical model isolates the root causes of anomalies as extreme
831 deviations from the conditional means vector, considering spatial and tem-
832 poral effects encoded in features, as demonstrated in our case studies. This
833 approach establishes the system’s operational state by analyzing the dis-
834 tribution of signal measurements, computing the distance from the mean
835 of conditional probability, and setting dynamic operating limits based on
836 multivariate distribution parameters. Additionally, the detector alerts for
837 non-uniform sampling due to packet drops and sensor malfunctions. These
838 adaptable limits can be seamlessly integrated into SCADA architecture, en-
839 hancing context awareness and enabling plug-and-play compatibility with
840 existing infrastructure.

841 The ability to detect and identify anomalies in the system, isolate the
842 root cause of anomaly to specific signal or feature, and identify signal losses
843 is shown in two case studies on data from operated industrial-scale energy
844 storages. These case studies highlight the model’s ability to adapt, diagnose
845 the root cause of anomalies, and leverage both physics-based models and
846 spatially distributed sensors. Unlike many anomaly detection approaches,
847 the proposed AID method does not require historical data or ground truth
848 information about anomalies, alleviating the general limitations of detection
849 methods employed in the energy industry.

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

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

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

866 models, in contrast to Gaussian distribution.

867 **Additional information**

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

870 **CRediT authorship contribution statement**

871 **Marek Wadinger:** Conceptualization; Data curation; Formal analysis;
872 Investigation; Methodology; Resources; Software; Validation; Visualization;
873 Writing - original draft; and Writing - review & editing. **Michal Kvasnica:**
874 Conceptualization; Funding acquisition; Project administration; Resources;
875 Supervision; Validation.

876 **Declaration of Competing Interest**

877 The authors declare that they have no known competing financial inter-
878 ests or personal relationships that could have appeared to influence the work
879 reported in this paper.

880 **Acknowledgements**

881 This work was supported by the Horizon Europe [101079342]; the Slovak
882 Research and Development Agency [APVV-20-0261]; and the Scientific Grant
883 Agency of the Slovak Republic [1/0490/23].

884 **References**

- 885 Ahmad, S., Lavin, A., Purdy, S., Agha, Z., 2017. Unsuper-
886 vised real-time anomaly detection for streaming data. Neuro-
887 computing 262, 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
888 Data Streams.
889
890 Amarasinghe, K., Kenney, K., Manic, M., 2018. Toward explainable deep
891 neural network based anomaly detection, in: 2018 11th International Con-
892 ference on Human System Interaction (HSI), pp. 311–317. doi:[10.1109/HSI.2018.8430788](https://doi.org/10.1109/HSI.2018.8430788).

- 895 Amer, M., Goldstein, M., Abdennadher, S., 2013. Enhancing one-class sup-
896 port vector machines for unsupervised anomaly detection, in: Proceedings
897 of the ACM SIGKDD Workshop on Outlier Detection and Description,
898 Association for Computing Machinery, New York, NY, USA. pp.
899 8–15. URL: <https://doi.org/10.1145/2500853.2500857>, doi:10.1145/
900 2500853.2500857.
- 901 Barbosa Roa, N., Travé-Massuyès, L., Grisales-Palacio, V.H., 2019. Dy-
902 clee: Dynamic clustering for tracking evolving environments. Pat-
903 tern Recognition 94, 162–186. URL: <https://www.sciencedirect.com/science/article/pii/S0031320319301992>, doi:<https://doi.org/10.1016/j.patcog.2019.05.024>.
- 906 Bosman, H.H., Iacca, G., Tejada, A., Wörtche, H.J., Liotta, A., 2015. En-
907 sembles of incremental learners to detect anomalies in ad hoc sensor net-
908 works. Ad Hoc Networks 35, 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
911 Sensing, Processing and Networking Technologies.
- 912 Brito, L.C., Susto, G.A., Brito, J.N., Duarte, M.A.V., 2023. Fault diag-
913 nosis using explainable ai: A transfer learning-based approach for ro-
914 tating machinery exploiting augmented synthetic data. Expert Systems
915 with Applications 232, 120860. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423013623>, doi:<https://doi.org/10.1016/j.eswa.2023.120860>.
- 918 Carletti, M., Masiero, C., Beghi, A., Susto, G.A., 2019. Explainable machine
919 learning in industry 4.0: Evaluating feature importance in anomaly detec-
920 tion to enable root cause analysis, in: 2019 IEEE International Conference
921 on Systems, Man and Cybernetics (SMC), pp. 21–26. doi:10.1109/SMC.
922 2019.8913901.
- 923 Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: A sur-
924 vey. ACM Comput. Surv. 41. URL: <https://doi.org/10.1145/1541880.1541882>, doi:10.1145/1541880.1541882.
- 926 Cook, A.A., Misirlı, G., Fan, Z., 2020. Anomaly detection for iot time-
927 series data: A survey. IEEE Internet of Things Journal 7, 6481–6494.
928 doi:10.1109/JIOT.2019.2958185.

- 929 Deldari, S., Smith, D.V., Xue, H., Salim, F.D., 2021. Time series change point
930 detection with self-supervised contrastive predictive coding, in: Proceedings
931 of the Web Conference 2021, Association for Computing Machinery,
932 New York, NY, USA. pp. 3124–3135. URL: <https://doi.org/10.1145/3442381.3449903>, doi:10.1145/3442381.3449903.
- 934 Du, X., Chen, J., Yu, J., Li, S., Tan, Q., 2024. Generative adversarial nets
935 for unsupervised outlier detection. Expert Systems with Applications 236,
936 121161. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423016639>, doi:<https://doi.org/10.1016/j.eswa.2023.121161>.
- 939 Fan, C., Sun, Y., Zhao, Y., Song, M., Wang, J., 2019. Deep learning-
940 based feature engineering methods for improved building energy predic-
941 tion. Applied Energy 240, 35–45. URL: <https://www.sciencedirect.com/science/article/pii/S0306261919303496>, doi:<https://doi.org/10.1016/j.apenergy.2019.02.052>.
- 944 Genz, A., 2000. Numerical computation of multivariate normal probabili-
945 ties. Journal of Computational and Graphical Statistics 1. doi:10.1080/
946 10618600.1992.10477010.
- 947 Gözüaçık, Ö., Can, F., 2021. Concept learning using one-class classi-
948 fiers for implicit drift detection in evolving data streams. Artificial
949 Intelligence Review 54, 3725–3747. URL: <https://doi.org/10.1007/s10462-020-09939-x>, doi:10.1007/s10462-020-09939-x.
- 951 Huang, J., Cheng, D., Zhang, S., 2023. A novel outlier detecting algorithm
952 based on the outlier turning points. Expert Systems with Applications 231,
953 120799. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423013015>, doi:<https://doi.org/10.1016/j.eswa.2023.120799>.
- 956 Iglesias Vázquez, F., Hartl, A., Zseby, T., Zimek, A., 2023. Anomaly detec-
957 tion in streaming data: A comparison and evaluation study. Expert Sys-
958 tems with Applications 233, 120994. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423014963>, doi:<https://doi.org/10.1016/j.eswa.2023.120994>.

- 961 Katser, I.D., Kozitsin, V.O., 2020. Skoltech anomaly benchmark
962 (skab). <https://www.kaggle.com/dsv/1693952>. doi:10.34740/KAGGLE/
963 DSV/1693952.
- 964 Kejariwal, A., 2015. Introducing practical and ro-
965 bust anomaly detection in a time series. URL:
966 https://blog.twitter.com/engineering/en_us/a/2015/introducing-practical-and-robust-anomaly-detection-in-a-time-series.
- 968 Krawczyk, B., Woźniak, M., 2015. One-class classifiers with incre-
969 mental learning and forgetting for data streams with concept drift.
970 Soft Computing 19, 3387–3400. URL: <https://doi.org/10.1007/s00500-014-1492-5>.
- 972 Laptev, N., Amizadeh, S., Flint, I., 2015. Generic and scalable frame-
973 work for automated time-series anomaly detection, in: Proceedings of
974 the 21th ACM SIGKDD International Conference on Knowledge Discov-
975 ery and Data Mining, Association for Computing Machinery, New York,
976 NY, USA. pp. 1939–1947. URL: <https://doi.org/10.1145/2783258.2788611>, doi:10.1145/2783258.2788611.
- 978 Li, J., Liu, Z., 2024. Attribute-weighted outlier detection for mixed
979 data based on parallel mutual information. Expert Systems with
980 Applications 236, 121304. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423018067>, doi:<https://doi.org/10.1016/j.eswa.2023.121304>.
- 983 Liu, B., Xiao, Y., Yu, P.S., Cao, L., Zhang, Y., Hao, Z., 2014. Uncertain
984 one-class learning and concept summarization learning on uncertain data
985 streams. IEEE Transactions on Knowledge and Data Engineering 26, 468–
986 484. doi:10.1109/TKDE.2012.235.
- 987 Lyu, Y., Li, W., Wang, Y., Sun, S., Wang, C., 2020. Rmhsforest: Relative
988 mass and half-space tree based forest for anomaly detection. Chinese Jour-
989 nal of Electronics 29, 1093–1101. doi:<https://doi.org/10.1049/cje.2020.09.010>.
- 991 Melnyk, I., Banerjee, A., Matthews, B., Oza, N., 2016. Semi-markov switch-
992 ing vector autoregressive model-based anomaly detection in aviation sys-
993 tems, in: Proceedings of the 22nd ACM SIGKDD International Confer-

- 994 ence on Knowledge Discovery and Data Mining, Association for Com-
995 puting Machinery, New York, NY, USA. pp. 1065–1074. URL: <https://doi.org/10.1145/2939672.2939789>, doi:10.1145/2939672.2939789.
- 997 Miao, X., Liu, Y., Zhao, H., Li, C., 2019. Distributed online one-class support
998 vector machine for anomaly detection over networks. IEEE Transactions
999 on Cybernetics 49, 1475–1488. doi:10.1109/TCYB.2018.2804940.
- 1000 Mishra, S., Datta-Gupta, A., 2018. Chapter 3 - distributions and models
1001 thereof, in: Mishra, S., Datta-Gupta, A. (Eds.), Applied Statistical
1002 Modeling and Data Analytics. Elsevier, pp. 31–67. URL: <https://www.sciencedirect.com/science/article/pii/B9780128032794000031>,
1003 doi:<https://doi.org/10.1016/B978-0-12-803279-4.00003-1>.
- 1005 Montiel, J., Halford, M., Mastelini, S.M., Bolmier, G., Sourty, R., Vaysse,
1006 R., Zouitine, A., Gomes, H.M., Read, J., Abdessalem, T., Bifet, A., 2021.
1007 River: machine learning for streaming data in python. Journal of Ma-
1008 chine Learning Research 22, 1–8. URL: <http://jmlr.org/papers/v22/20-1380.html>.
- 1009
1010 Nguyen, Q.P., Lim, K.W., Divakaran, D.M., Low, K.H., Chan, M.C., 2019.
1011 Gee: A gradient-based explainable variational autoencoder for network
1012 anomaly detection, in: 2019 IEEE Conference on Communications and
1013 Network Security (CNS), pp. 91–99. doi:10.1109/CNS.2019.8802833.
- 1014 Nogueira, F., 2014. Bayesian Optimization: Open source constrained
1015 global optimization tool for Python. URL: <https://github.com/fmfn/BayesianOptimization>.
- 1016
1017 Pannu, H.S., Liu, J., Fu, S., 2012. Aad: Adaptive anomaly detection system
1018 for cloud computing infrastructures, in: 2012 IEEE 31st Symposium on
1019 Reliable Distributed Systems, pp. 396–397. doi:10.1109/SRDS.2012.3.
- 1020 Ruff, L., Kauffmann, J.R., Vandermeulen, R.A., Montavon, G., Samek, W.,
1021 Kloft, M., Dietterich, T.G., Müller, K.R., 2021. A unifying review of deep
1022 and shallow anomaly detection. Proceedings of the IEEE 109, 756–795.
1023 doi:10.1109/JPROC.2021.3052449.
- 1024 Salehi, M., Rashidi, L., 2018. A survey on anomaly detection in evolving
1025 data: [with application to forest fire risk prediction]. SIGKDD Explor.

- 1026 Newslett. 20, 13–23. URL: <https://doi.org/10.1145/3229329.3229332>,
1027 doi:10.1145/3229329.3229332.
- 1028 Stauffer, T., Chastain-Knight, D., 2021. Do not let your safe oper-
1029 ating limits leave you s-o-l (out of luck). Process Safety Progress
1030 40, e12163. URL: <https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/prs.12163>,
1031 doi:<https://doi.org/10.1002/prs.12163>,
1032 arXiv:<https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/prs.12163>.
- 1033 Steenwinckel, B., 2018. Adaptive anomaly detection and root cause analy-
1034 sis by fusing semantics and machine learning, in: Gangemi, A., Gentile,
1035 A.L., Nuzzolese, A.G., Rudolph, S., Maleshkova, M., Paulheim, H., Pan,
1036 J.Z., Alam, M. (Eds.), The Semantic Web: ESWC 2018 Satellite Events,
1037 Springer International Publishing, Cham. pp. 272–282.
- 1038 Steenwinckel, B., De Paepe, D., Vanden Hautte, S., Heyvaert, P., Bente-
1039 frit, M., Moens, P., Dimou, A., Van Den Bossche, B., De Turck, F.,
1040 Van Hoecke, S., Ongena, F., 2021. Flags: A methodology for adap-
1041 tive anomaly detection and root cause analysis on sensor data streams
1042 by fusing expert knowledge with machine learning. Future Generation
1043 Computer Systems 116, 30–48. URL: <https://www.sciencedirect.com/science/article/pii/S0167739X20329927>, doi:<https://doi.org/10.1016/j.future.2020.10.015>.
- 1046 Talagala, P.D., Hyndman, R.J., Smith-Miles, K., 2021. Anomaly
1047 detection in high-dimensional data. Journal of Computational
1048 and Graphical Statistics 30, 360–374. URL: <https://doi.org/10.1080/10618600.2020.1807997>,
1049 doi:<https://doi.org/10.1080/10618600.2020.1807997>,
1050 arXiv:<https://doi.org/10.1080/10618600.2020.1807997>.
- 1051 Tartakovsky, A.G., Polunchenko, A.S., Sokolov, G., 2013. Efficient computer
1052 network anomaly detection by changepoint detection methods. IEEE Jour-
1053 nal of Selected Topics in Signal Processing 7, 4–11. doi:[10.1109/JSTSP.2012.2233713](https://doi.org/10.1109/JSTSP.2012.2233713).
- 1055 Wadinger, M., Kvasnica, M., 2023. Real-time outlier detection with dynamic
1056 process limits, in: 2023 24th International Conference on Process Control
1057 (PC), pp. 138–143. doi:[10.1109/PC58330.2023.10217717](https://doi.org/10.1109/PC58330.2023.10217717).

- 1058 Welford, B.P., 1962. Note on a method for calculating corrected sums of
1059 squares and products. *Technometrics* 4, 419–420. doi:10.1080/00401706.
1060 1962.10490022.
- 1061 Wetzig, R., Gulenko, A., Schmidt, F., 2019. Unsupervised anomaly alerting
1062 for iot-gateway monitoring using adaptive thresholds and half-space
1063 trees, in: 2019 Sixth International Conference on Internet of Things: Sys-
1064 tems, Management and Security (IOTSMS), pp. 161–168. doi:10.1109/
1065 IOTSMS48152.2019.8939201.
- 1066 Wu, H., He, J., Tömösközi, M., Xiang, Z., Fitzek, F.H., 2021. In-network
1067 processing for low-latency industrial anomaly detection in softwarized net-
1068 works, in: 2021 IEEE Global Communications Conference (GLOBECOM),
1069 pp. 01–07. doi:10.1109/GLOBECOM46510.2021.9685489.
- 1070 Wu, Z., Yang, X., Wei, X., Yuan, P., Zhang, Y., Bai, J., 2024. A self-
1071 supervised anomaly detection algorithm with interpretability. *Expert Sys-
1072 tems with Applications* 237, 121539. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423020419>, doi:<https://doi.org/10.1016/j.eswa.2023.121539>.
- 1075 Yamanishi, K., Takeuchi, J.i., 2002. A unifying framework for detecting out-
1076 liers and change points from non-stationary time series data, in: Proceed-
1077 ings of the Eighth ACM SIGKDD International Conference on Knowledge
1078 Discovery and Data Mining, Association for Computing Machinery, New
1079 York, NY, USA. pp. 676–681. URL: <https://doi.org/10.1145/775047.775148>, doi:10.1145/775047.775148.
- 1081 Yamanishi, K., Takeuchi, J.i., Williams, G., Milne, P., 2004. On-line
1082 unsupervised outlier detection using finite mixtures with discounting
1083 learning algorithms. *Data Mining and Knowledge Discovery* 8, 275–
1084 300. URL: <https://doi.org/10.1023/B:DAMI.0000023676.72185.7c>,
1085 doi:10.1023/B:DAMI.0000023676.72185.7c.
- 1086 Yang, W.T., Reis, M.S., Borodin, V., Juge, M., Roussy, A., 2022. An
1087 interpretable unsupervised bayesian network model for fault detection
1088 and diagnosis. *Control Engineering Practice* 127, 105304. URL: <https://www.sciencedirect.com/science/article/pii/S0967066122001502>,
1089 doi:<https://doi.org/10.1016/j.conengprac.2022.105304>.

- 1091 Zhang, K., Chen, J., Lee, C.G., He, S., 2024. An unsupervised spa-
1092 tiotemporal fusion network augmented with random mask and time-
1093 relative information modulation for anomaly detection of machines with
1094 multiple measuring points. Expert Systems with Applications 237,
1095 121506. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423020080>, doi:<https://doi.org/10.1016/j.eswa.2023.121506>.
- 1098 Zhang, R., Zhou, P., Qiao, J., 2023a. Anomaly detection of nonstationary
1099 long-memory processes based on fractional cointegration vector autore-
1100 gression. IEEE Transactions on Reliability , 1–12doi:10.1109/TR.2023.
1101 3314429.
- 1102 Zhang, X., Shi, J., Huang, X., Xiao, F., Yang, M., Huang, J., Yin,
1103 X., Sohail Usmani, A., Chen, G., 2023b. Towards deep probabilis-
1104 tic graph neural network for natural gas leak detection and localization
1105 without labeled anomaly data. Expert Systems with Applications 231,
1106 120542. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423010448>, doi:<https://doi.org/10.1016/j.eswa.2023.120542>.