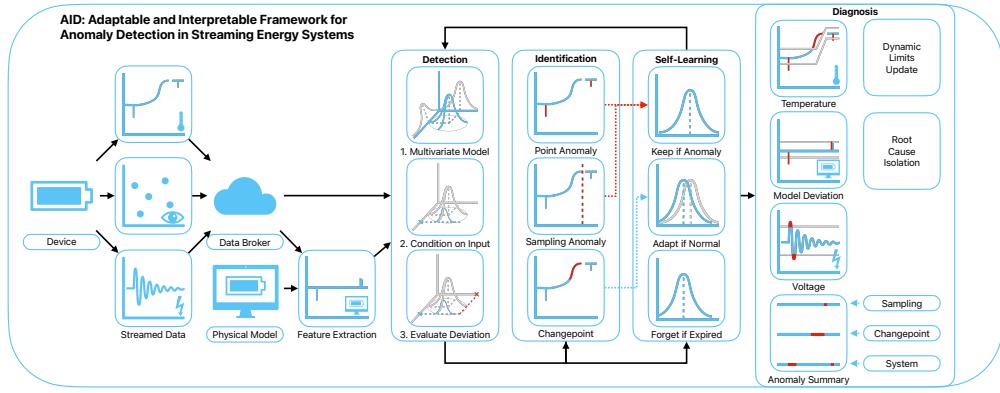


# Graphical Abstract

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

Marek Wadinger, Michal Kvasnica



## Highlights

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

Marek Wadinger, Michal Kvasnica

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

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

Marek Wadinger<sup>a,b</sup>, Michal Kvasnica<sup>a,b</sup>

<sup>a</sup>*Institute of Information Engineering, Automation and Mathematics, Slovak University of Technology in Bratislava,*

<sup>b</sup>*Tesla 50Hz s.r.o.,*

---

## Abstract

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

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

---

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

---

## <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> conducted an influential review of prior research efforts across diverse appli-  
<sup>8</sup> cation domains (Chandola et al. (2009)). Recent studies have underscored  
<sup>9</sup> the need for holistic and tunable anomaly detection methods accessible to  
<sup>10</sup> operators (Laptev 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. have suc-  
<sup>15</sup> cessfully employed spatially distributed sensors and time-relative modula-  
<sup>16</sup> tion. Their approach has proven effective, particularly in the context of com-  
<sup>17</sup> plex non-linear systems, offering potential solutions to some of the challenges  
<sup>18</sup> posed by IoT data (Zhang et al. (2024)). Huang et al., on the other hand,  
<sup>19</sup> tackled the problems of detecting global outliers, local outliers, and outlier  
<sup>20</sup> clusters simultaneously. Their proposed approach, based on density estima-  
<sup>21</sup> tion, relies on the notion that density distributions should exhibit minimal  
<sup>22</sup> variations in local areas. To achieve this, they introduce a novel turning ra-  
<sup>23</sup> tio metric, which reduces reliance on hyperparameters and enhances anomaly  
<sup>24</sup> detection (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 de-  
<sup>32</sup>tection algorithm, designed for high-dimensional datasets with mixtures of  
<sup>33</sup> categorical and numerical data. Their approach assigns different weights to  
<sup>34</sup> individual attributes based on their importance in anomaly detection and  
<sup>35</sup> uses these weights to calculate distances between data points. Notably, Li et

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

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

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

### 1.1. Related Work

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

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

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

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

91 This paper emphasizes the importance of combining adaptability in in-  
92 terpretable anomaly detection and proposes a method that addresses this  
93 challenge in real industrial systems. Here we report the discovery and char-  
94 acterization of an adaptive anomaly detection method for existing SCADA  
95 systems, employing streaming IoT data. The ability to diagnose multivariate  
96 data while providing root cause isolation via statistical learning, extends our  
97 previous contribution to the field as presented in (Wadinger and Kvasnica  
98 (2023)). The proposed algorithm represents a general method that aids a  
99 range of existing safety-critical systems where anomaly diagnosis and identi-  
100 fication are paramount.

### 101 *1.2. Novelty of proposed approach*

102 The idea of using statistical outlier detection is well-established. We  
103 highlight the impactful contributions of Yamanishi et al. in (Yamanishi and  
104 Takeuchi (2002); Yamanishi et al. (2004)). The authors propose a method for  
105 detecting anomalies in a time series. The method is based on the assumption  
106 that the continuous data is generated by a mixture of Gaussian distributions,  
107 while discrete data is modeled as histogram density. The authors solve the

108 problem of change point detection as well. However, the adaptation sys-  
109 tem is unaware of such changes, making the moving window the only source  
110 of adaptation. Our self-supervised approach facilitates intelligent adaptation  
111 concerning detected change points, to increase the speed of adaptation where  
112 the probability of concept drift is high. By leveraging its ability to adapt to  
113 changes in operational states, our proposed method operates autonomously  
114 when such changes occur. Moreover, Yamanishi et al. (2004) does not at-  
115 tempt to isolate the root cause of the anomaly. Our approach extends statis-  
116 tical outlier detection by incorporating interpretability. This is achieved by  
117 evaluating the inverse cumulative distribution function of the latest condi-  
118 tional probabilities for each measurement, considering the remainder of the  
119 measurements, and establishing limits that define the threshold for normal  
120 event probabilities.

121 A limited number of studies have focused on adaptation and interpretabil-  
122 ity within the framework of anomaly detection. Two recent contributions in  
123 this area are made by Steenwinckel et al. as reported in (Steenwinckel (2018);  
124 Steenwinckel et al. (2021)). In Steenwinckel (2018), the authors emphasize  
125 the importance of combining prior knowledge with a data-driven approach to  
126 achieve interpretability, particularly concerning root cause isolation. They  
127 propose a novel approach that involves extracting features based on knowl-  
128 edge graph pattern extraction and integrating them into the anomaly de-  
129 tection mechanism. This graph is subsequently transformed into a matrix,  
130 and adaptive region-of-interest extraction is performed using reinforcement  
131 learning techniques. To enhance interpretability, a Generative Adversarial  
132 Network (GAN) reconstructs a new graphical representation based on se-  
133 lected vectors. However, it is important to note that the validation of this  
134 idealized approach is pending further investigation. Lately, Steenwinckel  
135 et al. (2021) introduced a comprehensive framework for adaptive anomaly  
136 detection and root cause analysis in data streams. While the adaptation  
137 process is driven by user feedback, the specific mechanism remains undis-  
138 closed. The authors present an interpretation of their method through a user  
139 dashboard, featuring visualizations of raw data. This dashboard is capable of  
140 distinguishing between track-related problems and train-related issues, based  
141 on whether multiple trains at the same geographical location approach the  
142 anomaly. Meanwhile, our efforts are directed towards the development of a  
143 self-supervised method that can learn autonomously, reducing the reliance  
144 on human supervision, which is often constrained by time limitations and can  
145 lead to significant delays in adaptation. Our method is distinguished by its

146 straightforward statistical reasoning and the ability to isolate the root cause  
147 of anomalies. The interpretability of our method is demonstrated through  
148 the establishment of dynamic process limits for each signal, leveraging condi-  
149 tional probabilities derived from the signal and other system measurements  
150 and features. This provides operators with a clear understanding of the sys-  
151 tem’s state and the underlying causes of anomalies and aids existing alarm  
152 handling mechanisms in SCADA which utilize process limits. To the best of  
153 our knowledge, this is the first study to propose a self-supervised method for  
154 adaptive anomaly detection and root cause isolation in streaming IoT data.

155 *1.3. Broader Impact*

156 Potential applications of the proposed method are in the field of energy  
157 storage systems, where the ability to detect anomalies and isolate their root  
158 causes while adapting to changes in operation and environment, is crucial  
159 for the system safety. The proposed method is designed to be integrated  
160 into the existing infrastructure of the systems, utilizing IoT data streams on  
161 top of well-established SCADA systems. SCADA systems continuously mon-  
162 itor these process data in real-time, embodying alarm handling mechanisms,  
163 which are designed to notify operators of the system’s abnormal behavior  
164 and drive attention to the root of the problem. By comparing the current  
165 values to the upper and lower process limits, they take action when a vari-  
166 able exceeds or falls below these limits. However, safe operating limits are  
167 often established based on a combination of equipment design limits and the  
168 dynamics of the process (Stauffer and Chastain-Knight (2021)). Those are in-  
169 different to the actual state of the system and environmental conditions. The  
170 proposed method allows the establishment of dynamic process limits, which  
171 are based on the current state of the system and its environment. This allows  
172 the system to operate closer or further from its design limits, increasing its  
173 safety and profitability. The dynamic process limits allow operational metrics  
174 monitoring, making potential early detection and prevention easier. Using  
175 adaptable methods without interpretability, on the other hand, may pose  
176 safety risks and lower total financial benefits, as the triggered false alarms  
177 may need to be thoroughly analyzed, resulting in prolonged downtimes.

178 *1.4. Validation*

179 Two case studies show that our proposed method has the capacity to  
180 explain anomalies, isolate the root cause, and allow adaptation to change

181 points, advancing recently developed anomaly detection techniques for long-  
182 term deployment and cross-domain usage. We observe similar detection per-  
183 formance, albeit with lower scalability, on benchmark data when comparing  
184 our approach to well-established unsupervised anomaly detection methods in  
185 streamed data which create a bedrock for many state-of-the-art contributions,  
186 such as One-Class SVM (Amer et al. (2013); Liu et al. (2014); Krawczyk and  
187 Woźniak (2015); Miao et al. (2019); Gözüaçık and Can (2021)), and Half-  
188 Space Trees (Wetzig et al. (2019); Lyu et al. (2020)).

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

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

### 196 1.5. Paper Organization

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

## 203 2. Preliminaries

204 In this section, we present the fundamental ideas that form the basis  
205 of the developed approach. Subsection 2.1 explains Welford’s online algo-  
206 rithm, which can adjust distribution to changes in real-time. Subsection 2.2  
207 proposes a two-pass implementation that can reverse the impact of expired  
208 samples. The math behind distribution modeling in Subsection 2.3 estab-  
209 lishes the foundation for the Gaussian anomaly detection model discussed in  
210 Subsection 2.5, followed by conditional probability computation in Subsec-  
211 tion 2.4. The last subsection of the preliminaries is devoted to the definition  
212 of anomalies.

213    2.1. Welford's Online Algorithm

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

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

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

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

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

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

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

224    as it is less prone to numerical instability due to catastrophic cancellation,  
 225    significant loss of precision due to subtracting two nearly equal numbers.

226    Finally, the corresponding unbiased variance is

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

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

229    2.2. Inverting Welford's Algorithm

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

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

236 where the reverted mean is given as

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

237 Finally, the unbiased variance follows the formula:

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

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

### 240 2.3. Statistical Model of Multivariate System

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

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

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

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

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

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

255 The cumulative distribution function (CDF) of a multivariate Gaussian  
256 distribution describes the probability that all components of the random  
257 vector  $\mathbf{X}$  take on a value less than or equal to a particular point  $q$  in space,  
258 and can be used to evaluate the likelihood of observing a particular set of  
259 measurements or data points. In other words, it gives the probability of

260 observing a random vector that falls within a certain region of space. The  
261 standard notation of CDF is as follows:

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

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

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

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

#### 269 2.4. Conditional Probability Distribution

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

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

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

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

278 and

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

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

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

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

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

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

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

287 The conditional variance  $\sigma_{a|b}^2$  essentially represents the Schur complement  
288 of  $\Sigma_{bb}$  within the overall covariance matrix  $\Sigma$ .

### 289 2.5. Gaussian Anomaly Detection

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

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

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

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

303 Utilizing a probabilistic model of normal behavior, we can determine  
304 threshold values  $x_l$  and  $x_u$  corresponding to the closed interval of the CDF  
305 where this probability is established. The inversion of Equation (10) facilitates  
306 this calculation, yielding:

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

307 for the lower limit, and

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

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

316 The predicted state of the system, denoted as  $y_i$ , and the normality of  
 317 signals  $\mathbf{y}_{s,i}$  at time  $i$  are determined based on the maximum distance of  
 318 observations from the center of the probabilistic density. The center of the  
 319 probabilistic density corresponds to the vector of conditional means  $\mu_{a|\mathbf{b}}$  with  
 320 respect to other features. The calculation of this distance involves the cumu-  
 321 lative distribution function (CDF) of observations and conditional distribu-  
 322 tions, as follows:

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

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

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

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

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

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

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

331    *2.6. Anomaly Definition*

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

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

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

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

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

357    Notwithstanding the differences in terminology, methods employed for the  
358    identification of data points residing in low-probability regions, irrespective  
359    of whether they are referred to as "anomaly detection," "outlier detection,"  
360    or "novelty detection," share fundamental similarities (Iglesias Vázquez et al.  
361    (2023)).

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

363    In this section, we propose an adaptive and interpretable detection frame-  
364    work (AID) for multivariate systems with streaming IoT devices. This ap-  
365    proach models the system as a dynamic joint normal distribution, enabling  
366    it to effectively adapt to pervasive nonstationary effects on processes. Our

367 method handles various factors, including change points, concept drift, and  
368 seasonal effects. Our primary contribution lies in the fusion of an adaptable  
369 self-supervised system with root cause identification capabilities. This combi-  
370 nation empowers the online statistical model to diagnose anomalies through  
371 two distinct mechanisms. Firstly, it employs conditional probability calcu-  
372 lations to assess the system’s operating conditions’ normality. Secondly, it  
373 identifies outliers within individual signal measurements and features based  
374 on dynamic alert-triggering process limits. In the following sections, we de-  
375 scribe our proposed methodology across three subsections. The initial sub-  
376 section delves into the process of initializing the model’s parameters. The  
377 subsequent section describes online training and adaptation, while the final  
378 subsection expounds upon the model’s detection and diagnostic capabilities.  
379 For a concise representation of the proposed method, Algorithm 1 is provided.

380 *3.1. Model Parameters Initialization*

381 The model initialization is governed by defining two tunable hyperparam-  
382 eters of the model: the expiration period ( $t_e$ ) and the threshold ( $T$ ). The  
383 expiration period determines the window size for time-rolling computations,  
384 impacting the proportion of outliers within a given timeframe, and directly  
385 influencing the relaxation (with a longer expiration period) or tightening  
386 (with a shorter expiration period) of dynamic signal limits. Additionally, we  
387 introduce a grace period, which defaults to  $3/4t_e$ , allowing for model calibra-  
388 tion. During this grace period, system anomalies are not flagged to prevent  
389 false positives and speed up self-supervised learning, introduced in Subsec-  
390 tion 3.2. The length of the expiration period inversely correlates with the  
391 model’s ability to adapt to sudden changes. The adaptation and detection  
392 of shifts in the data-generating process, such as changes in mean or variance,  
393 is managed through the adaptation period  $t_a$ . A longer  $t_a$  results in slower  
394 adaptation but potentially longer alerts, which can be valuable when collec-  
395 tive anomalies are expected to occur. In most cases,  $t_a = t_e$  offers optimal  
396 performance.

397 As a general rule of thumb, expiration period  $t_e$  should be determined  
398 based on the slowest observed dynamics within the multivariate system. The  
399 threshold  $T$  defaults to the three-sigma probability of  $q$  in (17). Adjusting  
400 this threshold can fine-tune the trade-off between precision and recall. A  
401 lower threshold boosts recall but may lower precision, while a higher thresh-  
402 old enhances precision at the cost of recall. The presence of one non-default

403 easily interpretable hyperparameter facilitates adaptability to various sce-  
404 narios. We recommend starting with the default values of other parameters  
405 and making adjustments based on real-time model performance.

406 *3.2. Online training*

407 AID training follows an incremental learning approach, processing each  
408 new sample upon arrival. Incremental learning allows online parameter up-  
409 dates, albeit with a potential computational delay affecting response latency.

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

417 It is worth noting that adaptation relies on two self-supervised methods.  
418 Adaptation routine runs if the observation at time instance  $i$  is considered  
419 normal. Furthermore, adaptation period  $t_a$  allows the model to update the  
420 distribution on collective anomalies as well, thus speeding up the adaptation  
421 to change points. Given the predicted system anomaly state from (22) as  $y_i$   
422 over the window of past observations  $\mathbf{y}_i = \{y_{i-t_a}, \dots, y_i\}$ , the following test  
423 holds when adaptation is performed on outlier:

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

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

428 *3.3. Online prediction*

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

431 Firstly, we calculate the parameters of the conditional distribution con-  
432 cerning the dynamic multivariate Gaussian distribution. These calculations

433 are performed for the process observation vector  $\mathbf{x}_i$  at time instance  $i$ . Specifically,  
434 we compute the conditional mean using (16) and the conditional variance  
435 using (15). These computations yield univariate conditional distributions  
436 for individual signals and features. These conditional distributions play  
437 a crucial role in assessing the abnormality of signals and features concerning  
438 other observed values. This assessment relies on the strength of relationships  
439 defined by the covariance matrix of the dynamic multivariate Gaussian  
440 distribution. Consequently, our approach inherently considers the interactions  
441 between input signals and features. The determination of anomalous  
442 behavior is governed by (21).

443 Any anomaly detected within one of the features triggers an alert at the  
444 system level. The decision regarding the overall system's anomalous behavior  
445 is guided by (22). Nevertheless, individual determinations of anomalies serve  
446 as a diagnostic tool for isolating the root cause of anomalies.

447 To assist operators in their assessments, we establish a hypercube defined  
448 by lower and upper threshold values, denoted as  $\mathbf{x}_l$  and  $\mathbf{x}_u$ , respectively.  
449 These thresholds are derived from (18) and (19), incorporating updated  
450 model parameters. Lower and upper thresholds play a pivotal role as dynamic  
451 process limits. They replace the conservative process limits provided  
452 in sensor documentation, accounting for spatial factors, such as multipoint  
453 measurements, temporal factors such as aging, and actual environmental  
454 conditions that influence sensor operation.

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

464 The system is vigilant in identifying change points. When the adaptation  
465 test specified in (23) is satisfied, change points are flagged and isolated.  
466 This initiation of change points triggers updates to the model, as stated in  
467 Subsection 3.2. ensuring it adapts to evolving data patterns, such as changes  
468 in operation state, effectively.

---

**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**
- 

<sup>469</sup> **4. Case Study**

<sup>470</sup> This section presents two case studies on real industrial-scale energy stor-  
<sup>471</sup> ages and a real data benchmark to demonstrate the effectiveness and appli-  
<sup>472</sup>ability of our proposed approach. We investigate the properties and per-  
<sup>473</sup>formance of the approach using signals from IoT devices in an energy system

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

477 *4.1. Battery Energy Storage System TERRA*

478 In the first case study, we demonstrate our proposed method on real  
479 industrial-scale battery energy storage system (BESS) TERRA, depicted in  
480 Fig 1. TERRA has an installed capacity of 151 kWh distributed among  
481 10 modules with 20 cells. The Inverter's nominal power is 100 kW. The  
482 TERRA reports measurements of State of Charge (SoC), supply/draw energy  
483 set-points, and inner temperature, at 6 positions (channels) of each battery  
484 module. A substantial size of the system, which is 2.4x2.4x1.2m (HxWxD),  
485 requires a proper cooling mechanism. The cooling is handled by forced air  
486 from the HVAC system and inner fans, while the fire safety system is passive.  
487 Tight battery temperature control is needed to optimize performance and  
488 maximize the safety and battery's lifespan. Identifying anomalous events  
489 and removal of corrupted data might yield significant improvement in the  
490 process control level and increase the reliability and stability of the system.



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

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

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

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

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

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

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

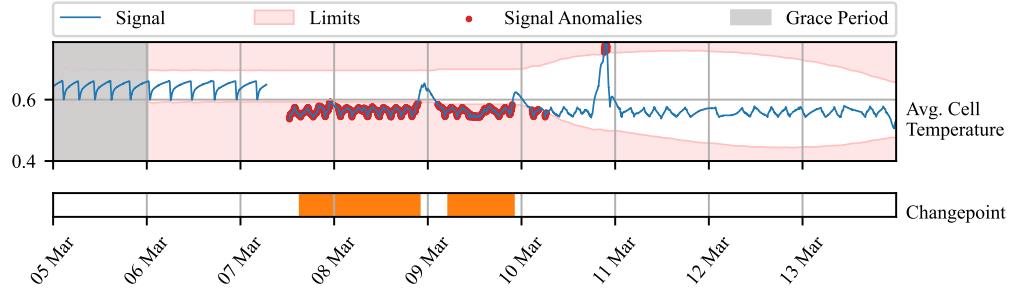


Figure 2: Time series of TERRA measurements observed over 9 days (blue line). The y-axis renders the average temperature of all cells and modules after the normalization to the range of [0, 1]. The light red area represents an area out of dynamic operating limits for individual signals. Observations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

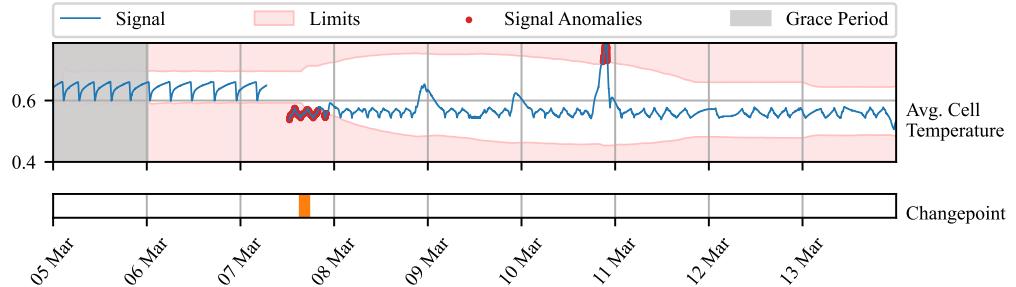


Figure 3: Time series of TERRA measurements observed over 9 days (blue line). The y-axis renders the average temperature of all cells and modules after the normalization to the range of [0, 1]. The light red area represents an area out of dynamic operating limits for individual signals. Observations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

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

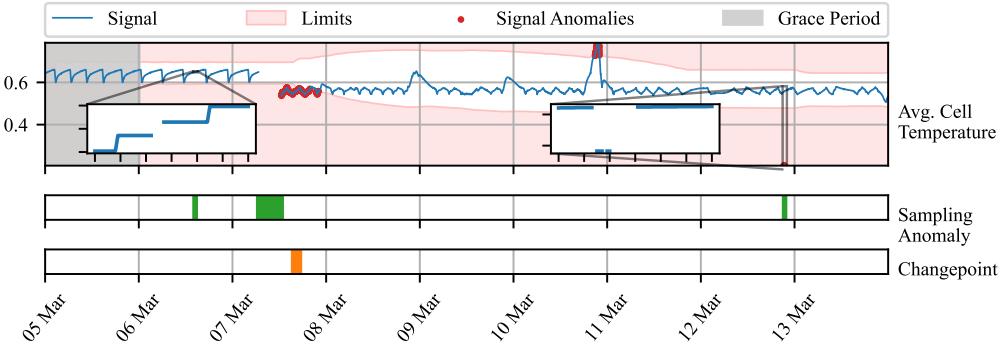


Figure 4: Time series of TERRA measurements observed over 9 days (blue line). The y-axis renders the average temperature of all cells and modules after the normalization to the range of [0, 1]. The light red area represents an area out of dynamic operating limits for individual signals. Observations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

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

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

540 When combined with an averaged measurement of battery module tem-  
 541 perature, we could compute the difference between real and predicted tem-  
 542 perature. Such deviation can be useful in detecting unexpected patterns in  
 543 temperature due to the impact of external disturbance and aging. Neverthe-  
 544 less, it may be inaccurate as the physical model is simplified and does not  
 545 account for spatial aspects, like temperature gradients as well as different

dynamic effects of charging and discharging on temperature. For instance, in Fig. 1 during the first two days we see, that the cooling dynamic is not captured well, resulting in a subtle positive difference between average cell temperature and the temperature predicted by the model. In combination with the raw measured average of the temperature, the AID captures the outlier on 9<sup>th</sup> March which could not be captured in a univariate setting. The physical model exposes temporal aspects of the behavior as it considers the dynamics of its inputs. The rapid increase in temperature w.r.t the modeled dynamics due to environmental conditions will draw a sharp positive peak in the difference between the real and predicted temperature, which will slowly vanish. Based on the significance of the deviation, the peak will be notified as a single-point anomaly or collective anomaly.

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

#### 4.2. Kokam Battery Module

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

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

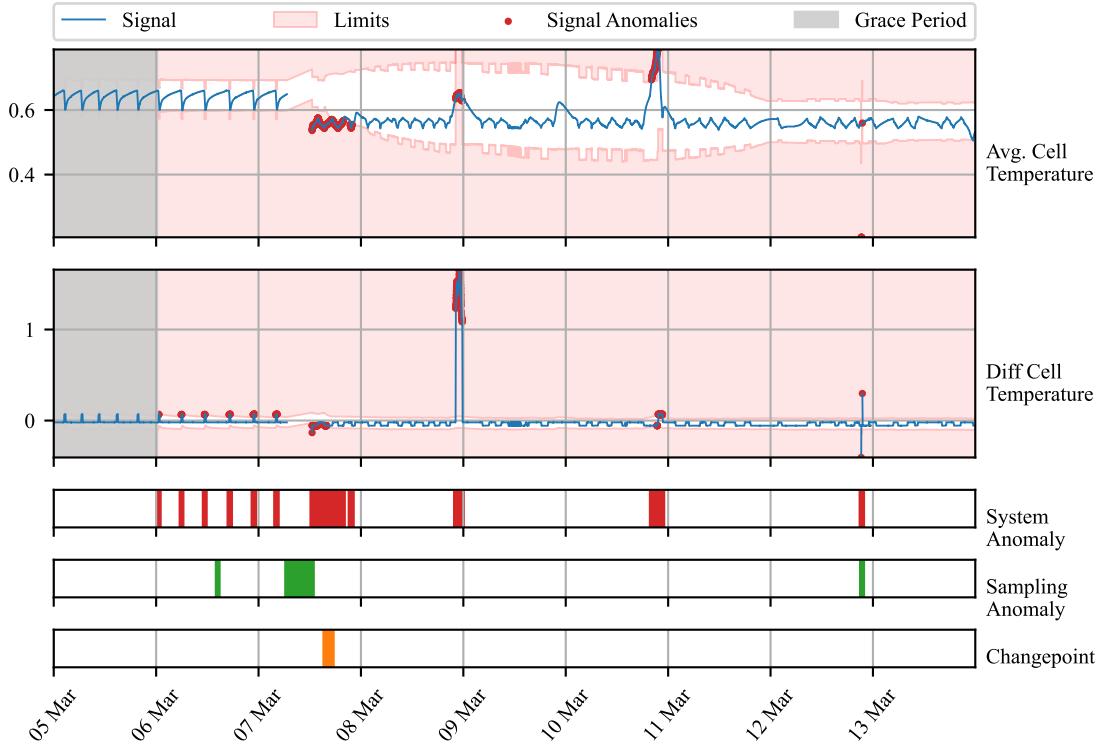


Figure 5: Time series of TERRA measurements observed over 9 days (blue line). The y-axis renders the average temperature of all cells and modules after the normalization to the range of [0, 1]. The light red area represents an area out of dynamic operating limits for individual signals. Observations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

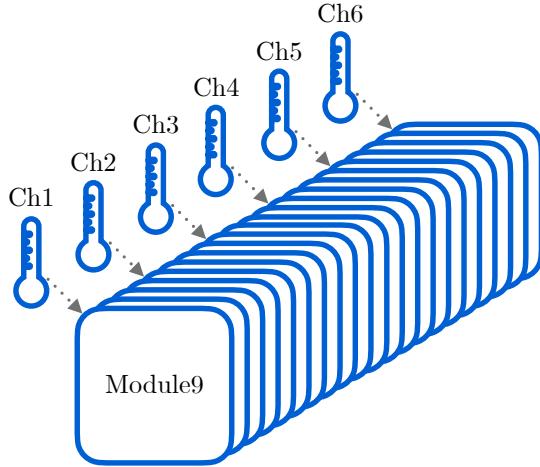


Figure 6: Module 9 with 20 cells and 6 sensors measuring the temperature at each 4<sup>th</sup> cell.

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

589 In Figure 7 we observe 4 days of operation around the period of fan  
 590 fault occurrence. The deviations between the observed temperature mea-  
 591 sured by channels of module 9 and the average temperature of all modules  
 592 are displayed. The dynamic operating limits tightly envelop temperatures  
 593 measured by the sensors in the middle of the module (refer to Figure 6),  
 594 while measurements at both sides deviate more due to the proximity to the  
 595 walls and sources of disturbance. We observed multiple alarms raised by var-  
 596 ious channels individually before the fan fault. These anomalies, while not  
 597 addressed here further, could be subjects of interest for further investigation  
 598 by system operators. Meanwhile, the fan fault at the center of our focus is  
 599 alarmed based on three measurements, namely channels 1, 2, and 3. From  
 600 the zoomed views, we can observe a sharp increase in the temperature devia-  
 601 tion. The alarm is on until 24<sup>th</sup> August at noon, when significant fluctuations  
 602 vanish followed by temporary settling of the temperature. On 25<sup>th</sup> August

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

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

614 Interestingly, during the presence of a fault in the fan, two more peri-  
615 ods where the fan started operating again followed as depicted in Figure 8.  
616 Periods of operation were interrupted again on 27<sup>th</sup> and 28<sup>th</sup> August respec-  
617 tively in the early morning hours. In both of the cases, AID detected the  
618 presence of the fault at the moment of occurrence. In the first case, channel  
619 3 reported an anomaly slightly before the increase in temperature, due to  
620 abnormal fluctuation happening prior to faults.

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

#### 634 4.3. Real Data Benchmark

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

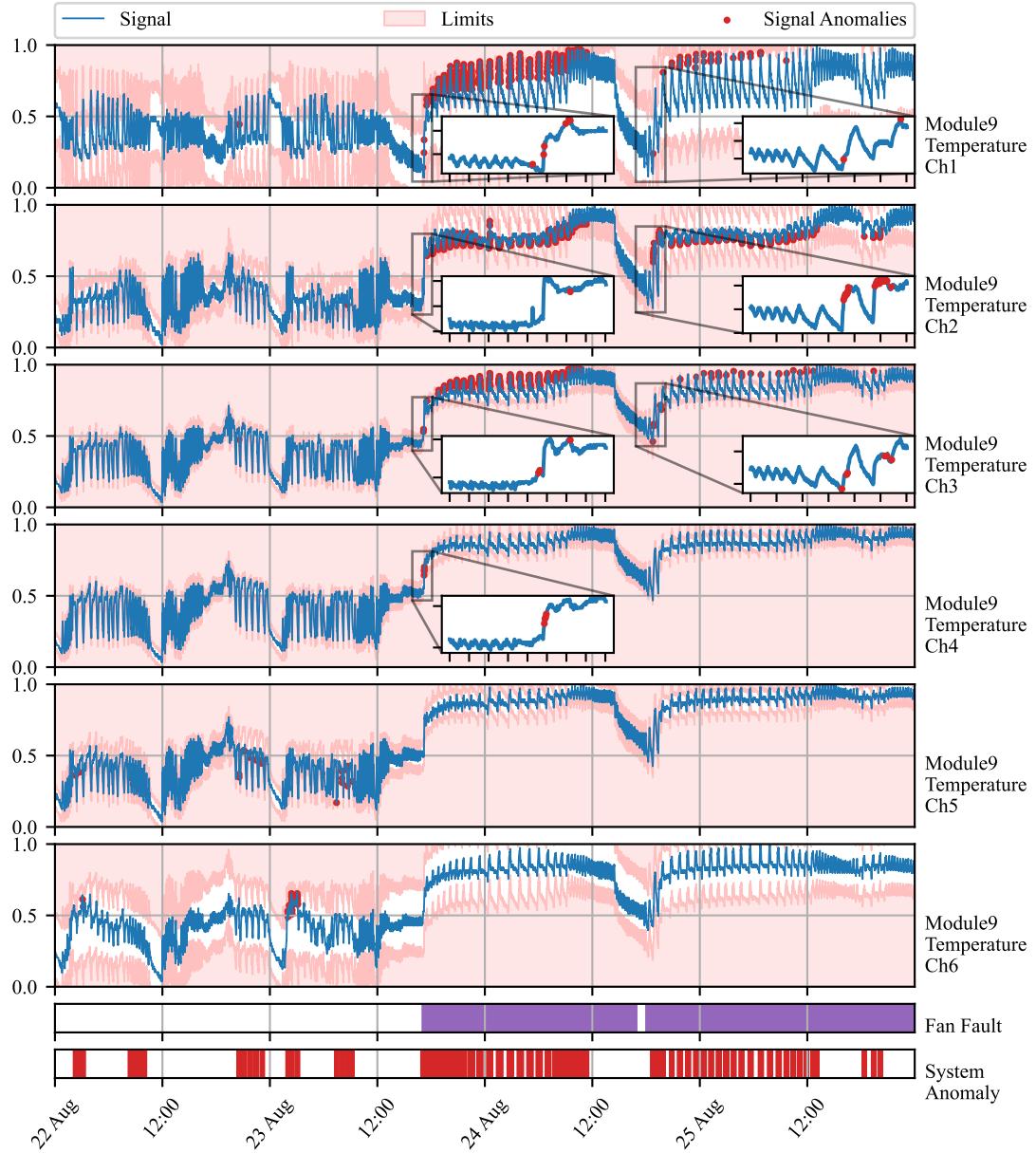


Figure 7: Time series of battery module 9 measurements (blue line). The y-axis renders the normalized deviations of temperature from the average of all modules. Signal anomalies are marked as red dots. The light red area represents an area out of dynamic operating limits. True fan faults are marked by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

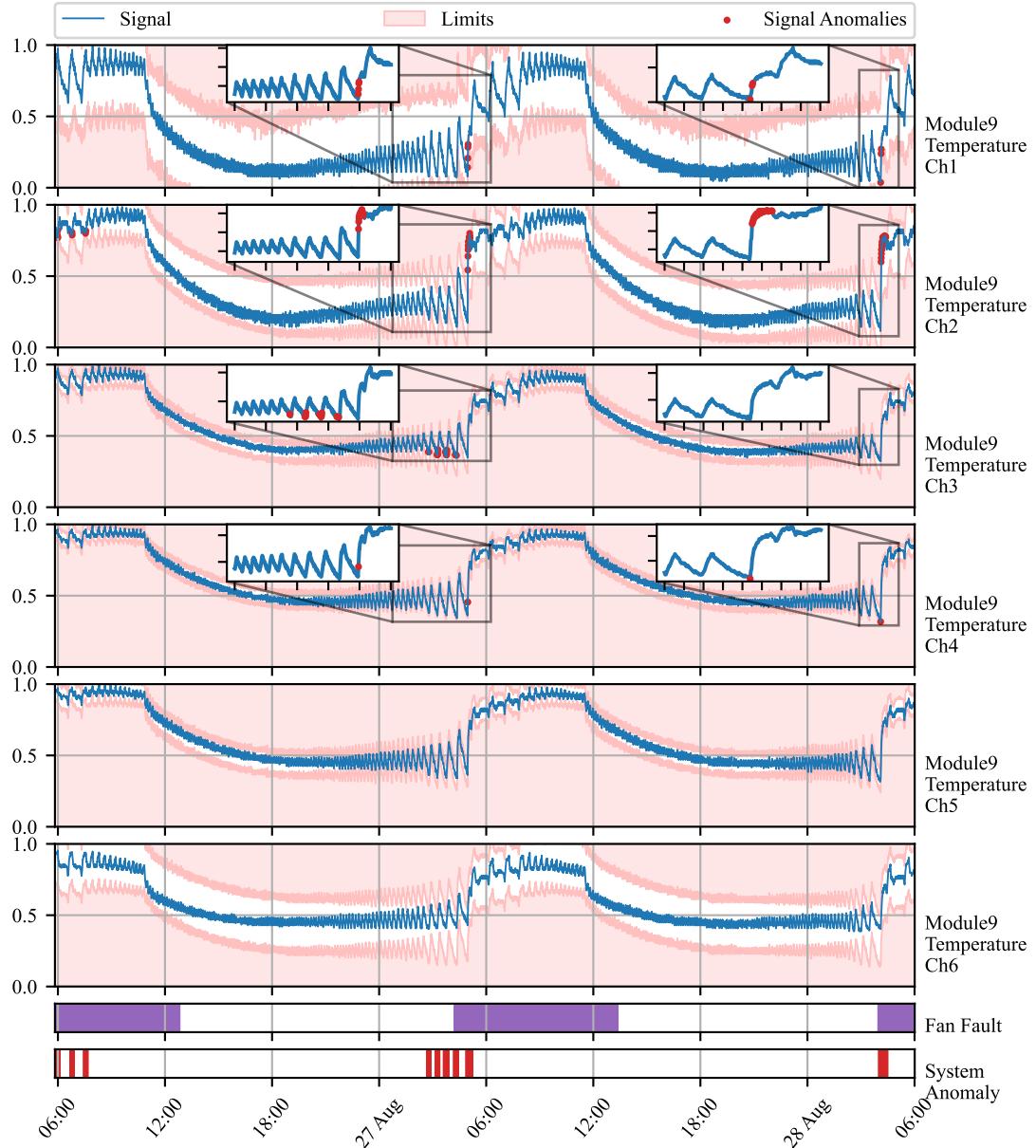


Figure 8: Time series of battery module 9 measurements (blue line). The y-axis renders the normalized deviations of temperature from an average of all modules. Signal anomalies are marked as red dots. The light red area represents an area out of dynamic operating limits. True fan faults are marked by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

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

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

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

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

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

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

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

Table 1: Hyperparameter Ranges for Detection Algorithms

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

679 The results for models optimized for the F1 score are summarized in Ta-  
 680 ble 2, which includes precision, recall, F1 score, and average latency. Macro  
 681 values are enclosed in brackets, representing the mean of the metric for both  
 682 anomalies and normal data. A perfect detection achieves 100% in each met-  
 683 ric. According to the Scoreboard for various algorithms on SKAB’s Kaggle  
 684 page, all iterative approaches perform comparably to the batch-trained iso-  
 685 lation forest and autoencoder, validating the optimization process. Notably,  
 686 the proposed AID method outperforms both reference methods in terms of  
 687 F1 score, recall, and precision, despite having a 30-fold higher latency per  
 688 sample. This highlights the scalability as a candidate for further develop-  
 689 ment. Nevertheless, in this case, sampling of the benchmark data still offers  
 690 enough time to deliver predictions with sufficient frequency.

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

## 696 5. Conclusion

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

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

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

Table 3: Optimal hyperparameters of methods optimized for F1 score

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

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

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

713 its can be seamlessly integrated into SCADA architecture, enhancing context  
714 awareness and enabling plug-and-play compatibility with existing infrastruc-  
715 ture.

716 The ability to detect and identify anomalies in the system, isolate the  
717 root cause of anomaly to specific signal or feature, and identify signal losses  
718 is shown in two case studies on data from operated industrial-scale energy  
719 storages. These case studies highlight the model’s ability to adapt, diag-  
720 nose the root cause of anomalies, and leverage both physical models and  
721 spatially distributed sensors. Unlike many anomaly detection approaches,  
722 the proposed AID method does not require historical data or ground truth  
723 information about anomalies, alleviating the general limitations of detection  
724 methods employed in the energy industry.

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

729 AID represents a significant advancement in the safety and profitability  
730 of evolving systems that utilize well-established SCADA architecture and  
731 streaming IoT data. By providing dynamic operating limits, AID seamlessly  
732 integrates with existing alarm mechanisms commonly employed in SCADA  
733 systems. To the best of our knowledge, this contribution is the first of its kind  
734 in the field of self-supervised adaptable and interpretable anomaly detection.

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

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

## 742 References

- 743 V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM  
744 Comput. Surv. 41 (2009). URL: <https://doi.org/10.1145/1541880.1541882>. doi:10.1145/1541880.1541882.  
745  
746 N. Laptev, S. Amizadeh, I. Flint, Generic and scalable framework for auto-  
747 mated time-series anomaly detection, in: Proceedings of the 21th ACM

- 748 SIGKDD International Conference on Knowledge Discovery and Data Min-  
749 ing, KDD '15, Association for Computing Machinery, New York, NY,  
750 USA, 2015, pp. 1939–1947. URL: <https://doi.org/10.1145/2783258.2788611>. doi:10.1145/2783258.2788611.
- 752 A. Kejariwal, Introducing practical and robust anomaly  
753 detection in a time series, 2015. URL: [https://blog.twitter.com/engineering/en\\_us/a/2015/introducing-practical-and-robust-anomaly-detection-in-a-time-series](https://blog.twitter.com/engineering/en_us/a/2015/introducing-practical-and-robust-anomaly-detection-in-a-time-series).
- 756 A. A. Cook, G. Misirlı, Z. Fan, Anomaly detection for iot time-series data:  
757 A survey, IEEE Internet of Things Journal 7 (2020) 6481–6494. doi:10.  
758 1109/JIOT.2019.2958185.
- 759 K. Zhang, J. Chen, C.-G. Lee, S. He, An unsupervised spa-  
760 tiotemporal fusion network augmented with random mask and time-  
761 relative information modulation for anomaly detection of machines  
762 with multiple measuring points, Expert Systems with Applica-  
763 tions 237 (2024) 121506. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423020080>. doi:<https://doi.org/10.1016/j.eswa.2023.121506>.
- 766 J. Huang, D. Cheng, S. Zhang, A novel outlier detecting algo-  
767 rithm based on the outlier turning points, Expert Systems with  
768 Applications 231 (2023) 120799. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423013015>. doi:<https://doi.org/10.1016/j.eswa.2023.120799>.
- 771 C. Fan, Y. Sun, Y. Zhao, M. Song, J. Wang, Deep learning-based fea-  
772 ture engineering methods for improved building energy prediction, Ap-  
773 plied Energy 240 (2019) 35–45. URL: <https://www.sciencedirect.com/science/article/pii/S0306261919303496>. doi:<https://doi.org/10.1016/j.apenergy.2019.02.052>.
- 776 P. D. Talagala, R. J. Hyndman, K. Smith-Miles, Anomaly detection  
777 in high-dimensional data, Journal of Computational and Graphi-  
778 cal Statistics 30 (2021) 360–374. URL: <https://doi.org/10.1080/10618600.2020.1807997>.  
779 arXiv:<https://doi.org/10.1080/10618600.2020.1807997>.

- 781 J. Li, Z. Liu, Attribute-weighted outlier detection for mixed data  
782 based on parallel mutual information, Expert Systems with Appli-  
783 cations 236 (2024) 121304. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423018067>. doi:<https://doi.org/10.1016/j.eswa.2023.121304>.
- 786 X. Du, J. Chen, J. Yu, S. Li, Q. Tan, Generative adversar-  
787 ial nets for unsupervised outlier detection, Expert Systems with  
788 Applications 236 (2024) 121161. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423016639>. doi:<https://doi.org/10.1016/j.eswa.2023.121161>.
- 791 M. Salehi, L. Rashidi, A survey on anomaly detection in evolving data:  
792 [with application to forest fire risk prediction], SIGKDD Explor. Newsl.  
793 20 (2018) 13–23. URL: <https://doi.org/10.1145/3229329.3229332>.  
794 doi:[10.1145/3229329.3229332](https://doi.org/10.1145/3229329.3229332).
- 795 N. Barbosa Roa, L. Travé-Massuyès, V. H. Grisales-Palacio, Dy-  
796 clee: Dynamic clustering for tracking evolving environments, Pat-  
797 tern Recognition 94 (2019) 162–186. URL: <https://www.sciencedirect.com/science/article/pii/S0031320319301992>. doi:<https://doi.org/10.1016/j.patcog.2019.05.024>.
- 800 A. G. Tartakovsky, A. S. Polunchenko, G. Sokolov, Efficient computer net-  
801 work anomaly detection by changepoint detection methods, IEEE Journal  
802 of Selected Topics in Signal Processing 7 (2013) 4–11. doi:[10.1109/JSTSP.2012.2233713](https://doi.org/10.1109/JSTSP.2012.2233713).
- 804 H. Wu, J. He, M. Tömösközi, Z. Xiang, F. H. Fitzek, In-network processing  
805 for low-latency industrial anomaly detection in softwarized networks, in:  
806 2021 IEEE Global Communications Conference (GLOBECOM), 2021, pp.  
807 01–07. doi:[10.1109/GLOBECOM46510.2021.9685489](https://doi.org/10.1109/GLOBECOM46510.2021.9685489).
- 808 H. S. Pannu, J. Liu, S. Fu, Aad: Adaptive anomaly detection system for cloud  
809 computing infrastructures, in: 2012 IEEE 31st Symposium on Reliable  
810 Distributed Systems, 2012, pp. 396–397. doi:[10.1109/SRDS.2012.3](https://doi.org/10.1109/SRDS.2012.3).
- 811 S. Ahmad, A. Lavin, S. Purdy, Z. Agha, Unsupervised real-time anomaly  
812 detection for streaming data, Neurocomputing 262 (2017) 134–  
813 147. URL: <https://www.sciencedirect.com/science/article/pii/>

- 814 S0925231217309864. doi:[https://doi.org/10.1016/j.neucom.2017.  
815 04.070](https://doi.org/10.1016/j.neucom.2017.04.070), online Real-Time Learning Strategies for Data Streams.
- 816 H. H. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, A. Liotta, Ensembles  
817 of incremental learners to detect anomalies in ad hoc sensor networks,  
818 Ad Hoc Networks 35 (2015) 14–36. URL: <https://www.sciencedirect.com/science/article/pii/S1570870515001481>. doi:<https://doi.org/10.1016/j.adhoc.2015.07.013>, special Issue on Big Data Inspired Data  
820 Sensing, Processing and Networking Technologies.
- 821
- 822 M. Carletti, C. Masiero, A. Beghi, G. A. Susto, Explainable machine learning  
823 in industry 4.0: Evaluating feature importance in anomaly detection to  
824 enable root cause analysis, in: 2019 IEEE International Conference on  
825 Systems, Man and Cybernetics (SMC), 2019, pp. 21–26. doi:[10.1109/SMC.2019.8913901](https://doi.org/10.1109/SMC.2019.8913901).
- 826
- 827 Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, M. C. Chan, Gee: A  
828 gradient-based explainable variational autoencoder for network anomaly  
829 detection, in: 2019 IEEE Conference on Communications and Network  
830 Security (CNS), 2019, pp. 91–99. doi:[10.1109/CNS.2019.8802833](https://doi.org/10.1109/CNS.2019.8802833).
- 831
- 832 K. Amarasinghe, K. Kenney, M. Manic, Toward explainable deep neural  
833 network based anomaly detection, in: 2018 11th International Conference  
834 on Human System Interaction (HSI), 2018, pp. 311–317. doi:[10.1109/HSI.2018.8430788](https://doi.org/10.1109/HSI.2018.8430788).
- 835
- 836 X. Zhang, J. Shi, X. Huang, F. Xiao, M. Yang, J. Huang,  
837 X. Yin, A. Sohail Usmani, G. Chen, Towards deep probabilistic  
838 graph neural network for natural gas leak detection and localiza-  
839 tion without labeled anomaly data, Expert Systems with Appli-  
840 cations 231 (2023) 120542. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423010448>. doi:<https://doi.org/10.1016/j.eswa.2023.120542>.
- 841
- 842 W.-T. Yang, M. S. Reis, V. Borodin, M. Juge, A. Roussy, An interpretable  
843 unsupervised bayesian network model for fault detection and diagno-  
844 sis, Control Engineering Practice 127 (2022) 105304. URL: <https://www.sciencedirect.com/science/article/pii/S0967066122001502>.  
845 doi:<https://doi.org/10.1016/j.conengprac.2022.105304>.
- 846

- 847 L. C. Brito, G. A. Susto, J. N. Brito, M. A. V. Duarte, Fault diagnosis  
848 using explainable ai: A transfer learning-based approach for rotating  
849 machinery exploiting augmented synthetic data, Expert Systems with  
850 Applications 232 (2023) 120860. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423013623>. doi:<https://doi.org/10.1016/j.eswa.2023.120860>.
- 853 Z. Wu, X. Yang, X. Wei, P. Yuan, Y. Zhang, J. Bai, A self-supervised  
854 anomaly detection algorithm with interpretability, Expert Systems with  
855 Applications 237 (2024) 121539. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423020419>. doi:<https://doi.org/10.1016/j.eswa.2023.121539>.
- 858 M. Wadinger, M. Kvasnica, Real-time outlier detection with dynamic process  
859 limits, in: 2023 24th International Conference on Process Control (PC),  
860 2023, pp. 138–143. doi:<https://doi.org/10.1109/PC58330.2023.10217717>.
- 861 K. Yamanishi, J.-i. Takeuchi, A unifying framework for detecting outliers and  
862 change points from non-stationary time series data, in: Proceedings of the  
863 Eighth ACM SIGKDD International Conference on Knowledge Discovery  
864 and Data Mining, KDD '02, Association for Computing Machinery, New  
865 York, NY, USA, 2002, pp. 676–681. URL: <https://doi.org/10.1145/775047.775148>. doi:<https://doi.org/10.1145/775047.775148>.
- 867 K. Yamanishi, J.-i. Takeuchi, G. Williams, P. Milne, On-line unsupervised  
868 outlier detection using finite mixtures with discounting learning  
869 algorithms, Data Mining and Knowledge Discovery 8 (2004) 275–  
870 300. URL: <https://doi.org/10.1023/B:DAMI.0000023676.72185.7c>.  
871 doi:<https://doi.org/10.1023/B:DAMI.0000023676.72185.7c>.
- 872 B. Steenwinckel, Adaptive anomaly detection and root cause analysis by fusing  
873 semantics and machine learning, in: A. Gangemi, A. L. Gentile, A. G.  
874 Nuzzolese, S. Rudolph, M. Maleshkova, H. Paulheim, J. Z. Pan, M. Alam  
875 (Eds.), The Semantic Web: ESWC 2018 Satellite Events, Springer International  
876 Publishing, Cham, 2018, pp. 272–282.
- 877 B. Steenwinckel, D. De Paepe, S. Vanden Hautte, P. Heyvaert, M. Bentefrit,  
878 P. Moens, A. Dimou, B. Van Den Bossche, F. De Turck, S. Van  
879 Hoecke, F. Ongena, Flags: A methodology for adaptive anomaly  
880 detection and root cause analysis on sensor data streams by fusing

- 881 expert knowledge with machine learning, Future Generation Com-  
882 puter Systems 116 (2021) 30–48. URL: <https://www.sciencedirect.com/science/article/pii/S0167739X20329927>. doi:<https://doi.org/10.1016/j.future.2020.10.015>.
- 885 T. Stauffer, D. Chastain-Knight, Do not let your safe operating lim-  
886 its leave you s-o-l (out of luck), Process Safety Progress 40 (2021)  
887 e12163. URL: <https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/prs.12163>.  
888 doi:<https://doi.org/10.1002/prs.12163>.  
889 arXiv:<https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/prs.12163>.
- 890 M. Amer, M. Goldstein, S. Abdennadher, Enhancing one-class support vec-  
891 tor machines for unsupervised anomaly detection, in: Proceedings of the  
892 ACM SIGKDD Workshop on Outlier Detection and Description, ODD '13,  
893 Association for Computing Machinery, New York, NY, USA, 2013, pp.  
894 8–15. URL: <https://doi.org/10.1145/2500853.2500857>. doi:[10.1145/2500853.2500857](https://doi.org/10.1145/2500853.2500857).
- 896 B. Liu, Y. Xiao, P. S. Yu, L. Cao, Y. Zhang, Z. Hao, Uncertain one-class  
897 learning and concept summarization learning on uncertain data streams,  
898 IEEE Transactions on Knowledge and Data Engineering 26 (2014) 468–  
899 484. doi:[10.1109/TKDE.2012.235](https://doi.org/10.1109/TKDE.2012.235).
- 900 B. Krawczyk, M. Woźniak, One-class classifiers with incremental  
901 learning and forgetting for data streams with concept drift, Soft  
902 Computing 19 (2015) 3387–3400. URL: <https://doi.org/10.1007/s00500-014-1492-5>. doi:[10.1007/s00500-014-1492-5](https://doi.org/10.1007/s00500-014-1492-5).
- 904 X. Miao, Y. Liu, H. Zhao, C. Li, Distributed online one-class support vector  
905 machine for anomaly detection over networks, IEEE Transactions on  
906 Cybernetics 49 (2019) 1475–1488. doi:[10.1109/TCYB.2018.2804940](https://doi.org/10.1109/TCYB.2018.2804940).
- 907 Ö. Gözüaçık, F. Can, Concept learning using one-class classifiers for  
908 implicit drift detection in evolving data streams, Artificial Intelli-  
909 gence Review 54 (2021) 3725–3747. URL: <https://doi.org/10.1007/s10462-020-09939-x>. doi:[10.1007/s10462-020-09939-x](https://doi.org/10.1007/s10462-020-09939-x).
- 911 R. Wetzig, A. Gulenko, F. Schmidt, Unsupervised anomaly alerting for  
912 iot-gateway monitoring using adaptive thresholds and half-space trees,  
913 in: 2019 Sixth International Conference on Internet of Things: Systems,

- 914 Management and Security (IOTSMS), 2019, pp. 161–168. doi:10.1109/  
915 IOTSMS48152.2019.8939201.
- 916 Y. Lyu, W. Li, Y. Wang, S. Sun, C. Wang, Rmhsforest: Relative mass  
917 and half-space tree based forest for anomaly detection, Chinese Journal  
918 of Electronics 29 (2020) 1093–1101. doi:<https://doi.org/10.1049/cje.2020.09.010>.
- 920 B. P. Welford, Note on a method for calculating corrected sums of squares  
921 and products, Technometrics 4 (1962) 419–420. doi:10.1080/00401706.  
922 1962.10490022.
- 923 S. Mishra, A. Datta-Gupta, Chapter 3 - distributions and models thereof, in: S. Mishra, A. Datta-Gupta (Eds.), Applied Statistical Modeling and Data  
924 Analytics, Elsevier, 2018, pp. 31–67. URL: <https://www.sciencedirect.com/science/article/pii/B9780128032794000031>. doi:<https://doi.org/10.1016/B978-0-12-803279-4.00003-1>.
- 928 A. Genz, Numerical computation of multivariate normal probabilities, Journal  
929 of Computational and Graphical Statistics 1 (2000). doi:10.1080/10618600.1992.10477010.
- 931 F. Iglesias Vázquez, A. Hartl, T. Zseby, A. Zimek, Anomaly detection in  
932 streaming data: A comparison and evaluation study, Expert Systems with  
933 Applications 233 (2023) 120994. URL: <https://www.sciencedirect.com/science/article/pii/S0957417423014963>. doi:<https://doi.org/10.1016/j.eswa.2023.120994>.
- 936 L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek,  
937 M. Kloft, T. G. Dietterich, K.-R. Müller, A unifying review of deep and  
938 shallow anomaly detection, Proceedings of the IEEE 109 (2021) 756–795.  
939 doi:10.1109/JPROC.2021.3052449.
- 940 I. D. Katser, V. O. Kozitsin, Skoltech anomaly benchmark (skab),  
941 <https://www.kaggle.com/dsv/1693952>, 2020. doi:10.34740/KAGGLE/DSV/1693952.
- 943 J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse,  
944 A. Zouitine, H. M. Gomes, J. Read, T. Abdessalem, A. Bifet, River: ma-  
945 chine learning for streaming data in python, Journal of Machine Learning

946 Research 22 (2021) 1–8. URL: <http://jmlr.org/papers/v22/20-1380.html>.  
947

948 F. Nogueira, Bayesian Optimization: Open source constrained global op-  
949 timization tool for Python, 2014. URL: <https://github.com/fmfn/>  
950 BayesianOptimization.