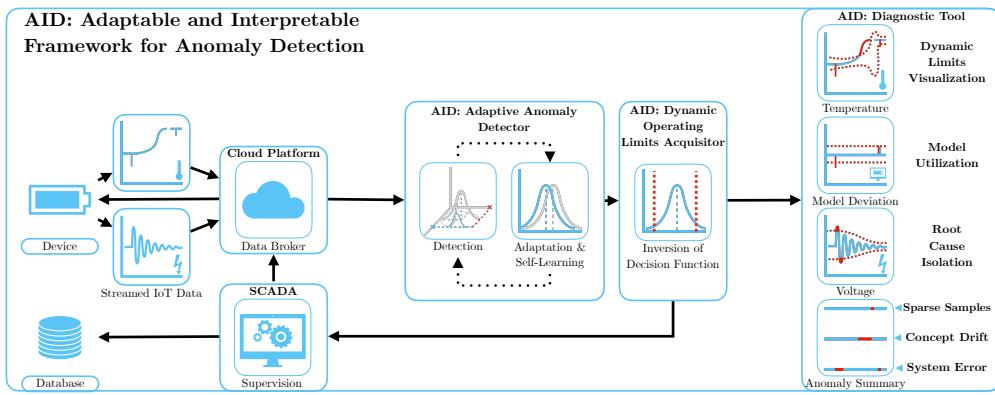


# Graphical Abstract

## 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<sup>a,\*</sup>, Michal Kvasnica<sup>a</sup>

<sup>a</sup>*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](mailto:marek.wadinger@stuba.sk) (Marek Wadinger), [michal.kvasnica@stuba.sk](mailto:michal.kvasnica@stuba.sk) (Michal Kvasnica)

---

## <sup>1</sup> 1. Introduction

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

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

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

<sup>25</sup> Additionally, feature engineering techniques play a crucial role in cap-  
<sup>26</sup> turing contextual properties and enhancing anomaly detection performance  
<sup>27</sup> (Fan et al., 2019). However, it is worth noting that feature engineering  
<sup>28</sup> may introduce categorical variables and significantly increase the dimen-  
<sup>29</sup> sionality of the data, requiring specific methods for handling large data, size-  
<sup>30</sup> able data storage, and substantial computational resources (Talagala et al.,  
<sup>31</sup> 2021). Recently, Li et al. introduced an attribute-weighted outlier detection  
<sup>32</sup> algorithm, designed for high-dimensional datasets with mixtures of categor-  
<sup>33</sup> ical and numerical data. Their approach assigns different weights to indi-  
<sup>34</sup> vidual attributes based on their importance in anomaly detection and uses  
<sup>35</sup> 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 second group

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

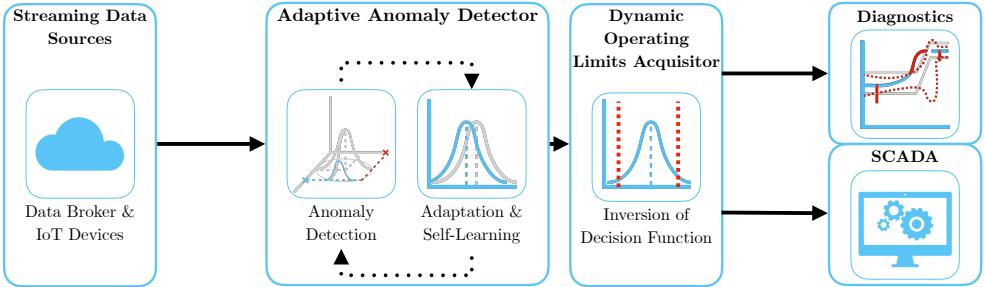


Figure 1: Schematic representation of the proposed method AID.

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 for  
111 detecting anomalies in a time series. The method is based on the assumption  
112 that the continuous data is generated by a mixture of Gaussian distributions,  
113 while discrete data is modeled as histogram density. The authors solve the  
114 problem of change point detection as well. However, the adaptation sys-  
115 tem is unaware of such changes, making the moving window the only source  
116 of adaptation. Our self-supervised approach facilitates intelligent adaptation  
117 concerning detected change points, to increase the speed of adaptation where  
118 the probability of concept drift is high. By leveraging its ability to adapt to  
119 changes in operational states, our proposed method operates autonomously  
120 when such changes occur. Moreover, Yamanishi et al. (2004) does not at-  
121 tempt to isolate the root cause of the anomaly. Our approach extends statis-  
122 tical outlier detection by incorporating interpretability. This is achieved by  
123 evaluating the inverse cumulative distribution function of the latest condi-  
124 tional probabilities for each measurement, considering the remainder of the  
125 measurements, and establishing limits that define the threshold for normal  
126 event probabilities.

127 A limited number of studies have focused on adaptation and interpretabil-  
128 ity within the framework of anomaly detection. Two recent contributions in  
129 this area are made by Steenwinckel et al. as reported in (Steenwinckel, 2018;  
130 Steenwinckel et al., 2021). In Steenwinckel (2018), the authors emphasize  
131 the importance of combining prior knowledge with a data-driven approach  
132 to achieve interpretability, particularly concerning root cause isolation. They

propose a novel approach that involves extracting features based on knowledge graph pattern extraction and integrating them into the anomaly detection mechanism. This graph is subsequently transformed into a matrix, and adaptive region-of-interest extraction is performed using reinforcement learning techniques. To enhance interpretability, a Generative Adversarial Network (GAN) reconstructs a new graphical representation based on selected vectors. However, it is important to note that the validation of this idealized approach is pending further investigation. Lately, Steenwinckel et al. (2021) introduced a comprehensive framework for adaptive anomaly detection and root cause analysis in data streams. While the adaptation process is driven by user feedback, the specific mechanism remains undisclosed. The authors present an interpretation of their method through a user dashboard, featuring visualizations of raw data. This dashboard is capable of distinguishing between track-related problems and train-related issues, based on whether multiple trains at the same geographical location approach the anomaly. Meanwhile, our efforts are directed towards the development of a self-supervised method that can learn autonomously, reducing the reliance on human supervision, which is often constrained by time limitations and can lead to significant delays in adaptation. Our method is distinguished by its straightforward statistical reasoning and the ability to isolate the root cause of anomalies. The interpretability of our method is demonstrated through the establishment of dynamic operating limits for each signal, leveraging conditional probabilities derived from the signal and other system measurements and features. This provides operators with a clear understanding of the system's state and the underlying causes of anomalies and offers interoperability with existing alarm handling mechanisms in SCADA which utilize operating limits. To the best of our knowledge, this study appears to be one of the initial attempts to introduce a self-supervised approach for adaptive anomaly detection and root cause isolation in SCADA-based systems utilizing IoT data streams.

### 1.3. Validation

Two real-world industrial-scale case studies showcase that our proposed 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

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

174 *1.4. Practical Impact*

175 Potential applications of the proposed method are in the field of energy  
176 storage systems, where the ability to detect anomalies and isolate their root  
177 causes while adapting to changes in operation and environment, is crucial  
178 for the system safety. The proposed method is designed to be integrated  
179 into the existing infrastructure of the systems, utilizing IoT data streams on  
180 top of well-established SCADA systems. SCADA systems continuously mon-  
181 itor these process data in real-time, embodying alarm handling mechanisms,  
182 which are designed to notify operators of the system’s abnormal behavior  
183 and drive attention to the root of the problem. By comparing the current  
184 values to the upper and lower operating limits, they take action when a  
185 variable exceeds or falls below these limits. However, safe operating limits  
186 are often established based on a combination of equipment design limits and  
187 the dynamics of the process (Stauffer and Chastain-Knight, 2021). Those  
188 are indifferent to the actual state of the system and environmental condi-  
189 tions. The proposed method allows the establishment of dynamic operating  
190 limits, based on the current state of the system and its environment, with  
191 direct utilization in SCADA systems expecting minimal intervention to exist-  
192 ing infrastructure. This allows the system to operate closer or further from  
193 its design limits, increasing its safety and profitability. The dynamic op-  
194 erating limits allow operational metrics monitoring, making potential early  
195 detection and prevention easier. Using general adaptable methods without  
196 interpretability, on the other hand, may pose safety risks and lower total  
197 financial benefits, as the triggered false alarms may need to be thoroughly  
198 analyzed, resulting in prolonged downtimes.

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

- 201 • Interpretable anomaly detector with self-supervised adaptation
- 202 • Demonstrates interpretability by providing dynamic operating limits
- 203 • Leverages self-learning approach on streamed IoT data

- 204     • Utilizes existing SCADA-based industrial infrastructure  
 205     • Offers faster response time to incidents due to root cause isolation

206     *1.5. Paper Organization*

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

213     **2. Preliminaries**

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

223     *2.1. Welford's Online Algorithm*

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

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

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

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

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

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

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

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

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

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

### 239 2.2. Inverting Welford's Algorithm

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

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

246 where the reverted mean is given as

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

247 Finally, the unbiased variance follows the formula:

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

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

250    2.3. Statistical Model of Multivariate System

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

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

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

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

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

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

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

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

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

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

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

279    2.4. Conditional Probability Distribution

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

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

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

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

288    and

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

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

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

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

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

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

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

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

299    2.5. Gaussian Anomaly Detection

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

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

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

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

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

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

317    for the lower limit, and

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

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

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

329 probabilistic density corresponds to the vector of conditional means  $\mu_{a|\mathbf{b}}$  with  
 330 respect to other features. The calculation of this distance involves the cumula-  
 331 tive distribution function (CDF) of observations and conditional distribu-  
 332 tions, as follows:

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

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

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

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

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

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

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

### 341 2.6. Anomaly Definition

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

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

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

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

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

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

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

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

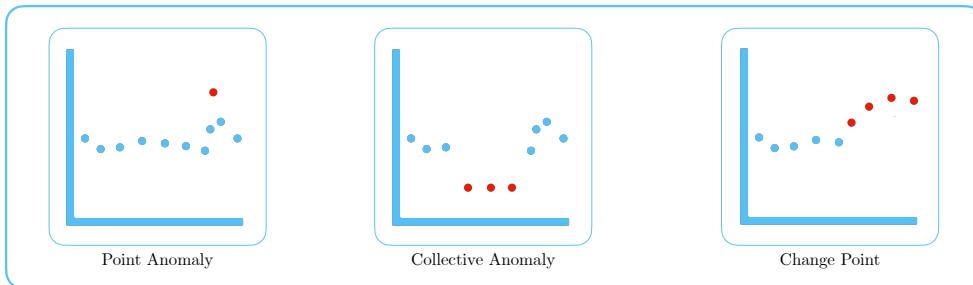


Figure 2: Illustration of point anomaly, collective anomaly, and change point.

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

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

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

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

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

407     *3.1. Online detection*

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

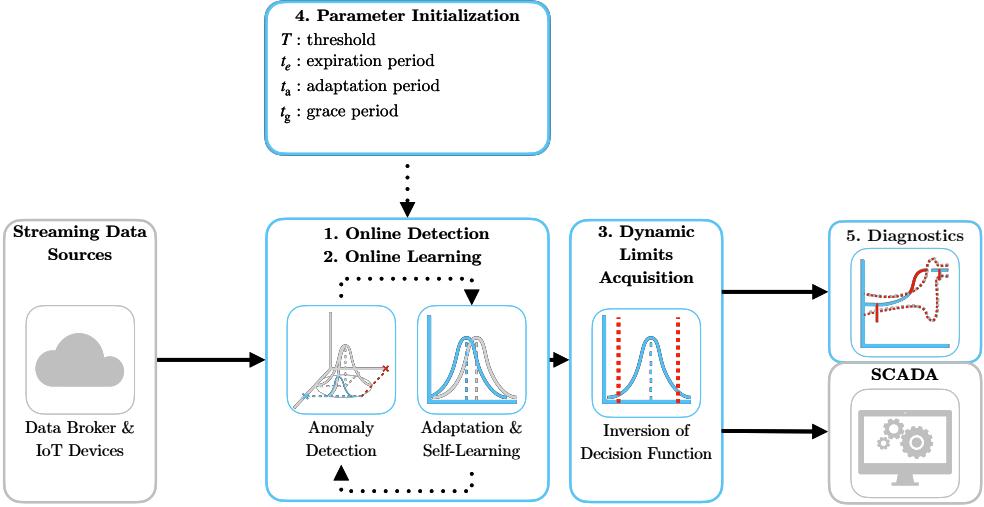


Figure 3: Schematic representation of the proposed method AID with parameter initialization. Colored boxes represent steps described within the subsection.

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

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

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

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

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

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

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

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

### 447 3.2. Online learning

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

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

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

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

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

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

494     3.3. *Dynamic limits acquisition*

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

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

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

### 518 3.4. Diagnostics

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

533    3.5. Model Parameters Initialization

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

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

564 **4. Case Study**

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

572 *4.1. Battery Energy Storage System TERRA*

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

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

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



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

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

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

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

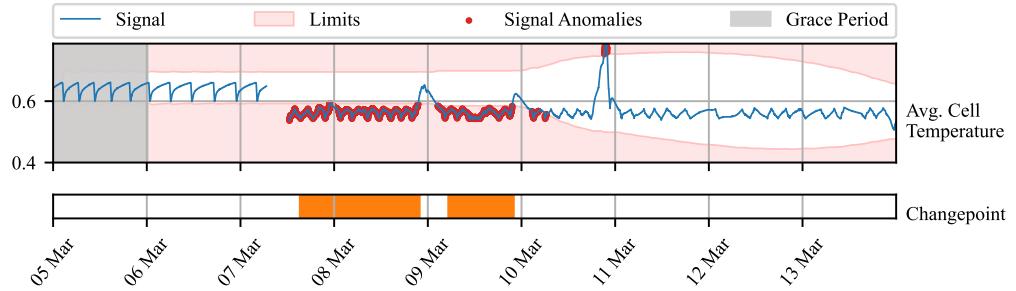


Figure 5: Normalized average cell temperature of TERRA Time series of TERRA measurements observed over 9 days (blue line). The y-axis renders the average temperature of all cells and modules after the normalization to the range of [0, 1]. The model without adaptation to change points holds dynamic operating limits (light red area) unchanged, and anomaly (red dots) alerted for approximately three days. light red area represents an area out of dynamic operating limits for individual signals. Observations out of the limits are marked by a red dot. Orange bars represent the times, at which changepoints were detected. Grace period is grayed out.

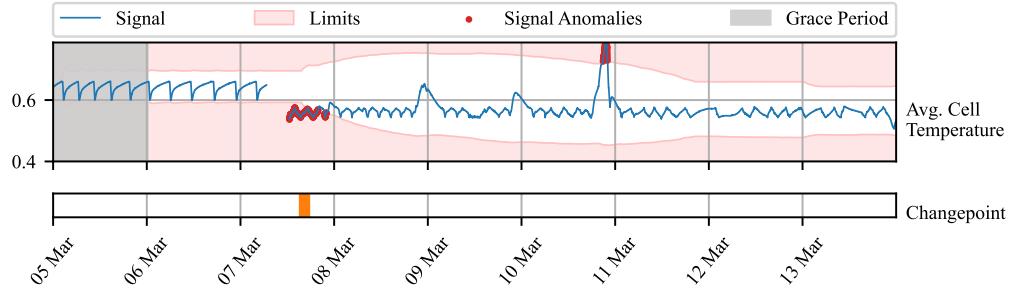


Figure 6: Normalized average cell temperature of TERRA Time series of TERRA measurements observed over 9 days (blue line). The y-axis renders the average temperature of all cells and modules after the normalization to the range of [0, 1]. The model with adaptation to change points starts to update dynamic operating limits (light red area) within 3 hours and alerts anomaly (red dots) for approximately 10 hours. light red area represents an area out of dynamic operating limits for individual signals. Observations out of the limits are marked by a red dot. Orange bars represent the times, at which changepoints were detected. Grace period is grayed out.

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

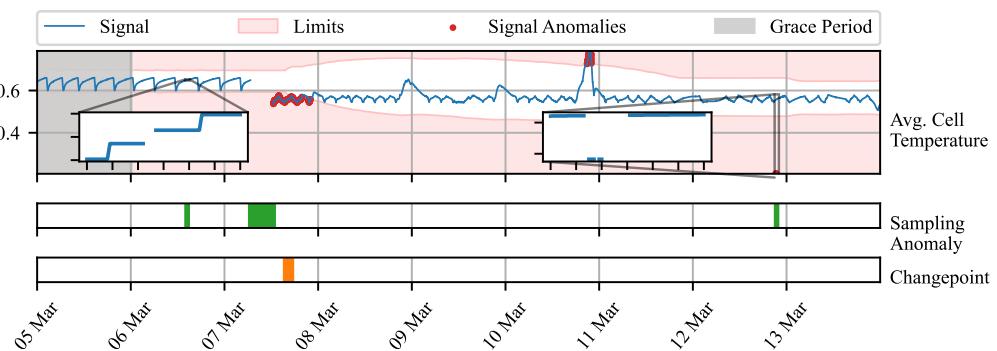


Figure 7: Normalized average cell temperature of TERRA Time series of TERRA measurements observed over 9 days (blue line). The y-axis renders the average temperature of all cells and modules after the normalization to the range of [0, 1]. The model with adaptation to change points and sampling nonuniformity alerts 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. Green bars represent the times, at which sampling anomalies are detected, while zoomed areas focus on short events of abnormal sampling. Orange bars represent the times, at which changepoints were detected. Grace period is grayed out.

631 Lastly, we want to acknowledge the outlier, left uncaptured due to in-  
 632 creased variance of the distribution in a period of adaptation. Observing  
 633 multiple variables, where some might be influenced less by the change in be-  
 634 havior, might be beneficial in such cases. The industrial partner provided a

635 physics-based model of the battery module temperature, defined as follows:

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

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

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

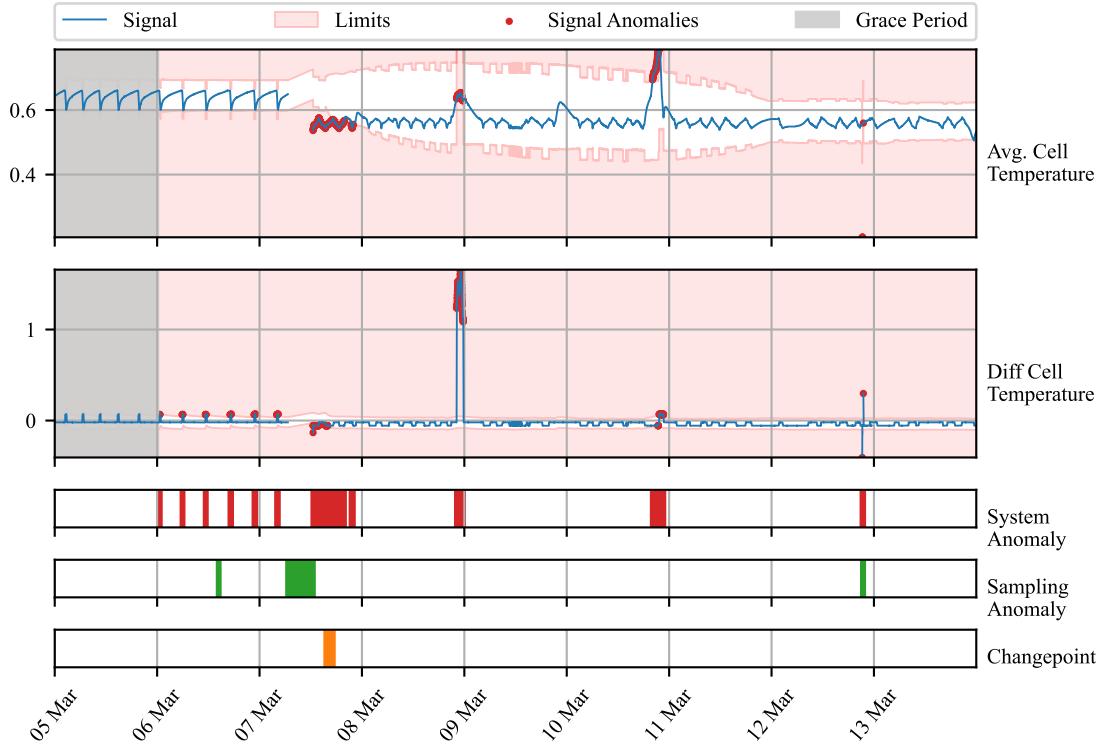


Figure 8: Time series of TERRA measurements observed over 9 days (blue line). The y-axis renders the average temperature of all cells and modules after the normalization to the range of [0, 1]. The light red area represents an area out of dynamic operating limits for individual signals. Observations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

666    4.2. Kokam Battery Module

667    A second case study presents temperature profile monitoring of individual  
668    modules of battery pack TERRA deployed at the premises of the end user.  
669    During the operation, a hardware fault of module's 9 cooling fan occurred on  
670    23<sup>rd</sup> August 2023 at 17:12:30. Our industrial partner was interested in finding  
671    out, whether such an event could be captured by an anomaly detection  
672    system. Each of the 10 modules, embodies 20 cells measured by 6 spatially  
673    distributed sensors as shown in Figure 9. The measurements are sent in 30-  
674    second intervals and processed in a streamed manner by SCADA. With the  
675    availability of the temperature profiles for all the modules, we computed the  
676    deviation of the observed value from the average of all the modules' temper-  
677    ature measurements. The ground truth information about the fan fault was  
678    provided to the best of the operator's knowledge. However, this information  
679    serves for evaluation only, as the system operates in a self-supervised manner.

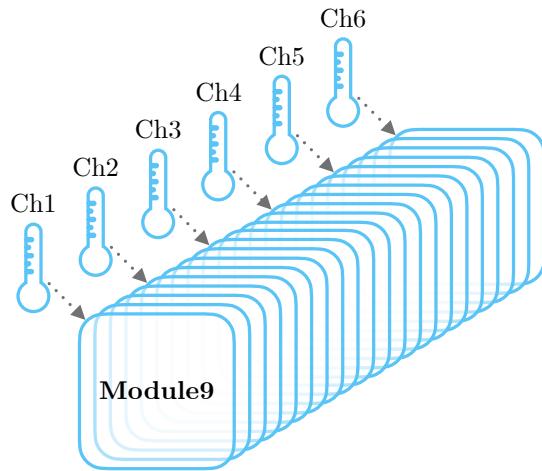


Figure 9: Module 9 with 20 cells and 6 sensors measuring the temperature at each 4<sup>th</sup> cell.

680    Our anomaly detection system was, in this case, initialized for the op-  
681    eration in production. The expiration period of 7 days, allowed the system  
682    to adapt to weekly seasonality due to the usage of the battery following  
683    work week. The grace period was kept at the default value, equal to  $t_e$ . The  
684    threshold value was shifted to a 4 sigma value of 99.977% which makes the the  
685    frequency of anomalous events approximately once a week given 30-second

686 sampling. The adaptation period was held constant as the deployed system  
687 is not expected to change its behavior dramatically on a daily basis.

688 In Figure 10 we observe 4 days of operation around the period of fan  
689 fault occurrence. The deviations between the observed temperature mea-  
690 sured by channels of module 9 and the average temperature of all modules  
691 are displayed. The dynamic operating limits tightly envelop temperatures  
692 measured by the sensors in the middle of the module (refer to Figure 9),  
693 while measurements at both sides deviate more due to the proximity to the  
694 walls and sources of disturbance. We observed multiple alarms raised by var-  
695 ious channels individually before the fan fault. These anomalies, while not  
696 addressed here further, could be subjects of interest for further investigation  
697 by system operators. Meanwhile, the fan fault at the center of our focus is  
698 alarmed based on three measurements, namely channels 1, 2, and 3. From  
699 the zoomed views, we can observe a sharp increase in the temperature devia-  
700 tion. The alarm is on until 24<sup>th</sup> August at noon, when significant fluctuations  
701 vanish followed by temporary settling of the temperature. On 25<sup>th</sup> August  
702 at 11:21, increased temperature fluctuations are followed by an increase of  
703 temperature similar to the initial one. AID alerts this fault again based on  
704 measurements by channels 1, 2, and 3.

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

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

720 This case study demonstrates the effectiveness of the AID framework in  
721 identifying hardware faults within the context of energy storage systems. It  
722 showcases the system’s ability to harness spatially distributed sensors that  
723 measure the same process variable. The AID system successfully pinpointed

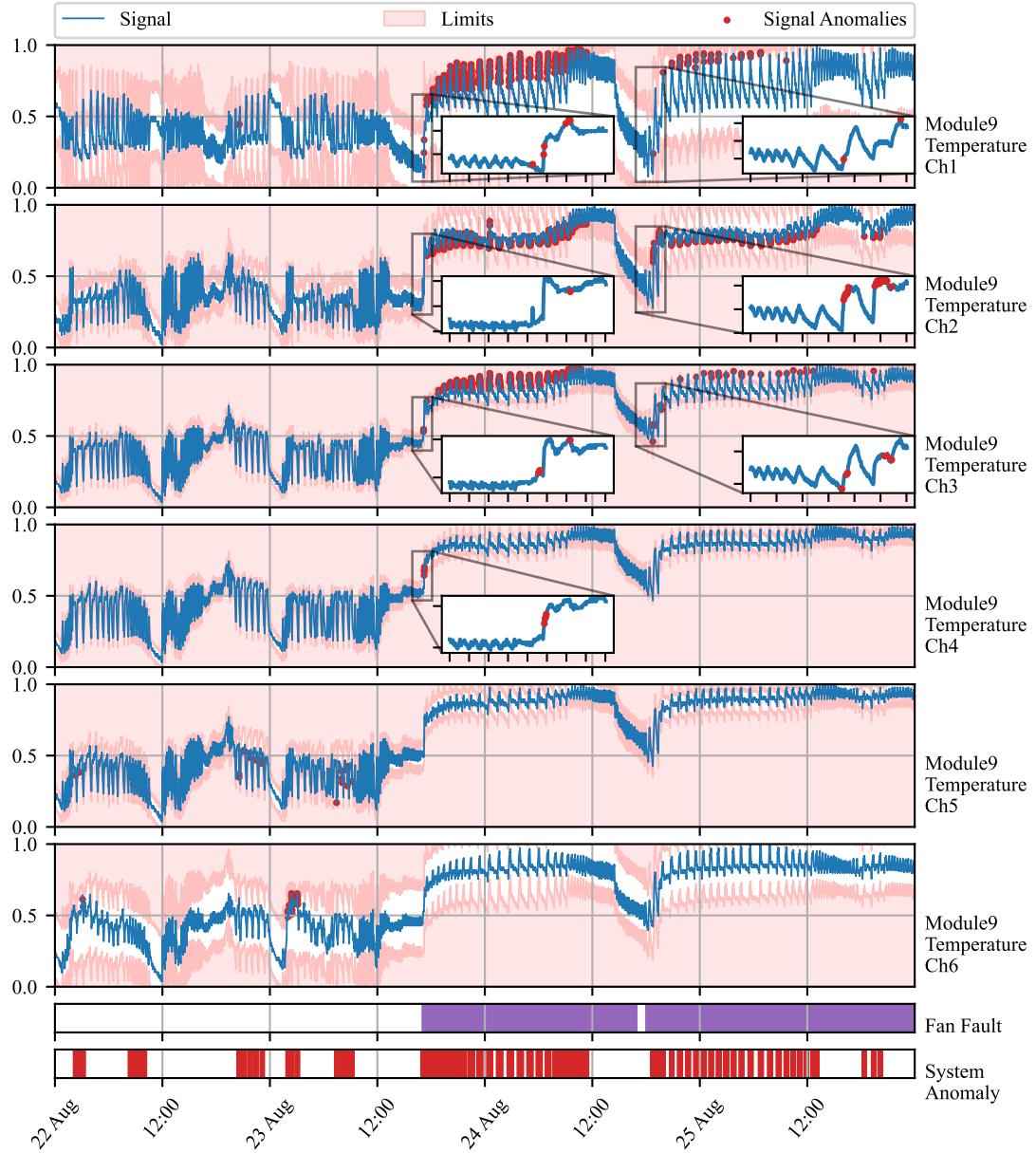


Figure 10: Time series of battery module 9 measurements (blue line). The y-axis renders the normalized deviations of temperature from the average of all modules. Signal anomalies are marked as red dots. The light red area represents an area out of dynamic operating limits. True fan faults are marked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

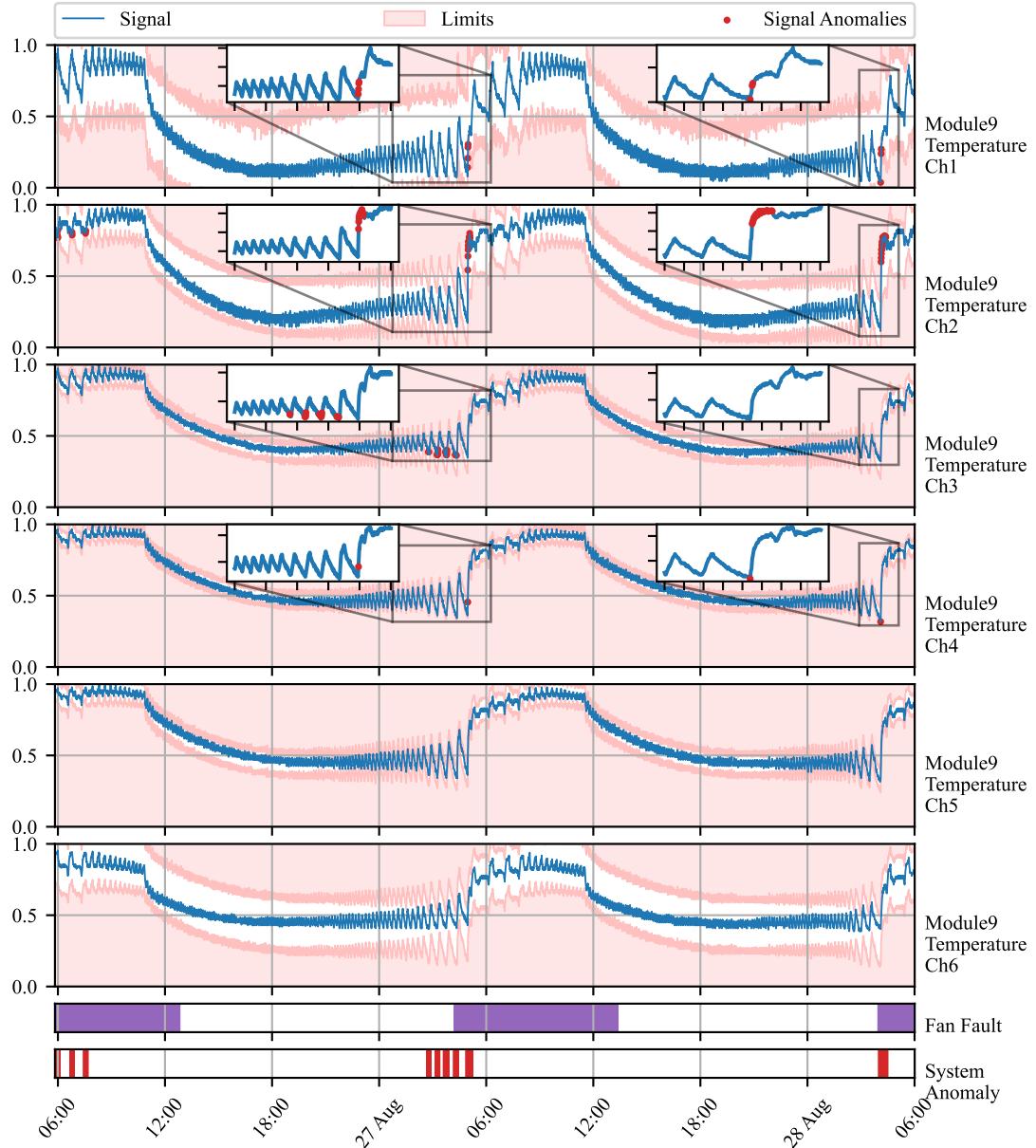


Figure 11: Time series of battery module 9 measurements (blue line). The y-axis renders the normalized deviations of temperature from an average of all modules. Signal anomalies are marked as red dots. The light red area represents an area out of dynamic operating limits. True fan faults are marked by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

724 a fault in a cooling fan during real-world production operations, underlining  
725 its practical utility and its relevance in enhancing the safety of energy storage  
726 systems. Furthermore, the incorporation of adaptation mechanisms ensures  
727 that the system can be deployed over extended periods without necessitating  
728 resource-intensive retraining. Additionally, the concept of dynamic operating  
729 limits introduced in this study holds promise for integration with Supervisory  
730 Control and Data Acquisition (SCADA) monitoring systems, enabling proactive  
731 responses in situations where human life, equipment, or the environment  
732 may be at risk.

733 *4.3. Real Data Benchmark*

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

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

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

756 The optimal hyperparameters for both reference methods are found us-  
757 ing Bayesian Optimization. Due to no further knowledge about the data  
758 generating process, and equity in benchmark, the hyperparameters of our  
759 proposed method were optimized using Bayesian Optimization as well. 20  
760 steps of random exploration with 100 iterations of Bayesian Optimization

were used, increasing default values set in the Bayesian Optimization library, to allow thorough exploration and increase the possibility of finding global optima in each case (Nogueira, 2014). The hyperparameters are optimized with the F1 score as a cost function first, to maximize both precision and recall on anomalous samples.

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

Hyperparameter search ranges are specified, with values centered around default library values for OC-SVM and HS-Trees. The ranges are intentionally set wide to facilitate comprehensive exploration. The quantile filter threshold used in OC-SVM and HS-Trees aligns with the threshold used in AID. These hyperparameter ranges are presented in Table 1.

Table 1: Hyperparameter Ranges for Detection Algorithms

| <b>Algorithm</b> | <b>Hyperparameters</b> | <b>Default</b> | <b>Ranges</b>   |
|------------------|------------------------|----------------|-----------------|
| 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)      |

The results for models optimized for the F1 score are summarized in Table 2, which includes precision, recall, F1 score, and average latency. Macro values are enclosed in brackets, representing the mean of the metric for both anomalies and normal data. A perfect detection achieves 100% in each metric **but false positive rate (FAR), where perfect detector achieves 0%**. According to the Scoreboard for various algorithms on SKAB’s Kaggle page, all iterative

784 approaches perform comparably to the batch-trained isolation forest and au-  
 785 toencoder, validating the optimization process. Notably, the proposed AID  
 786 method outperforms both reference methods in terms of **precision**, **recall**, **F1**  
 787 **score**, **area under curve**, and **false positive rate**~~F1 score, recall, and precision~~,  
 788 despite having a 30-fold higher latency per sample. This highlights the scal-  
 789 ability as a candidate for further development. Nevertheless, in this case,  
 790 sampling of the benchmark data still offers enough time to deliver predic-  
 791 tions with sufficient frequency.

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 [%]        | <b>41</b> (59) | 36 (51)     | 39 (54)     |
| Recall [%]           | <b>80</b> (59) | 74 (51)     | 63 (54)     |
| F1 [%]               | <b>54</b> (53) | 48 (44)     | 48 (52)     |
| AUC [%]              | <b>59</b>      | <b>51</b>   | <b>54</b>   |
| Mean Rolling AUC [%] | <b>57</b>      | <b>50</b>   | <b>53</b>   |
| FPR [%]              | <b>47</b>      | <b>56</b>   | <b>48</b>   |
| Avg. Latency [ms]    | 1.45           | <b>0.05</b> | <b>0.05</b> |

792 Optimal hyperparameters found during Bayesian Optimization are de-  
 793 tailed in Table 3. None of the parameters are at the edge of the provided  
 794 ranges, serving as necessary proof of ranges being broad enough. Never-  
 795 theless, sufficient proof is not possible as multiple parameter ranges are not  
 796 bounded by designed limits.

## 797 5. Conclusion

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

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     |

804 speed of adaptation in response to autonomously detected changes in operation.  
 805

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

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

826 The benchmark performed on industrial data indicates that our model  
 827 provides comparable results to other self-learning adaptable anomaly detec-

828 tion methods. This is an important property of our model, as it also allows  
829 for root cause isolation.

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

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

#### 843 Additional information

844 Our framework is openly accessible on GitHub at the following URL:  
845 [https://github.com/MarekWadinger/online\\_outlier\\_detection](https://github.com/MarekWadinger/online_outlier_detection).

#### 846 CRediT authorship contribution statement

847 **Marek Wadinger:** Conceptualization; Data curation; Formal analysis;  
848 Investigation; Methodology; Resources; Software; Validation; Visualization;  
849 Writing - original draft; and Writing - review & editing. **Michal Kvasnica:**  
850 Conceptualization; Funding acquisition; Project administration; Resources;  
851 Supervision; Validation.

#### 852 Declaration of Competing Interest

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

#### 856 Acknowledgements

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

860 **References**

- 861 Ahmad, S., Lavin, A., Purdy, S., Agha, Z., 2017. Unsuper-  
862       vised real-time anomaly detection for streaming data. Neuro-  
863       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  
864       Data Streams.
- 865       Amarasinghe, K., Kenney, K., Manic, M., 2018. Toward explainable deep  
866       neural network based anomaly detection, in: 2018 11th International Con-  
867       ference on Human System Interaction (HSI), pp. 311–317. doi:[10.1109/HSI.2018.8430788](https://doi.org/10.1109/HSI.2018.8430788).
- 868       Amer, M., Goldstein, M., Abdennadher, S., 2013. Enhancing one-class sup-  
869       port vector machines for unsupervised anomaly detection, in: Proceed-  
870       ings of the ACM SIGKDD Workshop on Outlier Detection and Descrip-  
871       tion, Association for Computing Machinery, New York, NY, USA. pp.  
872       8–15. URL: <https://doi.org/10.1145/2500853.2500857>, doi:[10.1145/2500853.2500857](https://doi.org/10.1145/2500853.2500857).
- 873       Barbosa Roa, N., Travé-Massuyès, L., Grisales-Palacio, V.H., 2019. Dy-  
874       cenee: Dynamic clustering for tracking evolving environments. Pat-  
875       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>.
- 876       Bosman, H.H., Iacca, G., Tejada, A., Wörtche, H.J., Liotta, A., 2015. En-  
877       sembles of incremental learners to detect anomalies in ad hoc sensor net-  
878       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  
879       Sensing, Processing and Networking Technologies.
- 880       Brito, L.C., Susto, G.A., Brito, J.N., Duarte, M.A.V., 2023. Fault diag-  
881       nosis using explainable ai: A transfer learning-based approach for ro-  
882       tating machinery exploiting augmented synthetic data. Expert Systems  
883       with Applications 232, 120860. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423013623>, doi:<https://doi.org/10.1016/j.eswa.2023.120860>.

- 894 Carletti, M., Masiero, C., Beghi, A., Susto, G.A., 2019. Explainable machine  
895 learning in industry 4.0: Evaluating feature importance in anomaly detec-  
896 tion to enable root cause analysis, in: 2019 IEEE International Conference  
897 on Systems, Man and Cybernetics (SMC), pp. 21–26. doi:10.1109/SMC.  
898 2019.8913901.
- 899 Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: A sur-  
900vey. ACM Comput. Surv. 41. URL: <https://doi.org/10.1145/1541880.1541882>.  
901 doi:10.1145/1541880.1541882.
- 902 Cook, A.A., Mısırlı, G., Fan, Z., 2020. Anomaly detection for iot time-  
903 series data: A survey. IEEE Internet of Things Journal 7, 6481–6494.  
904 doi:10.1109/JIOT.2019.2958185.
- 905 Deldari, S., Smith, D.V., Xue, H., Salim, F.D., 2021. Time series change point  
906 detection with self-supervised contrastive predictive coding, in: Proceedings  
907 of the Web Conference 2021, Association for Computing Machinery,  
908 New York, NY, USA. pp. 3124–3135. URL: <https://doi.org/10.1145/3442381.3449903>, doi:10.1145/3442381.3449903.
- 910 Du, X., Chen, J., Yu, J., Li, S., Tan, Q., 2024. Generative adversarial nets  
911 for unsupervised outlier detection. Expert Systems with Applications 236,  
912 121161. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423016639>, doi:<https://doi.org/10.1016/j.eswa.2023.121161>.
- 915 Fan, C., Sun, Y., Zhao, Y., Song, M., Wang, J., 2019. Deep learning-  
916 based feature engineering methods for improved building energy predic-  
917 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>.
- 920 Genz, A., 2000. Numerical computation of multivariate normal probabili-  
921 ties. Journal of Computational and Graphical Statistics 1. doi:10.1080/  
922 10618600.1992.10477010.
- 923 Gözüaçık, Ö., Can, F., 2021. Concept learning using one-class classi-  
924 fiers for implicit drift detection in evolving data streams. Artificial  
925 Intelligence Review 54, 3725–3747. URL: <https://doi.org/10.1007/s10462-020-09939-x>, doi:10.1007/s10462-020-09939-x.

- 927 Huang, J., Cheng, D., Zhang, S., 2023. A novel outlier detecting algorithm  
928 based on the outlier turning points. Expert Systems with Applications 231,  
929 120799. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423013015>, doi:<https://doi.org/10.1016/j.eswa.2023.120799>.
- 932 Iglesias Vázquez, F., Hartl, A., Zseby, T., Zimek, A., 2023. Anomaly detection  
933 in streaming data: A comparison and evaluation study. Expert Systems with Applications 233, 120994. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423014963>, doi:<https://doi.org/10.1016/j.eswa.2023.120994>.
- 937 Katser, I.D., Kozitsin, V.O., 2020. Skoltech anomaly benchmark  
938 (skab). <https://www.kaggle.com/dsv/1693952>. doi:10.34740/KAGGLE/DSV/1693952.
- 940 Kejariwal, A., 2015. Introducing practical and robust anomaly detection in a time series. URL:  
941 [https://blog.twitter.com/engineering/en\\_us/a/2015/introducing-practical-and-robust-anomaly-detection-in-a-time-series](https://blog.twitter.com/engineering/en_us/a/2015/introducing-practical-and-robust-anomaly-detection-in-a-time-series).
- 944 Krawczyk, B., Woźniak, M., 2015. One-class classifiers with incremental learning and forgetting for data streams with concept drift.  
945 Soft Computing 19, 3387–3400. URL: <https://doi.org/10.1007/s00500-014-1492-5>.
- 948 Laptev, N., Amizadeh, S., Flint, I., 2015. Generic and scalable framework for automated time-series anomaly detection, in: Proceedings of  
949 the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York,  
950 NY, USA. pp. 1939–1947. URL: <https://doi.org/10.1145/2783258.2788611>.
- 954 Li, J., Liu, Z., 2024. Attribute-weighted outlier detection for mixed  
955 data based on parallel mutual information. Expert Systems with Applications 236, 121304. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423018067>, doi:<https://doi.org/10.1016/j.eswa.2023.121304>.

- 959 Liu, B., Xiao, Y., Yu, P.S., Cao, L., Zhang, Y., Hao, Z., 2014. Uncertain  
960 one-class learning and concept summarization learning on uncertain data  
961 streams. *IEEE Transactions on Knowledge and Data Engineering* 26, 468–  
962 484. doi:10.1109/TKDE.2012.235.
- 963 Lyu, Y., Li, W., Wang, Y., Sun, S., Wang, C., 2020. Rmhsforest: Relative  
964 mass and half-space tree based forest for anomaly detection. *Chinese Jour-  
965 nal of Electronics* 29, 1093–1101. doi:<https://doi.org/10.1049/cje.2020.09.010>.
- 967 Miao, X., Liu, Y., Zhao, H., Li, C., 2019. Distributed online one-class support  
968 vector machine for anomaly detection over networks. *IEEE Transactions  
969 on Cybernetics* 49, 1475–1488. doi:10.1109/TCYB.2018.2804940.
- 970 Mishra, S., Datta-Gupta, A., 2018. Chapter 3 - distributions and models  
971 thereof, in: Mishra, S., Datta-Gupta, A. (Eds.), *Applied Statistical  
972 Modeling and Data Analytics*. Elsevier, pp. 31–67. URL: <https://www.sciencedirect.com/science/article/pii/B9780128032794000031>,  
973 doi:<https://doi.org/10.1016/B978-0-12-803279-4.00003-1>.
- 975 Montiel, J., Halford, M., Mastelini, S.M., Bolmier, G., Sourty, R., Vaysse,  
976 R., Zouitine, A., Gomes, H.M., Read, J., Abdessalem, T., Bifet, A., 2021.  
977 River: machine learning for streaming data in python. *Journal of Ma-  
978 chine Learning Research* 22, 1–8. URL: <http://jmlr.org/papers/v22/20-1380.html>.
- 980 Nguyen, Q.P., Lim, K.W., Divakaran, D.M., Low, K.H., Chan, M.C., 2019.  
981 Gee: A gradient-based explainable variational autoencoder for network  
982 anomaly detection, in: 2019 IEEE Conference on Communications and  
983 Network Security (CNS), pp. 91–99. doi:10.1109/CNS.2019.8802833.
- 984 Nogueira, F., 2014. Bayesian Optimization: Open source constrained  
985 global optimization tool for Python. URL: <https://github.com/fmfn/BayesianOptimization>.
- 987 Pannu, H.S., Liu, J., Fu, S., 2012. Aad: Adaptive anomaly detection system  
988 for cloud computing infrastructures, in: 2012 IEEE 31st Symposium on  
989 Reliable Distributed Systems, pp. 396–397. doi:10.1109/SRDS.2012.3.
- 990 Ruff, L., Kauffmann, J.R., Vandermeulen, R.A., Montavon, G., Samek, W.,  
991 Kloft, M., Dietterich, T.G., Müller, K.R., 2021. A unifying review of deep

- 992 and shallow anomaly detection. Proceedings of the IEEE 109, 756–795.  
993 doi:10.1109/JPROC.2021.3052449.
- 994 Salehi, M., Rashidi, L., 2018. A survey on anomaly detection in evolving  
995 data: [with application to forest fire risk prediction]. SIGKDD Explor.  
996 Newsl. 20, 13–23. URL: <https://doi.org/10.1145/3229329.3229332>,  
997 doi:10.1145/3229329.3229332.
- 998 Stauffer, T., Chastain-Knight, D., 2021. Do not let your safe oper-  
999 ating limits leave you s-o-l (out of luck). Process Safety Progress  
1000 40, e12163. URL: <https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/prs.12163>,  
1001 doi:<https://doi.org/10.1002/prs.12163>,  
1002 arXiv:<https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/prs.12163>.
- 1003 Steenwinckel, B., 2018. Adaptive anomaly detection and root cause analy-  
1004 sis by fusing semantics and machine learning, in: Gangemi, A., Gentile,  
1005 A.L., Nuzzolese, A.G., Rudolph, S., Maleshkova, M., Paulheim, H., Pan,  
1006 J.Z., Alam, M. (Eds.), The Semantic Web: ESWC 2018 Satellite Events,  
1007 Springer International Publishing, Cham. pp. 272–282.
- 1008 Steenwinckel, B., De Paepe, D., Vanden Hautte, S., Heyvaert, P., Bente-  
1009 frit, M., Moens, P., Dimou, A., Van Den Bossche, B., De Turck, F.,  
1010 Van Hoecke, S., Ongenae, F., 2021. Flags: A methodology for adap-  
1011 tive anomaly detection and root cause analysis on sensor data streams  
1012 by fusing expert knowledge with machine learning. Future Generation  
1013 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>.
- 1016 Talagala, P.D., Hyndman, R.J., Smith-Miles, K., 2021. Anomaly  
1017 detection in high-dimensional data. Journal of Computational  
1018 and Graphical Statistics 30, 360–374. URL: <https://doi.org/10.1080/10618600.2020.1807997>,  
1019 doi:10.1080/10618600.2020.1807997,  
1020 arXiv:<https://doi.org/10.1080/10618600.2020.1807997>.
- 1021 Tartakovsky, A.G., Polunchenko, A.S., Sokolov, G., 2013. Efficient computer  
1022 network anomaly detection by changepoint detection methods. IEEE Jour-  
1023 nal of Selected Topics in Signal Processing 7, 4–11. doi:10.1109/JSTSP.  
1024 2012.2233713.

- 1025 Wadinger, M., Kvasnica, M., 2023. Real-time outlier detection with dynamic  
1026 process limits, in: 2023 24th International Conference on Process Control  
1027 (PC), pp. 138–143. doi:10.1109/PC58330.2023.10217717.
- 1028 Welford, B.P., 1962. Note on a method for calculating corrected sums of  
1029 squares and products. *Technometrics* 4, 419–420. doi:10.1080/00401706.  
1030 1962.10490022.
- 1031 Wetzig, R., Gulenko, A., Schmidt, F., 2019. Unsupervised anomaly alerting  
1032 for iot-gateway monitoring using adaptive thresholds and half-space  
1033 trees, in: 2019 Sixth International Conference on Internet of Things: Sys-  
1034 tems, Management and Security (IOTSMS), pp. 161–168. doi:10.1109/  
1035 IOTSMS48152.2019.8939201.
- 1036 Wu, H., He, J., Tömösközi, M., Xiang, Z., Fitzek, F.H., 2021. In-network  
1037 processing for low-latency industrial anomaly detection in softwarized net-  
1038 works, in: 2021 IEEE Global Communications Conference (GLOBECOM),  
1039 pp. 01–07. doi:10.1109/GLOBECOM46510.2021.9685489.
- 1040 Wu, Z., Yang, X., Wei, X., Yuan, P., Zhang, Y., Bai, J., 2024. A self-  
1041 supervised anomaly detection algorithm with interpretability. *Expert Sys-  
1042 tems with Applications* 237, 121539. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423020419>, doi:<https://doi.org/10.1016/j.eswa.2023.121539>.
- 1045 Yamanishi, K., Takeuchi, J.i., 2002. A unifying framework for detecting out-  
1046 liers and change points from non-stationary time series data, in: Proceed-  
1047 ings of the Eighth ACM SIGKDD International Conference on Knowledge  
1048 Discovery and Data Mining, Association for Computing Machinery, New  
1049 York, NY, USA. pp. 676–681. URL: <https://doi.org/10.1145/775047.775148>. doi:10.1145/775047.775148.
- 1051 Yamanishi, K., Takeuchi, J.i., Williams, G., Milne, P., 2004. On-line  
1052 unsupervised outlier detection using finite mixtures with discounting  
1053 learning algorithms. *Data Mining and Knowledge Discovery* 8, 275–  
1054 300. URL: <https://doi.org/10.1023/B:DAMI.0000023676.72185.7c>,  
1055 doi:10.1023/B:DAMI.0000023676.72185.7c.
- 1056 Yang, W.T., Reis, M.S., Borodin, V., Juge, M., Roussy, A., 2022. An  
1057 interpretable unsupervised bayesian network model for fault detection

- 1058 and diagnosis. Control Engineering Practice 127, 105304. URL: <https://www.sciencedirect.com/science/article/pii/S0967066122001502>,  
1059 doi:<https://doi.org/10.1016/j.conengprac.2022.105304>.
- 1060  
1061 Zhang, K., Chen, J., Lee, C.G., He, S., 2024. An unsupervised spatiotemporal fusion network augmented with random mask and time-  
1062 relative information modulation for anomaly detection of machines with  
1063 multiple measuring points. Expert Systems with Applications 237,  
1064 121506. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423020080>, doi:<https://doi.org/10.1016/j.eswa.2023.121506>.  
1065  
1066  
1067  
1068 Zhang, X., Shi, J., Huang, X., Xiao, F., Yang, M., Huang, J., Yin,  
1069 X., Sohail Usmani, A., Chen, G., 2023. Towards deep probabilistic  
1070 graph neural network for natural gas leak detection and localization  
1071 without labeled anomaly data. Expert Systems with Applications 231,  
1072 120542. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423010448>, doi:<https://doi.org/10.1016/j.eswa.2023.120542>.  
1073  
1074