Context Aware Monitoring for Smart Grids

Daniel Hauer[a,b], Maximilian Götzinger[a,c], Axel Jantsch[a], Florian Kintzler[b]
[a]Institute of Computer Technology, TU Wien, Vienna, Austria
E-mail: {daniel.hauer, maximilian.goetzinger, axel.jantsch}@tuwien.ac.at
[b]Siemens AG, Vienna, Austria
E-mail: {daniel.hauer, florian.kintzler}@siemens.com
[c]Department of Computing, University of Turku, Turku, Finland

***Abstract*—Today's energy grids face an increasing number of decentralized and renewable energy sources as well as growing e-mobility. Therefore, reliable grid monitoring becomes a key element for a sustainable grid operation. Traditional grid monitoring concepts are either fully manual, need a detailed system model, or rely on computationally heavy machine learning concepts. However, with the given complexity of the energy grid, a model-free and context-aware monitoring approach can save resources and efforts. Recently, we introduced the Confidence-based Context-Aware Condition Monitoring (CCAM) system and successfully tested it on two different industrial use-cases: a hydraulic circuit and an AC motor. In this paper, we enhance CCAM for a third, entirely different industrial use case, an energy grid, by introducing two extensions - a continuous reevaluation and a state mooring approach. Furthermore, we present a new Smart Grid monitoring methodology on top of CCAM, paving the way for new real-time grid control systems. We evaluate our approach based on historical load data from a low voltage grid section. Our results show that characteristics of a daily load profile can be learned and outliers can be detected.**

***Index Terms*—Smart grids, Context awareness, Monitoring**

I. Introduction

The increase of diversity of distributed energy producers and consumers poses a great challenge for the future energy grid. The generation of renewable energy as well as e-mobility or demand-response applications require intelligent monitoring- and controlling concepts.

In general, it is expected that grid infrastructures will have to be operated closer to the limits to keep the level of new investment within an economically justifiable range. This approach follows the vision of Smart Grids, which refers to the electric power grid's modernization by introducing an intelligent bidirectional flow of energy and information [1]. Nevertheless, systems need to remain safe and secure in the advent of faults and threats, which could even be unpredictable at design-time or emerge during run-time [2]. Failing to address such issues in time results in supply outages, and the associated costs continually increase. First negative impacts can already be observed regarding the costs of congestion management. In Austria, those costs in August 2019 were almost equal to the costs for the entire year 2016 [3].

The future grid operation will and already does rely on a vast and heterogeneous data set from various sensors. Therefore, classic (threshold-based) monitoring is no longer sufficient for detecting anomalies and important events to prevent faults. While machine learning techniques based on Artificial Neural Networks (ANNs) constitute possible solutions, they are resource-intensive. However, edge and Internet of Things (IoT) devices require small footprint solutions.

In a recent work [4] we proposed the Confidence-based Context-Aware Condition Monitoring (CCAM) system, a black-box condition monitor system for detecting system states. It is based on the principles of self-awareness [5]–[7], and its little computational power footprint makes it a good match for on-site usage in grid monitoring devices. However, the complex behavior of energy grids leads to the necessity of further enhancing CCAM's signal state detection algorithm. This paper, therefore, presents a novel Smart Grid monitoring methodology for low voltage grids to be able to handle the growing amount and complexity of existing data.

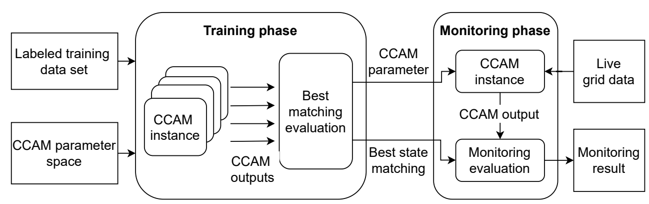

Fig. 1. Concept of the proposed Smart Grid monitoring methodology

Figure 1 gives an overview of our approach. During a training phase, multiple CCAM instances with different parameters are evaluated based on a training data set (e.g., favored daily load profile). As a result, the best CCAM parameter set as well as the best state matching are calculated and used for the monitoring phase. From now on, live grid data can be monitored by the single CCAM instance, and every new label under investigation (e.g., a *day*) will be evaluated. This enables the grid operator to detect anomalous behavior (e.g., a maintenance event) of specific grid segments among numerous others, enabling fast and targeted intervention. The key contributions of our work are:

(i) Enhancing CCAM with a continuous reevaluation concept to optimize the state detection,
(ii) Introducing a state mooring approach, which gives CCAM the ability to remember past states and prevents slow state drifts over time,
(iii) Presenting a novel Smart Grid monitoring methodology on top of CCAM and verify our approach based on

978-1-7281-9023-5/21/$31.00 ©2021 IEEE

historical data within a low voltage distribution substation of the Aspern[1] testbed in Vienna, Austria.

The rest of this paper is organized as follows: After a review of related work in Section II, Section III describes the proposed CCAM enhancements as well as the Smart Grid monitoring methodology. Subsequently, Section IV introduces the Smart Grid case study and provides results. Finally, Section V shows open directions for further research.

II. Related Work

A. Smart Grid Monitoring

Today, grid monitoring mostly depends on threshold-based protection systems (e.g., fuses) and human-centered supervision via so-called Supervisory Control and Data Acquisition (SCADA) systems [8]. These concepts are well suited for the classic grid operation with a unidirectional energy flow from power plants to consumers. However, they do not scale for the growing diversity of energy producers and consumers, along with a highly unpredictable bidirectional energy flow and an increasing number of sensors and heterogeneous data [2].

Smart Grid monitoring, therefore, has to evolve from a human-centered system to an intelligent and self-aware one, which can deal with a large number of sensors with different characteristics. It is suggestive that from 2010 to 2019 72% of Smart Grid-related machine learning research was published in the last three years [9]. While these approaches often depend on large training sets and are computationally heavy, solutions with a smaller footprint and without the need for complex system models could pave the way for new monitoring systems. Recent works state the need and challenges for context-aware (or more general self-aware) Smart Grid monitoring but lack concrete implementations and results (see [10]–[12]). This gap is closed by the presented work.

B. Black-Box Monitoring

Like in industrial applications where it is neither economical nor feasible to implement a monitoring unit for each system or machine from scratch, the multitude of different sections in a Smart Grid would also benefit from a generic solution.

While ANN-based monitoring methods (such as [13], [14]) are, as already mentioned, mostly computationally heavy and depend on a vast amount of training data, also other solutions exist. Guo *et al.* [15] used a Hidden Markov Model (HMM) to find probabilistic relationships between variables of interest. Their monitoring system could detect most of the events, but all of them had to be trained to the HMM in advance. In contrast, fuzzy logic-based condition monitoring requires less computing power, can model nonlinear behavior and, if desired, can be extended with expert knowledge [16]. On the other hand, their results are often less precise, and rule-based descriptions are cumbersome to develop.

Another example of such black-box condition monitoring systems is CCAM, introduced in one of our previous papers [4]. It proved its condition monitoring capabilities when monitoring systems of different types. Besides, we showed that it is executable on different ARM-based Embedded Systems (ESs) in real-time [17]. For these purposes, we further explore this system in this work.

[1]https://www.ascr.at

III. Smart Grid monitoring methodology

CCAM was initially designed to monitor a system which can be represent by a bijective function. Its main goal was to detect different states within the monitored in- and output signals and observe whether the system works as expected, drifts, or malfunctions. As shown in Figure 1, we introduce a novel Smart Grid monitoring methodology on top of CCAM. By using CCAM solely as a state detector and combining it with a training and monitoring concept, we enable monitoring of more complex systems, such as a Smart Grid. We will give a short overview of the original CCAM system and its proposed enhancement in subsection III-A and present our monitoring methodology in subsection III-B.

A. Confidence-based Context-Aware Condition Monitoring

CCAM was developed to provide a generic method to monitor a black-box system for determining in which working state it is and whether it functions correctly or malfunctions [4]. For this purpose, CCAM observes the system's in- and outputs (Figure 2) to determine in what states the corresponding signals are. CCAM avoids a large computational footprint by making all decisions using confidence values and only contextual knowledge. CCAM was successfully applied to two different use cases [4]: an AC motor and a water pipe system.

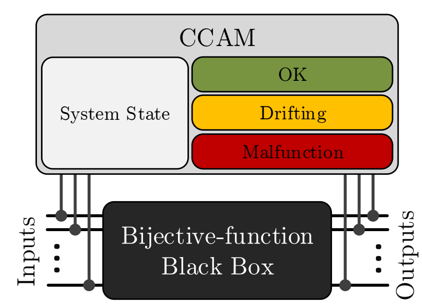

Fig. 2. Block diagram of the Confidence-based Context-Aware Condition Monitoring (CCAM) system

Because CCAM works on the principles of self-awareness and benefits from a hierarchical agent-based architecture, it was implemented in the Research on Self-Awareness (RoSA) framework, proposed in the work [17]. Roughly speaking, CCAM consists of one signal state detector for each monitored signal and one additional system state detector. The system state detector abstracts the system's operation based on all present signal states. Moreover, CCAM detects whether the system is working correctly, based on the assumption that the observed system behaves like a bijective function, which means that one single input data set corresponds exactly to one single output data set and vice versa. Thus, if one or more inputs change their states, at least one output also must change its state.

Due to the modular design of CCAM and its versatile modules, it can also be used for load profile monitoring in a Smart Grid. In the use case presented in this work, just one signal is monitored. Thus, the bijective condition monitoring is disabled. Here, CCAM's task is to recognize patterns in the power grid. For instance, considering the power measurement of a certain power line, CCAM can detect local behavior such as time- and date-dependent consumer patterns (e.g., evening peaks) or unusual operation modes (e.g., maintenance event at a battery storage system).

Due to the high complexity of a Smart Grid's load profile, with both slow and fast changes, a multitude of participants, and high noise, CCAM's algorithm needs to be extended. Thus, we enhance the CCAM system by introducing a continuous state reevaluation as well as a state mooring concept.

1) Continuous State Reevaluation: Within the original algorithm of CCAM [4], the currently active signal state is preferred over any other existing state. Only if the sample does not fit the active state anymore, it gets compared to all other existing states, and the first matching one is selected.

We propose to enhance this state detection process by a continuous evaluation over all existing states. Figure 3 (shaded area) visualizes the new approach in which each new sample is compared with all existing states to find the best matching. Although the computational effort grows linearly with the number of states, it is moderate and unproblematic for the application cases we considered.

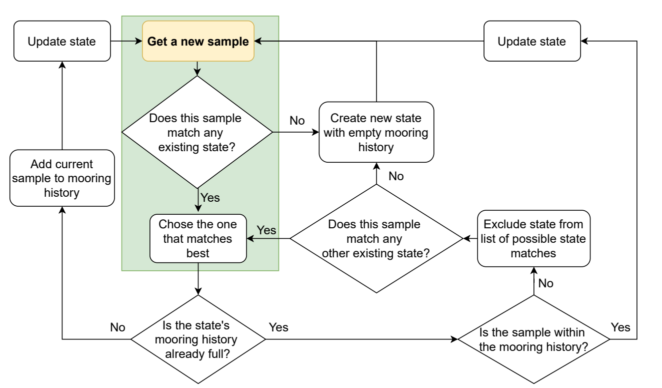

Fig. 3. Flow Chart of our continuous reevaluation (shaded area) and state mooring approach

2) State Mooring: CCAM compares every new signal sample with the sample history of all already existing states. However, as the history has size sh, CCAM does not remember all historical values of a state but only the last ones when choosing the best matching state for the new sample. Thus, the original CCAM system cannot interpret a slow change in a signal as a state change but would stay within the same state and eventually raise a drift alarm. This is a valid approach for stable systems, which have defined and step-wise state transitions (e.g., a motor that changes from one mode to another), but does not generate useful results for more fluctuating and complex systems.

We therefore propose a state mooring approach, which allows CCAM to remember the entry of a state even after the current sample history window has deleted those values. In particular, we replace the drift detection with a new mooring concept, also shown in Figure 3. Every signal state additionally stores a mooring history of size mh, which holds the state's initial sample values.

When a new signal state is created, its mooring history is empty and does not influence the state matching calculation of CCAM. Every new sample value is added to the state's mooring history until it is full (has mh entries). Once the mooring history is full, it characterizes the state's initial behavior and can be used to compare the evolution of the state over time. From now on, every new sample also has to match this history and not just the current sample history which might already be slowly drifting. This is done by adding an additional step to the state matching approach of CCAM. Every time the confidence that a sample matches a particular state is calculated, the new sample value is also compared to the mooring history's mean value. If the distance between those two values is too large, the matching confidence is set to zero. In other words, if the current state has slowly drifted away from its initial behavior and is not close enough to its mooring history, the matching confidence declines, and CCAM will either find another matching state or create a new one. Because we use the same fuzzy functions for the mooring history comparison as for the original state detection (see [4] for detailed information), we avoid the introduction of a new independent parameter and follow the original CCAM concept.

B. Monitoring methodology

While the already described modifications of CCAM improve the state detection (see subsection IV-B), an additional evaluation step on top of CCAM introduces a new concept for system monitoring. The approach described in the following also helps to overcome the difficulty of choosing suitable CCAM parameters for the given system under investigation. Although CCAM does not need to have apriori knowledge about the monitored system, the system's characteristics still influence the choice of parameter values.

As shown in Figure 1, we distinguish a training and monitoring phase:

1) Training Phase: At the beginning a time series for the training data set $\{x_t \in \mathbb{R}^n : t = 1, 2, ..\}$ has to be defined. This data set can either be a historical time series of the system under investigation or a defined period for real-time training. During the training phase, multiple CCAM instances with different parameter settings simultaneously process the same time series. Each run r creates an individual CCAM output, including the detected state ID $\{os_{t,r} \in S_i : t = 1, 2, ..\}$ (with state IDs S_i) as well as the confidence about this decision for every timestamp $\{oc_{t,r} \in \mathbb{R} : t = 1, 2, ..\}$. This training step could also be done off-line with historical data on an appropriate server.

After the training runs are finished, the data set has to be labeled manually according to the targeted monitoring approach with m different labels L_m resulting in a time series of labels $\{la_t \in L_m : t = 1, 2, ..\}$. This can, for example, be a daytime-based label such as *day* and *night*. Our current approach is limited to a single label per timestamp as well as continuous labeling for all sample points. However, if a specific event is of interest, the labeling could be done based on its occurrence (labels L_m: *event* and *regular*).

Using the labeled data set, the training phase now calculates the best matching state-to-label assignment and the corresponding state distribution for all training runs. This procedure assumes that every label can be defined by a specific set of CCAM states. The finite combination set C represents all possible state-to-label combinations c for a given number of states and labels (s, l). Now, the best matching combination in C for one run r is calculated according to Algorithm 1 with the following resulting parameters:

- $cl_{c,r}$, the confidence (0 .. 1) that this combination c is valid for the given training data set and run r.
- $co_{c,r}$, the confidence (0 .. 1) that additionally multiplies $cl_{c,r}$ with the percentage of valid states within the training data sets and therefore represents the overall confidence whether this is a valid state-to-label combination and a suitable CCAM parameter set.
- $d_{s,l,c,r}$, the percentage distribution of state s ($\in S_i$) within label l ($\in L_m$) (e.g., the state with ID 1 has an occurrence of 22% within the label *day*)

Algorithm 1 Finding the best combination in C for run r

Input: x_t, $os_{t,r}$, $oc_{t,r}$, L_m, la_t
Output: $cl_{c,r}$, $co_{c,r}$, $d_{s,l,c,r}$
for every state-to-label combination c in C **do**
  $cnt, labelcnt_l, statecnt_{l,os_{t,r}} = 0$
  for every timestamp, t **do**
    if $oc_{t,r} > 0.5$ **then**
      $cnt++$
      for every label l in L_m **do**
        if $l = la_t$ and $(la_t, os_{t,r})$ in c **then**
          $labelcnt_l++$
          $statecnt_{l,os_{t,r}}++$
  $confeval = cnt/len(x_t)$
  $cl_{c,r} = (\sum_l labelcnt_l)/cnt$
  $co_{c,r} = cl_{c,r} * confeval$
  for every label l in L_m **do**
    for every state s in $os_{t,r}$ **do**
      $d_{s,l,c,r} = statecnt_{l,s}/labelcnt_l$

After all runs have been evaluated, the one with the highest overall confidence $co_{c,r}$ provides the best CCAM parameter set r and the corresponding state-to-label assignment c.

2) Monitoring Phase: The results obtained from the training phase can then be used to monitor the operating system. From now on the signals of the system under investigation are fed into a single CCAM instance using the determined parameter set r. The sample points have to be additionally labeled, and every time the label changes (e.g., *day-night* change), the past time series (e.g., past day) is evaluated based on the training results (i.e. best parameter set r and state-to-label assignment c). The evaluation is done as follows:

- The distributions for all occurring states are calculated. The distribution cd_s of one state s is defined by the proportion between the number of data-points being assigned to the valid state s and the total number of data-points with valid states.
- Then, all distances between the distributions cd_s and the distributions from the training data set $d_{s,l,c,r}$ are calculated for the currently evaluated label l. If a state is not present in the training data, it is skipped.
- An overall evaluation result oe (0 ..1) is calculated by the mean of all distribution distances.

The overall evaluation result oe represents the matching of the currently analyzed time series and the training data set. It, therefore, indicates whether the monitored system is operating as expected (according to the trained label) or not. The value oe represents the proximity to the training data and can be used to detect outliers and unusual behavior within the current system behavior. The next section will evaluate this concept based on a Smart Grid use case to verify this approach.

IV. Smart Grid Case Study

For our case study, we are using the active power values from a low voltage distribution substation of the Aspern testbed in Vienna, Austria[1]. The grid parameters were recorded over a period of multiple years using grid monitoring devices. Specifically, we consider a time-series recorded for one selected substation transformer supported by a battery storage system, which prevents daily load peaks by intelligent charging and discharging. The battery is charged overnight and, during the daytime, it limits the grid load of each phase to 60 kW (peak-shaving). While this is the desired behavior, an analysis has shown that this behavior cannot be observed continuously over the whole year. Instead, the investigated data set can be divided into four different daily categories. Figure 4 gives an overview of the corresponding load profiles.

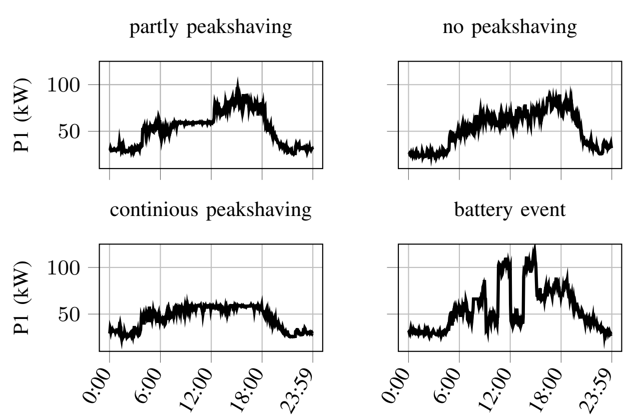

Fig. 4. Load day profiles of the investigated substation

While peak-shaving should be active over the whole day, the majority of the observed days are only partly peak-shaved, and for some periods, there is no peak-shaving at all. In addition, a few battery maintenance events occur, as this testbed was under continuous extension. As the correct operation of the battery storage is essential for a safe grid operation (to avoid overloads) and for optimized load distribution, information about the current situation is required for the grid operators.

Therefore, we use our proposed methodology to monitor the load of the transformer under investigation. Furthermore we train CCAM based on the identified daily load profiles. Thus we can evaluate every new day according to these profiles. This evaluation can then serve as information for the grid operator to distinguish correct operation from anomalies.

A. Experimental Setup

The evaluation is based on historical data from one specific transformer substation from January 1 to December 31, 2018, with a sample rate of 2.5 min. For our experiments we used the power flow measured on phase 1 ($P1$) as single CCAM input. The parameter space of CCAM was selected according to Table I.

TABLE I
CCAM PARAMETER RANGE

Parameter	Range	Steps
Inner Bound ib	[5%, 60%]	2%
Outer Bound ob	ib + [1%, 30%]	2%
Sample History size sh	[5, 30]	5
Down-sampling rate ds	[1, 7]	2
Mooring history size mh	[5, 15]	5

The *Inner-* and *Outer Bound* represent the borders of the fuzzy functions for the confidence calculations according to [4]. The *Sample History size* specifies the window size for the stored samples taken into account for the state detection of every new sample. The *Down-sampling rate* is used to modify the sampling rate of the time series (e.g., only take every third value). While the drift and malfunction functions of CCAM are turned off, the new mooring history is introduced.

Furthermore, we introduce the labels *day* and *night* ($L_m = \{day, night\}$), where a day starts at 4 am and ends at 8 pm. These timestamps were chosen based on the observed profiles — according to the morning load peak and evening drop. This evaluation aimed to train our monitoring system based on selected days with one specific day profile (Figure 4) and to evaluate the following days based on this training. Therefore, we selected seven days with a "partly peak-shaving" profile during September. This profile represents the typical operation of this grid segment, in which peak-shaving is turned on but is not active over an entire day because of a too small battery storage. Due to the goal of finding anomalies to the typical behavior, we chose this profile. As a result we get an overall evaluation for every new day, how well this day matches the trained ones. In accordance with subsection III-B, we trained our algorithm using those days. The result of this process is the best CCAM parameter set as well as the best state-to-label assignment to those two labels.

B. Results

Before we evaluate the final results, Figures 5 and 6 show the output of the original CCAM and the enhanced one (with continuous reevaluation and state mooring), respectively, using the same parameter set. Besides the original load profile, the detected State ID is plotted if CCAM is confident that a state is currently valid. The following improvements can be seen:

- *Continuous Reevaluation*: During the first night, a short peak leads to a state change from State ID 1 to 2. The original CCAM stays with this new ID, although the signal falls back to similar values as before the peak. In contrast, the continuous reevaluation improves this behavior as CCAM immediately changes back to State ID 1 as the old state matches better than the new one.
- *State Mooring*: While the original algorithm detects the same state (ID 4) before and after the battery event due to the slow drift of the sample history, our new approach distinguishes between multiple states (mostly IDs 3 and 5), as the values are not close enough to each other.
- *Combination of both*: While the original algorithm continuously introduces new states during the evening and often stays with them although the states are drifting, the two enhancements together lead to a better state separation as well as the change back to already known states.

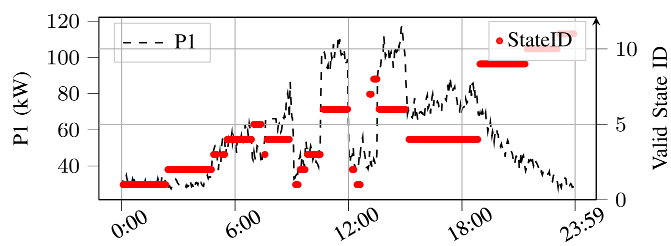

Fig. 5. Original CCAM result (ib = 19%; ob = 24%; sh = 5; ds = 1)

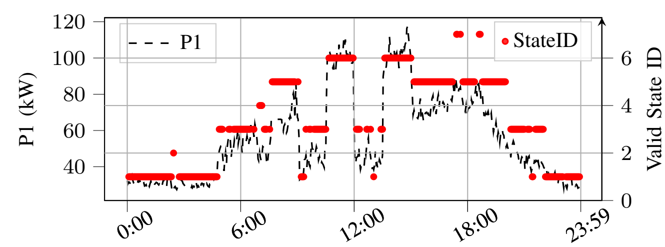

Fig. 6. New CCAM result (ib = 19%; ob = 24%; sh = 5; ds = 1; mh = 10)

As the two enhancements influence the state detection algorithm of CCAM, we also verified that this does not change its originally intended behavior. Therefore, we applied the new version to the AC motor case study from [4] and got equally good results with the exact same state detection.

For the Smart Grid monitoring setup described in subsection IV-A, the best CCAM parameter set is ib = 45%, ob = 1%, sh = 5, ds = 1 and mh = 10. Using these training results, we then monitored the subsequent 14 weeks. As a result, every day (more general, every label that has been trained) is evaluated, and an overall evaluation metric (oe) indicates whether the observed value progressed expected during the day.

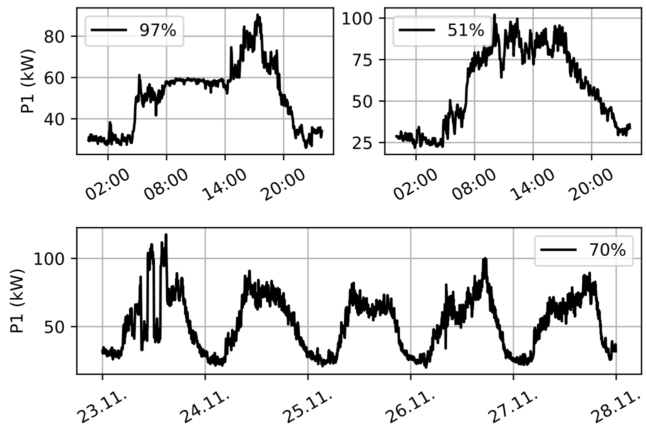

Fig. 7. Evaluation results: Upper left: best matching day (97%); Upper right: worst matching day (51%); Lower area: Battery event (70%) and following days.

Figure 7, finally exemplary, shows classified measurements of selected relevant days. The upper left day nearly perfectly matches the trained data set. In contrast, the upper right day has the lowest matching value because no peak-shaving is active and the load is way higher than usual. In the lower area of the figure, the battery event and the following days can be seen. All those days have a low matching value. Furthermore, a second maintenance event similar to the one in Figure 7 was detected on December 11 with oe = 74%. After this event, the peak-shaving was turned off for the rest of the year. Out of those 20 days, only eight had an $oe > 80\%$. Interestingly, seven of them can directly be related to a Sunday or Christmas holiday, suggesting that the load consumption during those days is low. Therefore, CCAM cannot differ whether it is low due to active peak-shaving or general low consumption. This is a reasonable result as the goal was to detect anomalous behavior and not to find the underlying cause of a particular daily load profile.

V. Conclusion and Outlook

In this paper, we have shown a novel Smart Grid monitoring methodology — a combination of the context-aware black-box monitoring system, CCAM, which we have enhanced with two new features, combined with a new evaluation concept based on training and monitoring. By finding a suitable CCAM parameter set and an appropriate state matching for given labels during a training phase, our approach can analyze new and live data streams and evaluate their similarity to the training data while still having a lightweight footprint compared to neural network-based ML algorithms. We have verified our approach by analyzing historical load profiles from a transformer substation within a testbed and could detect outliers such as a maintenance event. The output of our Smart Grid monitoring is useful for grid operators to identify potential anomalous behavior as support for preventive maintenance and during operation.

As the results are promising, we are working on the inclusion of CCAM into Smart Grid simulation environments to get more useful data and further enhance our methodology. For example, the training phase could be improved by dynamically adapting and changing the CCAM parameter space (e.g., with evolutionary algorithms). Finally, the future energy grid also depends on influences from other domains such as weather and socio-ecological aspects (e.g., pricing strategies). Including these data sets into our context-aware monitoring system can further enhance the quality and reliability of monitoring results.

Acknowledgments

This work was partially supported by the EU H2020 IoTwins Innovation Action project (g.a. 857191). The presented approach will be implemented and tested in the Smart Grid testbed of Aspern Smart City Research (ASCR), which also provided the data used for the first investigations.

References

[1] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on smart grid communication infrastructures: Motivations, requirements and challenges," *IEEE communications surveys & tutorials*, vol. 15, no. 1, pp. 5–20, 2012.

[2] D. Ratasich, F. Khalid, F. Geissler, R. Grosu, M. Shafique, and E. Bartocci, "A roadmap towards resilient internet of things for cyber-physical systems," *CoRR*, vol. abs/1810.06870, 2018.

[3] "Congestion management cost," https://www.apg.at/de/markt/Markttransparenz/Uebertragung/Engpassmanagementkosten, accessed: 2021-02-27.

[4] M. Götzinger, N. TaheriNejad, H. A. Kholerdi, A. Jantsch, E. Willegger, T. Glatzl, A. M. Rahmani, T. Sauter, and P. Liljeberg, "Model-free condition monitoring with confidence," *International Journal of Computer Integrated Manufacturing*, vol. 32, no. 4-5, pp. 466–481, 2019.

[5] A. Jantsch, N. Dutt, and A. M. Rahmani, "Self-awareness in systems on chip—a survey," *IEEE Design & Test*, vol. 34, no. 6, pp. 8–26, 2017.

[6] N. Dutt, A. Jantsch, and S. Sarma, "Toward smart embedded systems: A self-aware system-on-chip (soc) perspective," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 15, no. 2, p. 22, 2016.

[7] N. TaheriNejad, A. Jantsch, and D. Pollreisz, "Comprehensive observation and its role in self-awareness; an emotion recognition system example." in *FedCSIS Position Papers*, 2016, pp. 117–124.

[8] S. A. Boyer, *SCADA: supervisory control and data acquisition*. Isa Research Triangle Park, 1999, vol. 3.

[9] T. S. Bomfim, "Evolution of machine learning in smart grids," in *2020 IEEE 8th International Conference on Smart Energy Grid Engineering (SEGE)*, 2020, pp. 82–87.

[10] G. Dileep, "A survey on smart grid technologies and applications," *Renewable Energy*, vol. 146, pp. 2589–2625, 2020.

[11] M. Donohoe, B. Jennings, and S. Balasubramaniam, "Context-awareness and the smart grid: Requirements and challenges," *Computer Networks*, vol. 79, pp. 263–282, 2015.

[12] D. Hauer, D. Ratasich, L. Krammer, and A. Jantsch, "A methodology for resilient control and monitoring in smart grids," in *2020 IEEE International Conference on Industrial Technology (ICIT)*, 2020, pp. 589–594.

[13] K. M. Silva, B. A. Souza, and N. S. D. Brito, "Fault detection and classification in transmission lines based on wavelet transform and ann," *IEEE Transactions on Power Delivery*, vol. 21, no. 4, pp. 2058–2063, 2006.

[14] B. Fan, Z. Du, X. Jin, X. Yang, and Y. Guo, "A hybrid fdd strategy for local system of ahu based on artificial neural network and wavelet analysis," *Building and environment*, vol. 45, no. 12, pp. 2698–2708, 2010.

[15] Y. Guo, J. Wall, J. Li, and S. West, "Intelligent model based fault detection and diagnosis for hvac system using statistical machine learning methods," vol. 119, 01 2013, pp. 01–08.

[16] A. L. Dexter and D. Ngo, "Fault diagnosis in air-conditioning systems: A multi-step fuzzy model-based approach," *HAC&R Research*, vol. 7, pp. 83–102, 2001.

[17] M. Götzinger, D. Juhász, N. TaheriNejad, E. Willegger, B. Tutzer, P. Liljeberg, A. Jantsch, and A. M. Rahmani, "Rosa: A framework for modeling self-awareness in cyber-physical systems," *IEEE Access*, vol. 8, pp. 141 373–141 394, 2020.