Real-Time Outlier Detection with Dynamic Process Limits

1st Marek Wadinger
Institute of Information Engineering, Automation and Mathematics
Slovak University of Technology in Bratislava
Bratislava, Slovakia
marek.wadinger@stuba.sk

2nd Michal Kvasnica
Institute of Information Engineering, Automation and Mathematics
Slovak University of Technology in Bratislava
Bratislava, Slovakia
michal.kvasnica@stuba.sk

***Abstract*—Anomaly detection methods are part of the systems where rare events may endanger an operation's profitability, safety, and environmental aspects. Although many state-of-the-art anomaly detection methods were developed to date, their deployment is limited to the operation conditions present during the model training. Online anomaly detection brings the capability to adapt to data drifts and change points that may not be represented during model development resulting in prolonged service life. This paper proposes an online anomaly detection algorithm for existing real-time infrastructures where low-latency detection is required and novel patterns in data occur unpredictably. The online inverse cumulative distribution-based approach is introduced to eliminate common problems of offline anomaly detectors, meanwhile providing dynamic process limits to normal operation. The benefit of the proposed method is the ease of use, fast computation, and deployability as shown in two case studies of real microgrid operation data.**

***Index Terms*—anomaly detection, interpretable machine learning, online machine learning, real-time systems, streaming analytics**

I. Introduction

The era of Industry 4.0 is ruled by data. Effective data-based decision-making is driven by the quantity of collected data. Internet of Things (IoT) devices made data acquisition seamless and positively influenced a wide range of industries. It is estimated that the annual economic impact of IoT will further grow and reach up to $6.2 trillion by 2025 [1].

Various data collection mechanisms are used to buffer and store the data for future processing. However, the tremendous increase in data availability and the desire to extract valuable insight led to problems with unbounded buffering.

Streaming data analytics introduced mechanisms for online extraction and transformation while loading to the storage only a fraction of the former data load, which allowed the storage of the vital information carried by the data more comprehensively. However, the unstable quality of the data appeared to have the most crucial importance over the quantity.

Anomaly detection, well studied in the last decades, was reborn to the world of new challenges. Former studies were mainly concerned with a domain-specific detection of various anomalies while trained offline [2]. However, anomalies of diverse sources, from fraudulent web activity to sensor failure, malfunctioning of the hardware, and performance drops, mutate over time, and the model had to be updated.

Companies expanded their research activities on the creation and integration of generic frameworks combining prediction, detection, and alert mechanisms. One of the first projects, open-sourced for the public, are EGADS by Yahoo [3] and AnomalyDetection by Twitter [4]. The frameworks' modularity allowed the automation of the anomaly detection of time-series data and created space for discussion.

Moving from domain-specific to generic methods posed new problems connected to type I errors, i.e., a false-positive classification of normal behavior as anomalous. Accurate selection of forecaster, detector, and alerting mechanism allowed to tackle the problem, nevertheless, introduced considerable dependence on expert domain knowledge and fine-tuning.

Further work proved improvement in performance while relieving the tight requirements on domain knowledge [5]. However, strict demands on detection systems ranging from lasting up times to continuous monitoring with stable performance pointed to the challenge of data stationarity. Change points and concept drift troubled unsupervised models, which led to service downtime due to the model retraining.

The era of adaptive machine learning introduced incremental learning schemes as a solution. Multiple studies for learning modes, adaptation methods, and model management swept through the machine learning community. Pannu et al. proposed an adaptive anomaly detection system [6]. However, the method represented a supervised operator-in-the-loop solution. Zhang et al. introduced an adaptive kernel density-based algorithm that uses an adaptive kernel width [7]. Nonetheless, training the models on big data had limitations resulting from storing and unbounded data buffering. Online learning models relaxed the need for data availability during model training [8]. On the contrary, it processed the data from a bounded buffer sequentially as in [9] and [10].

Anomaly detection in microgrids, however, called for low

The Authors gratefully acknowledge the contribution of the Slovak Research and Development Agency under the project APVV-20-0261. The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grant 1/0490/23. This research is funded by the European Commission under the grant no. 101079342 (Fostering Opportunities Towards Slovak Excellence in Advanced Control for Smart Industries).

latency detection which implied real-time training and prediction processes [11]. Such adaptation of streamed modeling took into consideration strict boundaries on computational time. For work in this area see [12] and [13].

Alerting mechanisms in process automation detect situations where signal value deviates from constraints. An alert watchdog is triggered on threshold violation by individual signals. The constraints, or process limits, are usually predefined and fixed. Nevertheless, factors such as aging and environmental changes call for dynamic process limits. Setting up a procedure for an evergrowing number of signal measurements is time-consuming. Besides, it is impossible for signals where no prior information about a correct process range is known. Those are subject to external factors that are unknown at setup time.

In this article, we suggest using existing process automation infrastructure based on alerting (PLC, SCADA, among others) and applying machine learning for dynamic process range based on changing conditions. We propose an unsupervised anomaly detection algorithm capable of online adaptation to change points and concept drifts, which adds to a recently developed body of research. The approach is evaluated on two case studies of microgrid sensors. To the author's knowledge, there are no studies to date concerned with providing adaptive operation constraints.

The main benefits of the proposed solution are that it:

- Keeps existing IT infrastructure, saving costs, and does not require operator retraining
- Automates alerting thresholds setup for a high number of signals
- Automates alerting for signals with no a priori knowledge of process limits
- Assesses changing environmental conditions and device aging
- Uses self-learning approach on streamed data

II. Preliminaries

This section introduces the main concepts which are building pillars of the developed approach. Subsection II-A will discuss a one-pass algorithm that allows for online adaptation. The following Subsection II-B proposes the ability to invert the solution in a two-pass implementation. The mathematical background of distribution modeling in Subsection II-C provides a basis for the Gaussian anomaly detection model conceptualized in the last Subsection II-D of Preliminaries.

A. Welford's Method

Streaming data analytics restricts the uncontrolled growth of memory usage by keeping only the data required for computations. One-pass algorithms allow processing on-the-fly without storing the entire data stream.

Definition 2.1 (One-pass algorithm): The algorithm with a single access to the data items in the order of their occurrence, i.e., $x_1, x_2, x_3, ...$ is called one-pass algorithm [14]

Welford's method represents a numerically stable one-pass solution for the online computation of mean and variance [15]. Given x_i where $i = 1, ..., n$ is the sample index in given population n, the corrected sum of squares is defined as

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

where the running mean $\bar{x}_n$ is

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

The following identities to update the corrected sum of squares hold true

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

and the corresponding variance is

$$s_n^2 = \frac{S_n}{n-1}. \tag{4}$$

As we can see in (3), we do access only current data sample x_n and previous value of $\bar{x}_{n-1}$ which is updated in (2) using the same data sample and the size of seen population n.

B. Inverse Welford's Method

Let the incoming stream of data be subject to the concept drift. Such alternation in statistical properties has a negative influence on prediction accuracy. An adaptation of any machine learning model is crucial for successful long-term operation.

Definition 2.2 (Concept drift): Concept drift is a change in the statistical properties that occur in a sub-region of the feature space.

The previous Subsection II-A defined the main concept of online statistical computation that allows reacting to such changes. However, the further in time the shift occurs, the slower the adjustment of the running mean is, resulting from a negative relationship in (2) between population size n and influence of the last sample in population x_n on the updated value of $\bar{x}_n$. For this reason, we define the expiration period t_e, over which the running statistics are computed. After the expiration period, the data items are forgotten. Such reversal results in a need to store all the data in the window in order to revert their effect. Given $t_e = n-1$ we can revert the influence of the first data sample on the running mean as

$$\bar{x}_{n-1} = \frac{n}{n-1}\bar{x}_n - \frac{1}{n-1}x_{n-t_e} = \bar{x}_n - \frac{x_{n-t_e} - \bar{x}_n}{n-1}, \tag{5}$$

then reverting the sum of squares follows as

$$S_{n-1} = S_n - (x_{n-t_e} - \bar{x}_{n-1})(x_{n-t_e} - \bar{x}_n), \tag{6}$$

which allows the computation of variance

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

C. Modeling Distribution

Statistical distribution can be used to create a generalized model of a normal system behavior based on observed measurement. Specifically, in cases where a change point is not anticipated within a given subset of samples, we make the assumption that the data conforms to a Gaussian normal distribution. Parameters of the normal distribution are used to compute standard score for each new observation. Standard score z_i specifies the number of sample standard deviations s_n^2 by which x_i deviates from mean $\bar{x}_n$ of normal distribution

$$z_i = \frac{x_i - \bar{x}_n}{s_n^2}. \tag{8}$$

To compute the general probability of z_i belonging to anomaly using Cumulative Distribution Function (CDF), it is bounded using an error function into the interval from 0 to 1. The error function represents the approximate probability that the observation x_i drawn from random variable X lies in the range of $[-z_i, z_i]$ denoted as

$$E_A(z_i) = z_i \frac{e^{-z_i^2}}{\sqrt{\pi}} (\,2/1 + 4/3x_i^2 + 8/15x_i^4 + ...)\,. \tag{9}$$

CDF represents the probability that the random variable X takes a value less than or equal to x_i. $F_X \colon \mathbb{R} \rightarrow [0, 1]$. For generic normal distribution with sample mean $\bar{x}_n$ and sample deviation s_n the cumulative distribution function $F_X(x)$ equals to

$$F_X(x_i)_n = \frac{1}{2}(\,1 + E_A(\,\frac{z_i}{\sqrt{2}})\,). \tag{10}$$

Given the probability, we can also derive the value of x to which it belongs using a percent point function (PPF) for numerical approximation of inverse CDF (ICDF) denoted also as $F_X(x_i)_n^{-1}$.

PPF returns the threshold value for random variable X under which it takes a value less than or equal to the value, for which $F_X(x)$ takes probability lower than selected quantile q. $F_X^{-1} \colon [0, 1] \rightarrow \mathbb{R}$. An algorithm that calculates the value of the PPF is reported below as Algorithm 1.

Algorithm 1 Percent-Point Function for Normal Distribution

```
Input: quantile $q$, sample mean $\bar{x}_n$ (2), sample variance $s_n^2$ (4)
Output: threshold value $x_{n,q}$
    Initialisation :
    $f \leftarrow 10$; $l \leftarrow -f$; $r \leftarrow f$;
    LOOP Process
    while $F_X(l) - q > 0$ do
        $r \leftarrow l$;
        $l \leftarrow lf$;
    end while
    while $F_X(r) - q < 0$ do
        $l \leftarrow r$;
        $r \leftarrow rf$;
    end while
    $\tilde{x}_{n,q} = \arg\min_z \|F_X(z) - q\|$ s.t. $l \leq z \leq r$
    return $\tilde{x}_{n,q}\sqrt{s_n^2} + \bar{x}_n$
```

D. Gaussian Anomaly Detection

Anomalies come in various kinds and flavors. Commonly denoted types are point (spatial), contextual, and collective (temporal) anomalies [2]. Spatial anomalies take on a value that particularly deviates from the sample mean $\bar{x}_n$. From a statistical viewpoint, spatial anomalies can be considered values x that significantly differ from the data distribution.

In empirical fields, such as machine learning, the three-sigma rule defines a region of distribution where normal values are expected to occur with near certainty. This assumption makes approximately 0.27% of values in the given distribution considered anomalous.

Definition 2.3 (Three-Sigma Rule of Thumb (3σ rule)): 3σ rule represents a probability, that any value x_i of random variable X will lie within a region of values of normal distribution at the distance from the sample mean μ_n of at most 3 sample standard deviations σ_n.

$$P\{|x_i - \mu_n| < 3\sigma_n\} = 0.99730 \tag{11}$$

Anomalous values occur on both tails of the distribution. In order to discriminate the anomalies using the three-sigma rule on both tails of the distribution, we define the anomaly score as follows

$$y_i = 2\left|F_X(x_i)_n - \frac{1}{2}\right|, \tag{12}$$

where

$$y_i \in [0, P\{|x_i - \mu_n| < 3\sigma_n\}), \tag{13a}$$

applies for normal observations and

$$y_i \in [P\{|x_i - \mu_n| < 3\sigma_n\}, 1], \tag{13b}$$

for anomalies.

Using pure statistics to model normal behavior lets us ask the question about the threshold value x which corresponds to the area under the curve of CDF equal to the given probability. A such query can be answered using inversion of (12). However, inversion of (12) would fail the horizontal line test. Therefore, we restrict the applicability of the inverse only to $F_X(x)_i \in [0.5, 1]$ and define upper threshold as follows

$$x_i = F_X\left(\frac{y_i}{2} + \frac{1}{2}\right)_n^{-1}. \tag{14}$$

In order to derive a lower threshold, the Gaussian distribution is fitted to the negative value of the streamed data and evaluated accordingly using the previously defined equations.

III. ICDF-based Real-Valued Threshold System

We suggest a novel approach to provide dynamic process limits using an online outlier detection algorithm capable of handling concept drifts in real-time. Our main contribution is based on using an inverse cumulative distribution function (ICDF) to supply a dynamic real-valued threshold for anomaly detection, i.e., to find the values of the signal which corresponds to the alert-triggering process limits. Therefore, in the context of machine learning, we are tackling an inverse problem, i.e., calculating the input that produced the observation. To utilize an adaptive ICDF-based threshold system, the univariate Gaussian distribution has to be fitted to the data in online training and ICDF evaluated on the fly. It is important to note that the analysis is based on the assumption that the data collected over moving windows follow a Gaussian normal distribution, rather than assuming that the data over the entire observed period follows this distribution. Thus, the influence of trends in the data can be mitigated by selecting the appropriate window size. This method is divided into four parts and described in the following lines. For a simplified representation of the method see Algorithm 2.

A. Model Initialization

The initial conditions of the model parameters are $\mu_0 = x_0$ for mean and $s_0^2 = 1$ for variance. The score threshold q is constant and set to 3σ. Moreover, there are two user-defined parameters: the expiration period t_e, and the time constant of the system t_c. The expiration period, which defines the period over which the time-rolling computations are performed, can be altered to change the proportion of expected anomalies and allows relaxation (longer expiration period) or tightening (shorter expiration period) of the thresholds. The time constant of the system determines the speed of change point adaptation as it influences the selection of anomalous points that will be used to update the model for a window of values $Y = \{y_{i-t_c}, ..., y_i\}$ if the following condition holds true

$$\frac{\sum_{y \in Y} y}{n(Y)} > q, \tag{15}$$

where $n(Y)$ represents dimensionality of Y.

The existence of two tunable and easy-to-interpret hyperparameters makes it very easy to adapt the solution to any univariate anomaly detection problem.

B. Online training

Training of the model takes place in an online fashion, i.e., the model learns one sample at a time at the moment of its arrival. Learning updates the mean and variance of the underlying Gaussian distribution. The computation of moving mean (2) and variance (4) is handled by Welford's method. Each sample after the expiration period is forgotten and its effect reverted in the second pass. First, the new mean is computed using (5) which accesses the first value in the bounded buffer. The value is dropped in the same pass. Second, the new sample variance is reverted based on (7) using the new mean and current mean that is overwritten afterward. For details see Subsection II-B.

C. Online prediction

In the prediction phase, z-score (8) is computed and passed through E_A (9) in order to evaluate $F_X(x_i)$ from (10). The algorithm marks the incoming data points if their corresponding anomaly score from (12) is out of the range defined by threshold q. In other words, it marks signal value x_i that is higher or equal to the threshold, which bounds the 3σ region.

D. Dynamic Process Limits

Normal process operation is constrained online using ICDF. The constant value of q and parameters of the fitted distribution are both passed through Algorithm 1 to obtain value, which corresponds to the value of x that would trigger an upper bound outlier alarm at the given time instance. To obtain a lower bound of operation conditions the same procedure is applied to the distribution fitted on negative values of input.

Algorithm 2 Online Anomaly Detection Workflow

```
Input: expiration period $t_e$, time constant $t_c$
Output: score $y_i$, threshold $x_{i,q}$
    Initialisation :
$i \leftarrow 1;\ n \leftarrow 1;\ q \leftarrow 0.9973;\ \bar{x} \leftarrow x_0;\ s^2 \leftarrow 1;$
compute $F_X(x_0)$ using (8);
    LOOP Process
loop
    $x_i \leftarrow$ RECEIVE();
    $y_i \leftarrow$ PREDICT($x_i$) using (12);
    $x_{i,q} \leftarrow$ GET($q, \bar{x}, s^2$) using Algorithm 1;
    if (13a) or (15) then
        $\bar{x}, s^2 \leftarrow$ UPDATE($x_i, \bar{x}, s^2, n$) using (2), (4);
        $n \leftarrow n + 1;$
        for $x_{i-t_e}$ do
            $\bar{x}, s^2 \leftarrow$ REVERT($x_{i-t_e}, \bar{x}, s^2, n$) using (5), (7);
            $n \leftarrow n - 1;$
        end for
    end if
    $i \leftarrow i + 1;$
end loop
```

IV. Case Study

In this section, we demonstrate the applicability of the proposed ICDF-based approach in two case studies of the microgrid operation. The properties and performance were investigated using streamed signals from the IoT devices. The successful deployment indicates that our approach is suitable for existing alerting mechanisms of process automation infrastructure.

The case studies were realized using Python 3.10.1 on a MAC with an M1 CPU and 8 GB RAM. The percent point function was solved using an iterative root-finding algorithm, Brent's method.

A. Battery Energy Storage System (BESS)

First, we verify our proposed method on BESS. Tight control of the battery cell temperature is needed for the optimal performance and maximum lifespan of the battery. Identifying anomalous events and removal of corrupted data might yield significant improvement on the process control level.

The sampling rate of the signal measurement is 1 minute. However, network communication is prone to packet dropout, which results in non-uniform sampling. To protect the sensitive business value of the data, we normalize all signals to the range $[0, 1]$. The goal is to mark anomalous events in the data and provide adaptive process limits from the self-learning model.

Fig. 1 renders measurement of average battery cell temperature from 21st February until 26th March. We can observe multiple anomalies of various sources given this span, for instance, packet dropout, suspicious events, intermittent sensor failure, and change points in data distribution. Dates of observation given the listed events will be provided later in the paper.

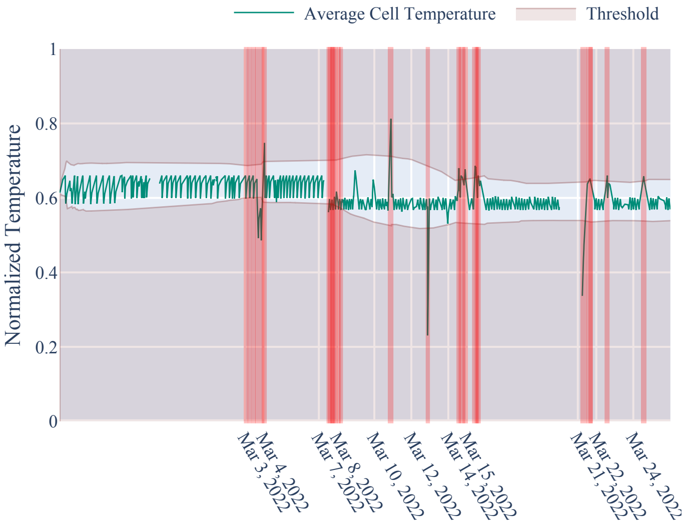

Fig. 1. Time Series of Average Battery Cell Temperature measurement (green line) and predicted anomalous events (red vertical rectangles). The reddish fill bonded by the red line represents an area of anomalous behavior as given by the anomaly detector.

The user-defined parameters were set to 7 days for the expiration period and 5 hours for the time constant. Anomalies found during the first day of the service are ignored due to the initialization of the detector. In this case study, the anomaly detection problem was approached by the online model fitting based on Subsection III-B

Using the online prediction described in Subsection III-C we tag the sample as the anomaly or normal data point. Fig. 1 renders red vertical rectangles over the regions from the start until the end of the predicted anomalous event.

The results on Average Cell Temperature in Fig. 1 show that the model could capture anomalous patterns of various sources. Despite self-learning without supervision, the model-classified anomalies were also confirmed by the data provider after inspection. For instance, a rare event of manipulation with BESS on 3rd, followed by peak on 4th March. BESS relocation on 7th, led to a change point which was alerted and the system adapted completely over the course of 1 day. Calibration resulted in peak values through 10th to 15th March, and faulty measurements on the 12th March followed by a packet loss on 21st March were alerted too. The system tagged the next two tests of temperature control switch-offs.

The model's ability to detect anomalous behavior is crucial for effective dynamic process thresholding. The real-valued threshold mechanism, described in Subsection III-D, provided up-to-date upper and lower bounds for the signal bounds. The model accurately detected breakouts from the range, as illustrated in Fig. 1. The system adapted quickly to a change point on March 7th and mitigated the effects of anomalies on the signal distribution. The user-defined parameters t_e and t_c govern the speed of adaptation and the tightness of the limits.

B. Power Inverter

A second case study demonstrates the proposed method's applicability to the temperature of the power inverter. During high load periods, inverters can heat up swiftly. Technical documentation of every inverter provides details on continuous output rating as a function of temperature that implies static process limits. Normally, for high temperatures, the rating drops rapidly. Nevertheless, the impact of aging and ambient conditions may render conservative limits impractical. Thus the alerting mechanism for the detection of abnormal heating shall be developed. Providing a real-valued anomaly threshold tightens the theoretical operating conditions and gives the ability to track the performance and deviations.

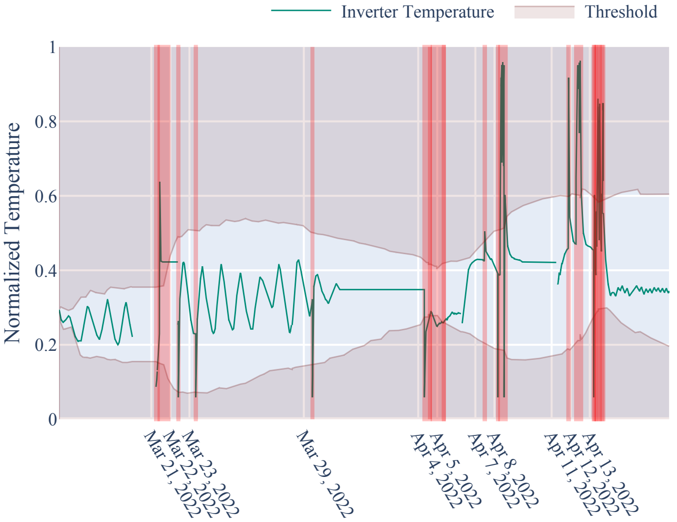

Fig. 2. Time Series of Inverter Temperature measurement (green line) and predicted anomalous events (red vertical rectangles). The reddish fill bonded by the red line represents an area of anomalous behavior.

Fig. 2 depicts one month of operation of the inverter from 16th March to 17th April 2022. After the packet loss before 21st March, rare temperature events occurred. Both events fell out of the normal operating conditions given by the dynamic process limit. Four faulty sensor readings follow from 22nd, 23rd, 29th March and 4th April. The first two are tagged as anomalies, though almost missed due to the prolonged data loss. If t_e is not fully reached, the self-learning algorithm uses anomalies for updating, indicating the need for a shorter grace period to prevent false positives. The third faulty reading was tagged without influencing the distribution and operational boundaries due to the effect of t_c. Oscillations, that kept the boundaries relaxed vanished after 29$^{\text{th}}$ March, which further tightened the process limit range. After the fourth caught fault, which was not used to update the model, the detector deliberately adapted the range of normal operation during the next day. Outliers during the sensors rescaling period from 7th April were all tagged. However, the relaxed operational conditions would probably lead to ignorance of smaller anomalous oscillations in a given period.

Lastly, Fig. 3 provides a comparison with other methods routinely used for online anomaly detection, namely, Half-Space Trees (Trees) and One-class SVM (OSVM). The detection is performed using default model parameters and a tuned quantile filter using grid search for the selection of anomalous scores. The ICDF-based method brings improved adaptation to changing conditions, retaining a low false positive rate.

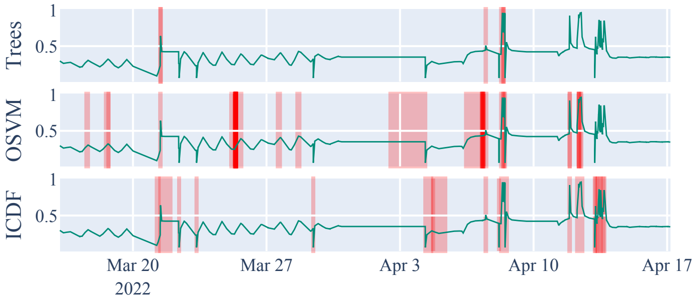

Fig. 3. Time Series of Inverter Temperature measurement (green line) and predicted anomalous events (red vertical rectangles) for Half-Spaced Trees (Trees), One-class SVM (OSVM), and proposed method (ICDF).

V. Conclusion

This paper proposes a novel approach to real-time anomaly detection that provides a physical threshold that bounds normal process operation. Such an approach has wide applicability in all the process automation fields where low latency evaluation and online adaptation are crucial. Moreover, adaptive operation constraints provide less conservative process limits and govern important insight into systems behavior. The plug-and-play feature of the model makes it easily deployable as shown in two case studies.

The first case study performed on BESS examined the average battery cell temperature and demonstrated the ability to capture anomalies as well as the capacity to restrict the operational area by inversion of the cumulative distribution function. Following our investigation of state-of-the-art online anomaly detection described in Section I, we conclude, that although the robustness and performance of complex methods may exceed the performance of the proposed method, the ability to invert the prediction to depict real-time operational restrictions and eschew using non-comprehensible parameters makes it superior for a wide range of use cases. However, the performance might be greatly afflicted when the time constraints of the observed system are not known. This restriction is much weaker than the restriction of the need for data scientists skilled in the hyper-parameter tuning of unsupervised models. Moreover, hyper-parameter tuning calls for ground truth information about anomalies, which requires an exhaustive collection and is not possible in real-time.

Future work on this method will focus on multivariate online detection, varying positive and negative process limits, and automated system identification. These challenges aim to improve the method's performance and simplify its usage.

References

[1] J. Manyika, M. Chui, J. Bughin, R. Dobbs, P. Bisson, and A. Marrs, *Disruptive technologies: Advances that will transform life, business, and the global economy*. McKinsey Global Institute, 2013.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, jul 2009. [Online]. Available: https://doi.org/10.1145/1541880.1541882

[3] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 1939–1947. [Online]. Available: https://doi.org/10.1145/2783258.2788611

[4] A. Kejariwal, "Introducing practical and robust anomaly detection in a time series," 2015. [Online]. Available: https://blog.twitter.com/engineering/en_us/a/2015/introducing-practical-and-robust-anomaly-detection-in-a-time-series

[5] S. Ahmad and S. Purdy, "Real-time anomaly detection for streaming analytics," 2016. [Online]. Available: https://arxiv.org/abs/1607.02480

[6] H. S. Pannu, J. Liu, and S. Fu, "Aad: Adaptive anomaly detection system for cloud computing infrastructures," in *2012 IEEE 31st Symposium on Reliable Distributed Systems*, 2012, pp. 396–397.

[7] L. Zhang, J. Lin, and R. Karim, "Adaptive kernel density-based anomaly detection for nonlinear systems," *Knowledge-Based Systems*, vol. 139, pp. 50–63, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705117304707

[8] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and H. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys (CSUR)*, vol. 46, 04 2014.

[9] H. H. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, and A. Liotta, "Ensembles of incremental learners to detect anomalies in ad hoc sensor networks," *Ad Hoc Networks*, vol. 35, pp. 14–36, 2015, special Issue on Big Data Inspired Data Sensing, Processing and Networking Technologies. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870515001481

[10] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017, online Real-Time Learning Strategies for Data Streams. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231217309864

[11] J. Liu, C. Su, X. Wang, W. Fang, S. Niu, and L. Cheng, "Abnormality in power system transient stability control of bess/statcom," *The Journal of Engineering*, vol. 2017, no. 13, pp. 1040–1044, 2017. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/joe.2017.0487

[12] X. Wang and S.-H. Ahn, "Real-time prediction and anomaly detection of electrical load in a residential community," *Applied Energy*, vol. 259, p. 114145, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030626191931832X

[13] Y. Dai, S. Sun, and L. Che, "Improved dbscan-based data anomaly detection approach for battery energy storage stations," in *Journal of Physics: Conference Series*, vol. 2351, 07 2022, p. 012025.

[14] N. Schweikardt, *One-Pass Algorithm*. Boston, MA: Springer US, 2009, pp. 1948–1949. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_253

[15] B. P. Welford, "Note on a method for calculating corrected sums of squares and products," *Technometrics*, vol. 4, no. 3, pp. 419–420, 1962. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00401706.1962.10490022