Incremental Change Detection Method For Data Center Power Efficiency Metrics (Work In Progress Paper)

Jana Backhus
Hitachi America, Ltd.
Santa Clara, USA
jana.backhus@hal.hitachi.com

Yasutaka Kono
Hitachi, Ltd.
Kokubunji, Japan
yasutaka.kono.vy@hitachi.com

ABSTRACT

We propose an incremental change detection method for data center (DC) energy efficiency metrics and consider its application to the power usage efficiency (PUE) metric. In recent years, there is an increasing focus on the sustainability of DCs and PUE is playing an important role to evaluate the DC's energy efficiency. Publicly reported PUE values are mostly calculated over a whole year as there are many fluctuations caused by outside influences as outdoor air temperature (OAT). In this paper, we propose a method to detect short-term changes in the DC energy efficiency (e.g., PUE) , while considering outside influences (e.g., OAT) observing related daily aggregated DC data. We also conduct a few preliminary experiments for PUE change detection based on real-world DC data, where we have manually labeled changes in the PUE using visualization tools. The experimental results show that the method can detect important major and minor changes in the PUE with a very low false positive rate. However, due to the small number of positive labels, the recall rate is currently between 57% and 70%. Further investigation is necessary to see how representative the current recall rates are and what kind of improvements are necessary to make the change detection method more stable.

CCS CONCEPTS

• **Computing methodologies → Learning linear models**; • **Information systems → Data analytics**; **Data centers**.

KEYWORDS

power usage efficiency, data center, sustainability, change detection

ACM Reference Format:
Jana Backhus and Yasutaka Kono. 2023. Incremental Change Detection Method For Data Center Power Efficiency Metrics (Work In Progress Paper). In *Companion of the 2023 ACM/SPEC International Conference on Performance Engineering (ICPE '23 Companion), April 15–19, 2023, Coimbra, Portugal.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3578245.3585027

1 INTRODUCTION

With the globally increasing demand in digital services, the data center (DC) market has kept growing rapidly over the past decade and further demand surge is expected in the future. The growth of DCs in scale and number causes an impact on the global electricity usage, where DCs currently consume approximately 1% of the total global electricity and this proportion likely will rise in the future [9]. Therefore, there is an increasing pressure on DCs to become more sustainable and many of the leading DC operators invest in ongoing efforts to be more sustainable by investments in renewable energy and improvements of energy efficiency [11]. The leading DC operators with full control over their DC hardware were able to achieve carbon neutrality and good energy efficiency based on their sustainability efforts. However, other DC operators, e.g., operators of multi-tenant DCs, struggle more to reach their sustainability goals due to lack of control over the whole DC hardware or lack of data availability to do deep analysis of DC operations.

Many energy benchmarking metrics are provided in the literature that support DC operators in evaluating their DC performance on the level of the whole DC as well as individual systems [13]. One of the most industry-preferred metrics is the power usage efficiency (PUE) [2] that measures the DC infrastructure energy efficiency and many DC operators report PUE values publicly. PUE is defined by

$$PUE = \frac{TotalFacilityEnergy}{ITEquipmentEnergy}, \tag{1}$$

where the total energy involves all DC facilities including cooling and power provision system infrastructure as well as other supporting infrastructure, e.g., lighting. On the other hand, IT equipment energy is limited to IT-related infrastructure like network devices, storages, and servers. PUE is widely used as a metric to compare the infrastructure power efficiency of different DCs. Highly efficient DCs can achieve PUE values around 1.1 [8], but the average reported DC PUE was 1.58 in 2020, which is only marginally better than 7 years earlier [10]. The PUE metric is based on the idea that the supporting infrastructure's energy usage should be as low as possible compared to the IT equipment energy, and the theoretically best achievable PUE value is 1.0 meaning that all energy in the DC is consumed only by the IT equipment. Therefore PUE is not accounting for the energy efficiency of the IT equipment but only for the efficiency of DC's supporting infrastructure [14].

In this paper, we propose an incremental change detection method that is designed with PUE in mind, but application to other efficiency metrics is possible in the future. Publicly reported PUE values are mostly average values aggregated over longer time periods, like a whole year, or even spatially over several DC locations. One important reason for the often-practiced long-term aggregation of PUE values is to account for all observable weather conditions in different seasons giving. Therefore, long-term PUE calculation draws an overall picture of the DC's energy efficiency ignoring

ICPE '23 Companion, April 15–19, 2023, Coimbra, Portugal.

ACM ISBN 979-8-4007-0072-9/23/04…$15.00
https://doi.org/10.1145/3578245.3585027

weather influences. However, we believe that observing PUE values on a more frequent basis, e.g., daily, can give us new operational insights. Unfortunately, detailed breakdowns of operational DC data are not widely available in the literature. For instance, we want to identify short-term changes in the energy efficiency that have been caused by operational changes in the DC infrastructure. Timely feedback about these changes, i.e., increasing or decreasing PUE values, can support operators to continuously improve their DC's sustainability. We have observed real-world energy data from a multi-tenant air-cooled DC, where sudden PUE changes are likely caused by differences in the operational setup of the cooling infrastructure or IT equipment usage rates. We have observed some bigger PUE changes in the data that worsen energy efficiency over longer time periods for up to a year. Early feedback of such worsening efficiency use cases can help DC operators to investigate performance inefficiencies sooner.

To detect energy efficiency changes in a timely manner, we design our detection method to be incremental. The change detection method uses a predictive model-based approach where first a machine learning (ML) model is learned and then based on a prediction error it is decided whether a change has occurred or not. Preliminary experiments with real-world DC data show that the proposed method can detect major and some minor changes in the PUE from the data. However, not all changes are detected, and our current recall rate is between 57% to 70% for three experiments with a relatively small change sample size. On the other hand, there are only two false positives detected over all three experiments where the total time span for each experiment is 33 months.

The paper is divided as follows: in Section 2 we discuss background and related research, in Section 3 we introduce our proposed methodology and in Section 4 we discuss preliminary experimental results. Finally in Section 5 we give our conclusion and mention future work.

2 BACKGROUND AND RELATED RESEARCH

2.1 Partial PUE

The PUE is generally comparing the total DC facility energy to the IT equipment energy as defined in Equation 1. Depending on the use case, it is however not always feasible or of interest to calculate the PUE of the whole facility. Here, the Green Grid [2] additionally introduced the partial PUE that is defined as the total energy inside a boundary divided by the IT equipment energy inside the same boundary. For our purpose, we are interested in detecting energy efficiency changes within less aggregated boundaries, e.g., server rooms, for the same DC facility. Therefore, we will use PUE with the meaning of partial PUE for the rest of this paper. For our experiments, we obtain real-world energy data from distribution panels of a DC that are physically divided for IT equipment and cooling energy consumption. Data from these panels doesn't give us complete information about the DC infrastructure energy consumption, and therefore it is not possible to calculate completely accurate partial PUE values but it gives us good approximations. Examples of the PUE time series of our real-world dataset can be observed in Figure 7 to 9.

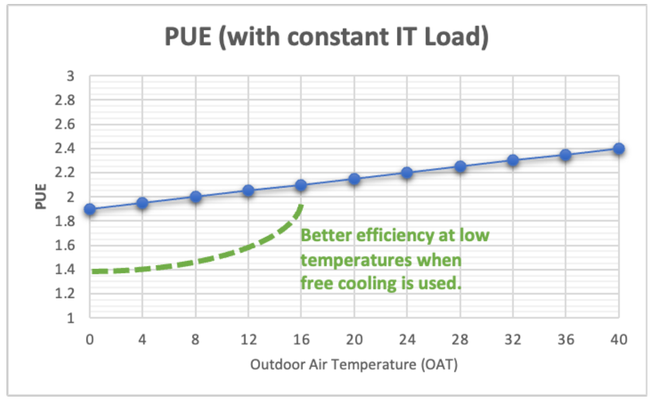

Figure 1: Influence of OAT on PUE ([2]).

2.2 PUE and OAT Dependency

PUE is calculated based on the DC energy consumption and recent estimates state that approximately 43% of the total facility energy is consumed by cooling and power provision systems and the rest by the IT equipment [12]. Since cooling energy consumption plays an important role in calculating the PUE, it is not surprising that PUE values are affected by outside influences, i.e., weather and especially outdoor air temperature (OAT). Figure 1 shows the dependency of PUE on OAT according to [2]. There are several related research works ([15], [5], [16]), [3] where the influence of weather on energy consumption is investigated.

In the scope of this paper, we consider the influence of OAT on PUE as the most important factor. OAT measurements can be either obtained from (DC-related) building management software or from publicly available weather data. Since OAT may change with the seasons, it can take up to 3 quarters of the year to observe the whole spectrum of temperature values. However, in this paper we want to propose an incremental change detection method that can account for OAT influences on PUE using shorter time periods of daily data. Therefore, our proposed method must work with limited data that correctly models the dependencies between OAT and PUE. The influence of OAT on PUE as shown in Figure 1 can be depicted as approximately linear when IT load is relatively constant and advanced features as free cooling are not considered. We were able to confirm this assumption by investigating our real-world DC energy consumption dataset.

2.3 Related Work

A broad corpus of change (point) detection methods can be found in the literature employing different supervised and unsupervised methods [1] or time series analyses including spectral and wavelet analysis [7]. One reason for this vast amount of literature is that the most appropriate change detection method is highly dependent on the actual change detection problem. In our case, we want to detect changes in a PUE time series with daily frequency, but under the consideration of OAT influences and the constraint of limited data. Therefore, we decided to propose a change detection method using a predictive model-based approach. This type of approach relies on the estimation of a predictive model from a change-free dataset

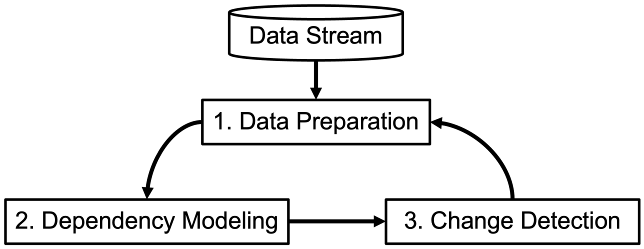

Figure 2: Overview over the main processing steps of the proposed change detection method.

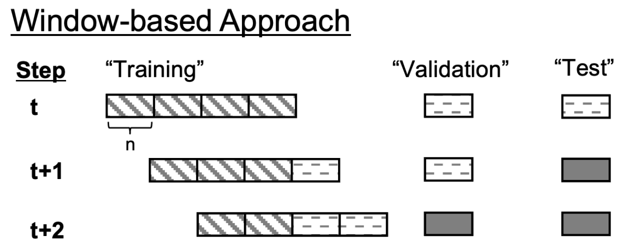

Figure 3: Incremental window-based updates over processing time steps.

and then monitors the discrepancies between model prediction outputs and real observations with a change detection test (CDT) [4]. We found several examples of ML-based PUE prediction in the literature, e.g., [6] and [17], which rely on deep neural networks and large input feature dimensions to predict PUE. However due to the considerations discussed previously, we decided to keep this preliminary study simple assuming an approximately linear relationship between OAT and PUE and using a light weighted linear regression predictive model for the change detection. We assume that change is defined as a sufficiently big gap between two distributions of the OAT to PUE linear relationship. Examples of this will be discussed later in Section 4.

3 METHODOLOGY

We propose a change detection method for DC power efficiency metrics with focus on PUE in this preliminary study. As mentioned in Section 2, PUE is influenced by the weather, e.g. outdoor air temperature (OAT). Therefore, we propose to model the dependency of PUE on OAT with ML predictive models, here linear regression as we assume a linear dependency, and then conduct a change detection test (CDT) based on the prediction error. We design the method to be incremental for continuous change detection. A simple overview of the proposed method is shown in Figure 2 and the details for each step are explained in the following three sections: data preparation in Section 3.1, dependency modeling in Section 3.2, and change detection in Section 3.3.

3.1 Data Preparation

Here, we discuss considerations for data preparation so that the data can be used by our incremental change detection method.

3.1.1 Data grouping. First, we prepare the data so that it is grouped by the spatial granularity of interest (e.g., whole DC, server room, rack-level, etc.) and aggregated to the time frequency of interest (e.g., 15 minutes, 1 hour, 6 hours, 1 day, etc.). For this step, detailed information about the setup of the DC is necessary.

3.1.2 Time Window Handling. Secondly, we split our data into time windows with the window length n. Furthermore, we assign at each processing time step t one or more time windows to one of the following three categories: *training-use*, *validation-use* and *test-use*. The *training-use* windows are used for training the machine learning model (for this paper: a linear regression model). The *test-use* windows are used for calculating the prediction error and then conducting a change detection test. The *validation-use* windows have been observed and tested on change at the previous processing time step and are now used to update the control limits of the change detection test. Then an input feature vector at processing time step t is $\vec{x}_t$ that can be further split into $\vec{x}_{train(t)}$, $\vec{x}_{valid(t)}$ and $\vec{x}_{test(t)}$ respectively. Similarly, an output feature vector $\vec{y}_t$ can be defined. In the scope of this paper, we have decided to assign four windows for *training-use*, one window for *validation-use* and one window for *test-use*. An example of the incremental window updates is shown in Figure 3. Here, we can observe how the differently shaded windows move over the three processing time steps t, $t+1$ and $t+2$ every time we observe a new window of length n.

3.2 Dependency Modeling

We train a ML predictive model to learn the dependencies between PUE and its influencing factor (OAT). As discussed in Section 2, the influence of OAT on PUE is approximately linear when ignoring advanced technologies as free cooling. Therefore, we use linear regression (LR) modeling in this initial study, but other non-linear modeling approaches, e.g., deep learning models or support vector machines, could be tested in the future. Considering that frequent model updates are necessary after change was detected, a light-weight approach like LR modeling also seems preferable.

Another challenge is to model OAT to PUE relationship as accurate as possible with small time periods of data, because the full scale of the temperatures is normally only observed over three quarters of a whole year. Here, we use our previously proposed data differencing step before training the LR model that can better model underlying relationships regardless of an overall trend [3]. Each feature dimensions of the data has sequential measurements $x_1, \ldots, x_s$ that can be differenced by

$$\nabla_m x_s = x_s - x_{s-m}. \quad (2)$$

The difference between a sequential step s and a previous step $s-m$ is calculated, where m is an arbitrary number. The differencing step ensures that the LR model learns how much the value change of the independent input (OAT) between two steps s and $s-m$ has affected the value change of the dependent output (PUE) without involving numerical effects of any baseline trend changes. It therefore also helps to ensure that the training data is change free.

Subsequently, the LR model is trained on the differenced *training-use* window data with OAT as input and PUE as output. Therefore, all model prediction outputs are differenced values, and it is necessary to return them to raw values. To avoid the inclusion of any newly observed information into the model predictions, we use train data averages to return differenced to raw values for the *validation-use* and *test-use* window data. Here, we calculate the average from the latest training time window as baseline values $\overline{x}_{train(t)}, \overline{y}_{train(t)}$. The baseline values can then be used to obtain differenced data for the input vector $\vec{x}_{valid(t),test(t)}$ by calculating

$$\nabla\vec{x}_{valid(t),test(t)} = \vec{x}_{valid(t),test(t)} - \overline{x}_{train(t)}, \tag{3}$$

and return the output prediction $\hat{y}_{valid(t),test(t)}$ to raw values by

$$\hat{y}_{valid(t),test(t)} = \nabla\hat{y}_{valid(t),test(t)} + \overline{y}_{train(t)}. \tag{4}$$

3.3 Change Detection Test

After obtaining PUE predictions $\hat{y}_{valid(t),test(t)}$ for the *validation-use* and *test-use* windows using the LR model described in Section 3.2, we conduct a change detection test (CDT). The CDT is based on the prediction error that is defined by

$$err_{valid(t),test(t)} = \vec{y}_{valid(t),test(t)} - \hat{y}_{valid(t),test(t)}. \tag{5}$$

The prediction error values are then compared to some upper and lower control limits (UCL and LCL) of the CDT to detect potential changes. The UCL and LCL for the current time step t are calculated based on the *validation-use* window by

$$UCL_t, LCL_t = \bar{err}_{valid(t)} \pm k * std_{valid(t)}, \tag{6}$$

where $\bar{err}_{valid(t)}$ is the sample mean and $std_{valid(t)}$ is the sample standard deviation of $err_{valid(t)}$. k is a parameter that defines how many standard deviations from the mean the control limits are set. Here, we only use the *validation-use* window data that previously was observed as test data and where no major changes were detected, therefore making it suitable for control limit updates. This also avoids using *training-use* windows to obtain control limits since training data is often overfitted to the prediction model and can't give accurate information about prediction performance for unseen data.

Furthermore, previous observation of UCL and LCL are incorporated in the control limit update at time step t by calculating the exponentially weighted moving average (EWMA):

$$EWMA_t = \lambda * r_t + (1 - \lambda) * EWMA_{t-1}. \tag{7}$$

Here, the current value r_t would be either UCL_t or LCL_t and the previous EWMA at time step $t-1$ would be denoted by $UCL_{EWMA_{t-1}}$ or $LCL_{EWMA_{t-1}}$. We then newly obtain the two control limits UCL_{EWMA_t} and LCL_{EWMA_t}. λ is a parameter that defines the proportion of forgetting old information. With this additional step we ensure incremental updates of the two control limits.

The previously calculated prediction error vector for the *test-use* window $err_{test(t)}$ is then compared to UCL_{EWMA_t} and LCL_{EWMA_t} by observing the following conditions

$$err_{test(t)} > UCL_{EWMA_t}, \tag{8}$$

$$err_{test(t)} < LCL_{EWMA_t}. \tag{9}$$

The goal is to detect a change when a certain amount of error values are above or below the control limits. To avoid a high sensibility to shorter fluctuations in the data or other outliers, the total amount of error values exceeding the control limits should exceed a greater percentage of the whole window length. This could be for example half the window size, i.e. 50% of the prediction errors are exceeding the control limits.

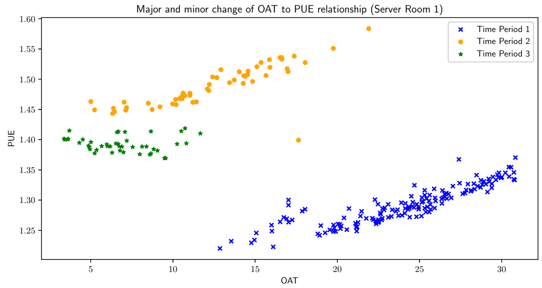

Figure 4: Observable changes in distribution with big and small gap.

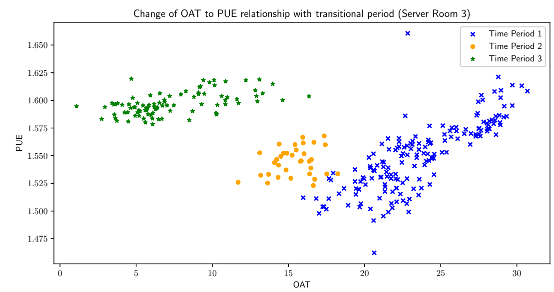

Figure 5: Observable changes in distribution with clean transitional period.

4 EXPERIMENTAL RESULTS

We conducted some preliminary experiments for our proposed method using real-world DC data. The observed data is from a multi-tenant DC with air-based cooling where energy consumption can be observed at the distribution panel level. In this DC, the power distribution panels are mostly shared by several server rooms and therefore we aggregate DC energy consumption data into "server room groups" in the spatial context and to a daily sampling frequency in the temporal context. In the scope of this experiment, we use data from three "server room groups" that will be called "server room 1", "server room 2" and "server room 3" for simplicity. Since the real-world data has no readily available labels for observed changes, we manually investigated the data and labeled a few change points by hand. We then evaluate the proposed method on whether it can detect the manually labeled changes. The results are discussed in the following sections.

4.1 Manually labeled changes

First, we explain how we manually detected changes and labeled them. Here, we observe the data distribution in a two-dimensional space by plotting the OAT against PUE for smaller time periods of several weeks to several months with different color schemes

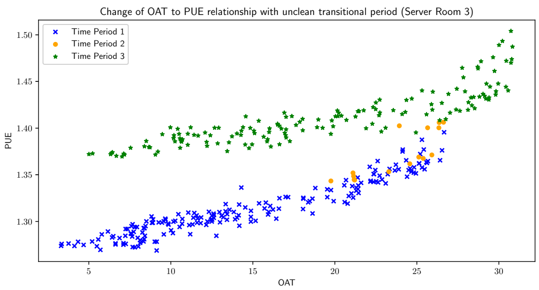

Figure 6: Observable changes in distribution with unclean transitional period.

Table 1: Parameter settings for the change detection method.

Parameter	Symbol	Value
Window size	n	15 days
Control limit deviation	k	3
EWMA updates	λ	0.1

and see if there are distinctive changes in the distribution so that these are separable by time. An example of a large sudden change in distribution that is cleanly separable by time is shown in Figure 4 between time period 1 and time period 2, 3 . For this plot we can also observe that there is a second smaller change from time period 2 to time period 3 (both are spanning less than 2 months in time). A second example of a change in distribution with a clean transitional period is show in Figure 5, where time period 2 is a transitional period. A third example in Figure 6 shows that these transitional periods are not always clean, because during the two weeks long time period 2, the data distribution is not cleanly separable between the distributions of time period 1 and 3. It will be interesting to see in the experiments whether our proposed change detection method is able to accommodate such use cases or not.

4.2 Change Detection Results

We have manually identified change points for three server rooms as explained in Section 4.1 and will compare these to the automatically detected change results obtained with our proposed method. The parameter settings used for our proposed change detection method are mentioned in Table 1. For each server room, we visually present the automatic change detection results in Figures 7-9 with the actual observed PUE in black, the predicted PUE for the *test-use* window in orange and all time periods with detected change in red.

4.2.1 Server Room 1. As shown in Figure 7, server room 1 observes a major change in PUE twice, first the PUE worsens in October 2020 and then improves again in October 2021. Here, we can assume that the DC operator might have changed some operational settings causing the major change in energy efficiency, i.e., PUE, but somehow this went unnoticed. An automatic change detection system could support DC operators here to be more sustainable.

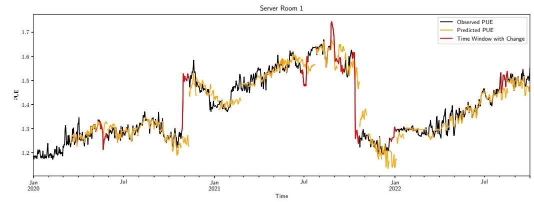

Figure 7: Observed PUE and automatically detected changes in server room 1.

Table 2: Manually detected changes for Server Room 1.

No.	Change Date	Comments
0	2020-03-05	Minor change
1	2020-05-23	Minor change
2	2020-10-29	Major change
3	2020-12-22	
4	2021-02-04	
5	2021-06-24	Minor change
6	2021-07-06	Same distribution as before 2021-06-24
7	2021-10-13	Major change
8	2021-12-21	
9	2022-02-27	
10	2022-08-02	

Our proposed method was able to identify these major changes in the PUE as well as more subtle changes at other time periods.

Table 2 shows the manually detected changes for server room 1. In total, we manually detected ten changes in the distribution, where change no. 5 is a very minor change to a new distribution followed by change no. 6 back to the old distribution of no. 4 after 2021-07-06. Also, we added a minor change no. 0 that is excluded from the detection scope since it falls into the very beginning of the observations and is not considered in any test window. It might however influence the initially learned control limits negatively as it falls into our first validation window causing UCL and LCL to be set to more loose limits, and failing to detect change no. 1. On the other hand, the detection method was able to detect changes no. 2, 4, 5, 6, 7, 8 and 10. The recall for this experiment is therefore 70%. In addition, the method detected a false positive change in September 2021, which we did not identify with a change label manually. After rechecking the data, we found a minor change in distribution for a time period of 6 days at the very end of August 2021. It is likely possible that this change might have triggered the change detection method belatedly to falsely detect the change in September 2021 but we are still considering it a false positive.

4.2.2 Server Room 2. As shown in Figure 8, the PUE of server room 2 is very stable and shows no signs of drastic increase or decrease over time. We can observe here that the PUE is increasing and decreasing in correlation with the OAT, since this DC is placed in a region where it is hot in summer and cold in winter. Nevertheless, there are three time periods where change is detected by our proposed method. This compares to the three manually observed

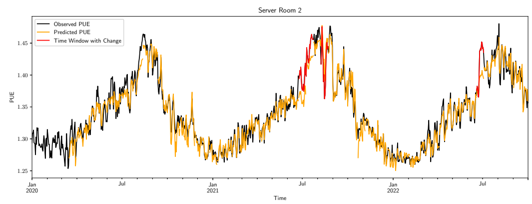

Figure 8: Observed PUE and automatically detected changes in server room 2.

Table 3: Manually detected changes for server room 2.

No.	Change Date	Comments
1	2021-06-22	
2	2021-10-01	
3	2022-06-24	very minor change

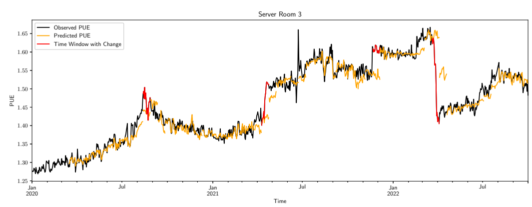

Figure 9: Observed PUE and automatically detected changes in server room 3.

changes that are stated in Table 3. When we compare the change detection results, we observe that our proposed method was able to detect two out of three changes correctly. Therefore, the recall rate is 66.7%. Only change no. 2 was not detected but instead another change was detected earlier in time around August 2021. Even after revisiting the data, we were not able to confirm any change in August 2021. Therefore, we assume that the automatic change detection method might have difficulties to recover after the recently detected change no. 1 but further investigation is necessary. For the false negative change no. 2, we can observe from Figure 8 that the prediction error was not large enough to trigger the UCL or LCL.

4.2.3 Server Room 3. As shown in Figure 9, the PUE of server room 3 is changing very drastically over time. Still our proposed change method can detect changes of meaning without finding any false positives. Compared to the manually detected changes that are stated in Table 4, the proposed method was able to detect change no. 1, 2, 4 and 6. That is a recall rate of 57%. For the three changes no. 3, 5 and 7 that were not detected, we can observe a deterioration of prediction performance in Figure 9, but apparently it was not enough to trigger the control limits. We believe that these changes could also be detected after further parameter tuning and additional method improvements to better handle model and control limit updates after changes are detected.

Table 4: Manually detected changes for server room 3.

No.	Change Date	Comments
1	2020-07-12	transition period until 2020-07-26
2	2021-04-16	
3	2021-10-17	
4	2021-11-20	
5	2022-02-18	
6	2022-03-26	first 4 days are on different scale
7	2022-07-06	transition period until 2022-07-13

4.3 Discussion

The experimental results for our three server rooms show that the proposed method can detect many of the changes accurately with just very few cases of false positives. The proposed method was able to detect all major and some of the minor PUE changes. In all experiments, we have observed some false negatives where the proposed method was not able to detect changes even when deterioration in the prediction performance could be observed. This might be caused by differing overall prediction quality that affects the control limits to be more loose or stricter. Therefore, the observed recall rates are still quite low, but on the other hand the experimental sample size is also quite small and further investigations in additional experiments is necessary. In addition, the sensibility of the change detection method should be investigated, and sensibility guidelines could be provided in the future, especially regarding parameter tuning. Other areas of improvements are the handling of the model training and control limit updates after a change is detected. For the practical use of our method, it is necessary to establish fast recovery after change detection. We believe that our current approach that relies on model training of differenced data and baseline values is a first step in the right direction, but further improvements are necessary.

5 CONCLUSION AND FUTURE WORK

In this paper, we proposed a change detection method for a DC energy efficiency metrics, i.e., PUE. With this method, we observe PUE values over shorter time periods with daily sampling frequency and then automatically detect important changes in the DC energy efficiency while considering outdoor air temperature (OAT) influences. The proposed method was tested in preliminary experiments with real-world DC energy consumption data at a "server room group" aggregation level and the results show that our method can detect most of the important PUE changes in the data but further improvements regarding recall quality are necessary. We believe that this method will be helpful to provide feedback to DC operators especially after operational setups in the DC have been changed, since not all short-term energy efficiency changes can be obviously observed from time series data as other influences as for example OAT affect it.

In the scope of this paper, we concentrated on a simple real-world DC use case without advanced technologies as e.g., free cooling, and considerations for extended use cases remains future work. Furthermore, the following list shows other future topics we want to work on to improve our method:

- Extended experiments with more data and improving recall quality.
- Sensibility study of change detection method and tunable parameters to provide user guidance.
- Application of method to other efficiency metrics.
- Testing other predictive non-linear ML models.

REFERENCES

[1] Samaneh Aminikhanghahi and Diane Cook. 2017. A Survey of Methods for Time Series Change Point Detection. *Knowledge and Information Systems* 51 (05 2017). https://doi.org/10.1007/s10115-016-0987-z

[2] Víctor Avelar, Dan Azevedo, and Alan French. 2012. PUE™: A COMPREHENSIVE EXAMINATION OF THE METRIC.

[3] Jana Backhus and Yasutaka Kono. 2022. Cooling Power Consumption Dependency Simulation for Real-World Data Center Data. In *30th International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2022, Split, Croatia, September 22-24, 2022*. IEEE, 1–6. https://ieeexplore.ieee.org/document/9911308

[4] Giacomo Boracchi and Manuel Roveri. 2014. Exploiting self-similarity for change detection. In *2014 International Joint Conference on Neural Networks (IJCNN)*. 3339–3346. https://doi.org/10.1109/IJCNN.2014.6889860

[5] Chengliang Fan, Yundan Liao, and Yunfei Ding. 2019. Development of a cooling load prediction model for air-conditioning system control of office buildings. *International Journal of Low-Carbon Technologies* 14, 1 (01 2019), 70–75.

[6] Jim Gao. 2014. Machine Learning Applications for Data Center Optimization. (2014).

[7] Ebrahim Ghaderpour, Spiros Pagiatakis, and Quazi Hassan. 2021. A Survey on Change Detection and Time Series Analysis with Applications. *Applied Sciences* 11 (07 2021), 6141. https://doi.org/10.3390/app11136141

[8] Google. 2021. *Google Data Center PUE performance.* https://www.google.com/about/datacenters/efficiency/

[9] IEA. 2021. *Data Centres and Data Transmission Networks.* https://www.iea.org/reports/data-centres-and-data-transmission-networks

[10] Uptime Institute. 2020. *Data center PUEs flat since 2013.* https://journal.uptimeinstitute.com/data-center-pues-flat-since-2013/

[11] William Lintner, Bill Tschudi, and Otto VanGeet. 2011. Best Practices Guide for Energy-Efficient Data Center Design: Revised March 2011 (Brochure). (3 2011). https://doi.org/10.2172/972929

[12] Eric Masanet and Nuoa Lei. 2020. *How Much Energy Do Data Centers Really Use?* https://energyinnovation.org/2020/03/17/how-much-energy-do-data-centers-really-use/

[13] Paul Mathew, Srirupa Ganguly, Steve Greenberg, and Dale Sartor. 2009. Self-benchmarking Guide for Data Centers: Metrics, Benchmarks, Actions. (01 2009). https://doi.org/10.2172/983248

[14] Patrick Pegus, Benoy Varghese, Tian Guo, David Irwin, Prashant Shenoy, Anirban Mahanti, James Culbert, John Goodhue, and Chris Hill. 2016. Analyzing the Efficiency of a Green University Data Center. In *Proceedings of the 7th ACM/SPEC on International Conference on Performance Engineering* (Delft, The Netherlands) *(ICPE '16)*. Association for Computing Machinery, New York, NY, USA, 63–73. https://doi.org/10.1145/2851553.2851557

[15] Georgios Smpokos, Mohamed A. Elshatshat, Athanasios Lioumpas, and Ilias Iliopoulos. 2018. On the Energy Consumption Forecasting of Data Centers Based on Weather Conditions: Remote Sensing and Machine Learning Approach. In *2018 11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*. 1–6.

[16] Qi Zeng and et al. 2016. An optimum regression approach for analyzing weather influence on the energy consumption. In *2016 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*. 1–6.

[17] Peng Zhao, Lina Yang, Zong Kang, and Jie Lin. 2019. On Predicting the PUE with Gated Recurrent Unit in Data Centers. In *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. 1664–1670. https://doi.org/10.1109/ICCC47050.2019.9064040