Case-Based Activity Detection from Segmented Internet of Things Data*

Ronny Seiger[1]†, Alexander Schultheis[2,3]†, and Ralph Bergmann[2,3]

[1] University of St. Gallen, Rosenbergstrasse 30, St. Gallen, 9000, Switzerland
[2] German Research Center for Artificial Intelligence (DFKI)
Branch Trier University, Behringstraße 21, 54296 Trier, Germany
[3] Artificial Intelligence and Intelligent Information Systems, Trier University, 54296 Trier, Germany, http://www.wi2.uni-trier.de
`Ronny.Seiger@unisg.ch`
`{Alexander.Schultheis,Ralph.Bergmann}@dfki.de`
`{schultheis,bergmann}@uni-trier.de`

Abstract The use of Internet of Things (IoT) technologies drives the automation of business processes. However, such environments often lack process awareness and corresponding systems to monitor process executions. Due to its too fine-grained nature and variations, the direct use of data from IoT devices for monitoring is problematic, requiring an event abstraction step to lift the data to the business process level. This work investigates the application of Temporal Case-Based Reasoning (TCBR) as a novel experience-based approach to detect process activity executions in IoT data. The proposed TCBR approach uses *activity signatures*–representations of process and IoT data for an activity prototype–as a case base to classify unknown IoT time series data from a smart factory. A data flow architecture is presented that supports analysts in selecting a suitable activity prototype and evaluating its quality for activity detection. The results enable both, the development of high-quality activity detection services and the identification of improvement opportunities in IoT monitoring systems. The approach is evaluated using data produced by a smart factory. The results indicate that the TCBR methods used are very suitable for detecting activities in this IoT use case.

Keywords: Temporal Case-Based Reasoning · Time Series Data · Activity Detection · Internet of Things

1 Introduction

The automation of business processes in many domains of every-day life is rapidly advancing, empowered by the ongoing developments and dissemination of *Internet of Things* (IoT) technologies. The IoT introduces new data sources–*Sensors*–providing insights into the operations, interactions, and environment of smart devices, humans, and digital software systems. Although processes are ubiquitous

* The final authenticated publication is available online at https://doi.org/10.1007/978-3-031-96559-3_29

† These authors contributed equally.

in many IoT-related domains [47], *Process-Aware Information Systems* (PAIS) are often not available to monitor or even orchestrate their executions [39]. Instead, it is necessary to rely on the data produced by the sensors to derive information about the execution of processes and activities in these processes, which can be the basis for more advanced process analysis via process mining [19]. Relying on IoT data, however, entails several challenges as the sensor data is often recorded at a too fine-grained, low level, and it might be subject to variations, which negatively influences the quality of process activity detections.

To gain insights into the execution of process activities, an *event abstraction* step is necessary to lift the low-level sensor data from IoT devices to the level of process events (cf. Fig. 2) [10]. While many related works propose to apply rather heavy-weight machine learning approaches in this context, we discuss the application of *Temporal Case-Based Reasoning* (TCBR) to classify the sensor data, in the form of a time series, according to the process activities they belong to. We start with the data analyst identifying one *activity signature* as a case, which represents the IoT data for a prototypical execution of one type of activity over time [39]. The main challenge is selecting a representative activity execution because the underlying IoT data is often affected by variations for the same activity type. The cases from several activity types are used to classify the segmented IoT data recorded from a small-scale smart factory during the execution of multiple instances of typical production processes with manufacturing activities. We propose a data analysis architecture based on TCBR to support data analysts with selecting a suitable activity signature and evaluating its quality for detecting instances of the corresponding process activity. From the calculated similarities, we can inform the analyst about the match of the selected signature with specific instances, potential ambiguities [14], and other data quality issues that may arise from the aforementioned variations in the IoT sensor data [27]. These insights will 1) enable the development of high-quality activity detection services that are independent of PAIS as described in [38], which is especially relevant in less automated domains with manual activities (e.g., smart healthcare [13]); and, 2) they will also enable IoT engineers to identify opportunities for improving the monitoring of IoT data systems via additional sensors.

The paper is structured as follows: Sect. 2 introduces foundations and the novel use case of activity executions in IoT systems. In Sect. 3 related work is presented and discussed. Sect. 4 presents our approach to detect activity executions from IoT data based on TCBR. This approach is evaluated and discussed in Sect. 5. Finally, Sect. 6 concludes the paper and outlines future work.

2 Foundations and Use Case

2.1 IoT Data and Process Activities in the Smart Factory

IoT System: In this work, we investigate IoT systems as novel data sources and application domains for *Case-Based Reasoning* (CBR). The examples used in this work are taken from the domain of manufacturing. The small-scale smart factory depicted in Fig. 1 represents the IoT system that executes processes

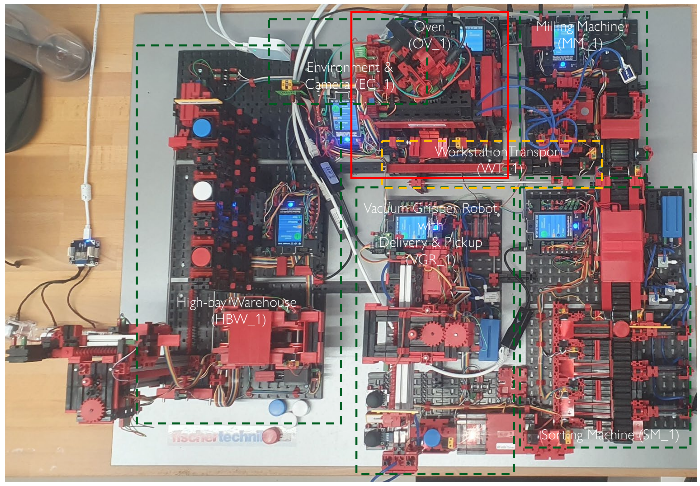

Fig. 1. Small-scale Smart Factory with Production Stations used as IoT System.


```
{ "id": "8ee0fa9a-6dad-4bf8-a0dc-df01333b493f", "station": "MM_1",
"timestamp": "2023-01-30 13:06:22.24", "i1_pos_switch": 0, "o7_valve": 0,
"i2_pos_switch": 0, "i3_pos_switch": 0, "i4_light_barrier": 1,
"m1_speed": 512, "m2_speed": 0, "m3_speed": 0, "o8_compressor": 0 }
```


Listing 1. Example IoT Data (in JSON) from Milling Station.

and emits low-level IoT data [26]. The smart factory features seven production stations including a *high-bay warehouse* (HBW) for storage, a *vacuum gripper crane* (VGR) for transportation, an oven and milling machine for production tasks, and a sorting machine. All the hardware components can be controlled via software executed on embedded controllers, which are networked and expose an interface to be remotely accessed by more complex software applications, e.g., in a service-based software architecture [41].

IoT Data: During its operations, the smart factory regularly emits low-level IoT data regarding the status of its sensors (e.g., buttons and light barriers) and actuators (e.g., motors, valves, and compressors). An exemplary low-level IoT data reading from the milling machine is presented in Listing 1. This reading has a unique ID, timestamp, and the specific status of all sensors and actuators that are part of the station. The factory emits these readings for all stations at a configurable frequency. Thus, we are faced with time series data to be analyzed.

IoT Processes: In this work, we will analyze two topical processes from discrete manufacturing [39]. The *Storage* process contains a sequence of automated process activities to store a new workpiece in the HBW including transport and loading activities. The *Production* process contains a sequence of activities to unload a workpiece and produce it in the oven and milling machine stations with subsequent sorting and transportation. Both processes can be executed in the smart factory using a *Workflow Management System* (WfMS) based on the software stack presented in [41]. The process activities contained in both processes are the entities whose execution we aim to detect solely based on the low-level IoT data from all production stations. Hereby, an activity is considered to be an atomic unit of work executed either by a machine, human, or digital service [47]. An activity has a label, and the execution of an activity is characterized by a timestamped start event and end event, which determine its duration.

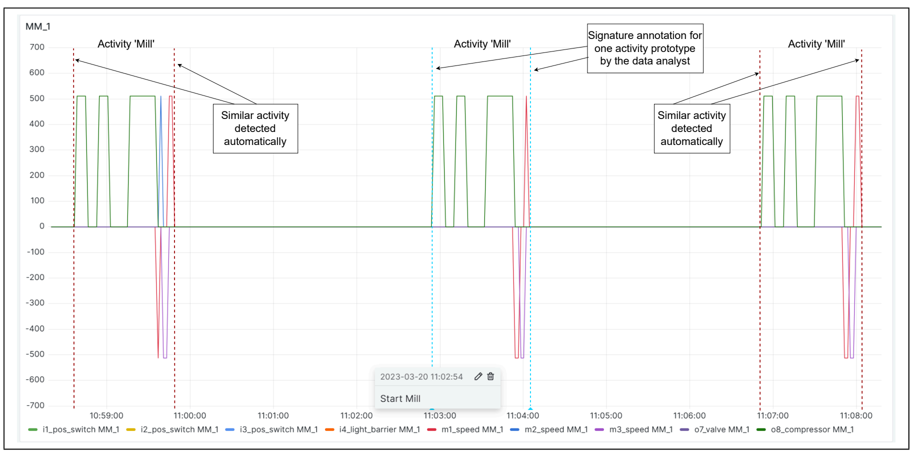

Fig. 2. IoT Data for Three Instances of the *Mill* Activity with Signature Annotation for One Execution as Event Abstraction (Middle, Dashed Light Blue Lines). The Two Similar Instances (Left and Right, Dashed Dark Red Lines) Should Be Detected Automatically.

IoT Data–Process Activity Correlation (Event Abstraction): Fig. 2 shows an example of the correlation between the low-level IoT data and process activity as an event abstraction step. Each differently colored solid line represents the values of one sensor or actuator over time. Given such time series of low-level data readings from all sensors and actuators of the IoT system, the corresponding process activity should be inferred. We investigate the applicability of CBR to this problem and type of data. For example, Fig. 2 contains the execution of three instances of the *Mill* activity that should be classified accordingly only given the IoT data. One instance thereby serves as prototype determined by the data analyst to detect the other occurrences. One challenge in creating this correlation and finding a suitable prototype activity is the existence of variations in the IoT data associated with different executions of the *same* type of activity. In Fig. 2, we can observe slight variations in the durations of how long a sensor or actuator stays in a specific state, and we can see an additional peak in the value of a sensor for the first instance. However, all three sequences represent executions of the same type of activity. These variations make it challenging to select a suitable prototype segment within the low-level IoT data (cf. Fig. 2) that best represents a process activity from start to end. We propose to use CBR to support the IoT data analysts with this selection process.

IoT Data Variations: Our experience with IoT systems [27] and analysis of the variations in the IoT data unveiled the following reasons for their occurrence between instances of the same type of activity:

– Different execution times: Activity executions may vary in their duration due to different process parameters and different start/end conditions.
– Different process/activity parameters: Activity executions may involve different sequences of sensors and actuators, depending on specific parameters.
– Different start/end conditions: Activity executions may not always start or end with the same conditions, resulting in different movement patterns.
– External factors: IoT systems interact with the physical world, where external conditions and factors are harder to control than in pure digital systems. This might create different types of variations in the IoT data.
– Different sampling frequencies: The sampling frequencies of the underlying IoT data may also contribute to variations.
– Parallel activity executions: The parallel execution of activities in the IoT system might lead to different sensor and actuator patterns in the IoT data than for consecutive executions, possibly superimposing each other. In this work, we assume that these scenarios are out of scope and the execution of an activity is limited to only one execution on one station at a time.

2.2 Temporal Case-Based Reasoning

The application of CBR to IoT sensor data falls within the research area of TCBR [18, 22]. This deals with the representation of temporal relationships in cases and their handling within the CBR cycle. TCBR is used in various application areas such as classification and error detection, prediction or medicine [25]. In each use case, the underlying data is complex, resulting in challenges for filling the knowledge containers [32]. The application of TCBR requires the definition of a suitable vocabulary for modeling the sensor data and the selection of suitable similarity measures to enable correct reuse of experience knowledge.

Various forms of case representation exist for filling the vocabulary and on this foundation the case base [25]. While approaches such as episodes, graphs or event sequences exist, time series are the most established form of representation in TCBR. A time series is defined as a sequence of values that are associated with specified points in time. In most applications, this data is represented in symbolic form and summarized as feature vectors. Simplified forms of representation such as temporal abstraction [43] can be used to reduce the volume of data. These abstract the time series to a higher level by aggregating data points, which decreases the following computation complexity. However, such an abstraction leads to a loss of information, which can be unsuitable for certain use cases [37].

Usually, a case representing IoT sensor data contains one or more time series [25]. These are local attributes that are aggregated in a higher-level case description.

Based on the selected vocabulary, suitable similarity measures must be used to identify similar time series [25]. Table 1 presents a categorization of syntactic similarity measures between time series. In these measures, no domain knowledge is included so that these can be defined as knowledge-poor [45]. Furthermore, there are knowledge-intensive similarity measures that compare time series not only based on syntax, but also taking semantics into account (e.g., by Nakanishi [29]). Such semantic time series measures have not yet been used in TCBR and for the use cases in the analysis of IoT sensor data. A syntactic consideration regarding stretching and compression has so far been sufficient [25]. The categories of similarity measures are based on the local-global principle [4, pp. 106-107], in which the similarities of individual attribute values are calculated and aggregated into a global similarity value. This means that at a local level, similarity measures are used for the point in time and the corresponding value characteristic. Depending on the use case, these measures can be semantic or syntactic measures. As several time series might be available for most sensor data cases, the similarity values between these time series can also be aggregated at a global level and merged into one similarity value.

Table 1. The Three Categories of Syntactic Similarity Measures for Time Series, With Example Algorithm for Each Category (according to Malburg et al. [25]).

Category 1	Category 2	Category 3
Similarity measures that can only be applied to time series of the same length, comparing only the values at the corresponding times.	Similarity measures that can be applied to time series of different lengths, considering the values and the time points.	Similarity measures, like those in Cat. 2, but that can detect stretching and compression in addition.
List Mapping [25]	Smith-Waterman Algorithm (SWA) [44]	Dynamic Time Warping (DTW) [33]

3 Related Work

3.1 Classification Based on Temporal Case-Based Reasoning

Related work already exists in which TCBR is applied to the smart manufacturing domain and applied for classification. Klein et al. [21] learn a similarity measure using expert knowledge for classification in the field of predictive maintenance. Based on the similarity-based results, the approach classifies whether faults will occur based on sensor data so that preventive actions can be taken. Malburg et al. [25] present the use case of event and activity detection based on TCBR as an example of an implementation in the CBR framework ProCAKE [5].

Based on a large case base with collected sensor data, a retrieval is performed and based on the most similar cases, a classification is done. Based on this data, Weich et al. [48] investigate the use of embeddings in TCBR for this use case to improve retrieval efficiency. Schultheis et al. [37] use TCBR to identify specific data quality issues by using *Dynamic Time Warping* (DTW) similarity analysis to detect anomalies and classify these on experience base. All three approaches use IoT data from factories and apply empirical knowledge for classification.

TCBR is used for classification in the medical domain in the context of decision support systems. For a binary classification to determine the creatinine course as critical or non-critical, Schlaefer et al. [35] use CBR. Their similarity calculation is based on a distance metric determined by linear regression. Funk and Xiong [16] perform classification of diseases based on similar cases represented as sequences. Similarly, Nilsson et al. [30] pursue the goal of identifying stress-related disorders based on patterns in time series. Elsayed et al. [11] convert medical images into curves and perform retrieval on them using DTW as a similarity measure. The knowledge stored in the case base is used to classify disease findings. The analysis of patient data from people with back pain is investigated by Szczepanski et al. [46], using time series to derive targeted treatment recommendations based on similar cases. As with activity detection, these areas of application are based on sensor data that is to be classified using CBR.

CBR can also be used to analyze IoT data outside the manufacturing and medical domains. Fritsche et al. [15] use TCBR to identify critical situations by comparing current time series with historical cases, using DTW to detect potential failure cases. Borck et al. [7] apply a similar method to monitor astronauts, recognizing activities and identifying errors in their execution. To monitor elder people at home, Lupiani et al. [24] use TCBR with temporal edit distance as a measure. Data from a smart home environment is used to identify what activities people are currently performing and if these are risks. Ariza et al. [2] use CBR to classify player skill levels based on multidimensional time series measurements. These are also case-based classifications based on time series data.

3.2 Process Activity Detection from IoT Data

The BPM-IoT manifesto is among the advocates of applying Business Process Management (BPM) technologies to the IoT, and vice versa [19]. It highlights the opportunities and challenges of using IoT data for and together with process mining, which is discussed in more detail in [27]. These works highlight the bridging of the abstraction gap between low-level IoT data and process-level data as an important, non-trivial task, which also includes choosing a suitable level of event granularity for process mining [3]. The study in [27] discusses the contributions of sensor data to the execution of a process activity, including sensors emitting data that is of no relevance to sensor data that is relevant for several activities. In our work, we start from the concept of an *activity signature* [39], which involves the data analyst specifying the relevant IoT sensors in a first step.

Various approaches investigating activity detection and IoT data segmentation have been proposed in [20, 31], and more specialized in the context of smart homes [9, 12], smart cars [6], human activity detection [8], and industrial processes [28]. A common challenge lies in the bridging of the abstraction gap between low-level event data and high-level process events [10], which is the goal of our work. Most of these approaches are either tied to specific types of sensors, which limits their general applicability, or they train expensive overly complex machine learning models to classify activity executions. The presence of aforementioned variations within the IoT data negatively influences the quality of the activity detections in most of these cases. We aim to develop a more generic light-weight approach that is able to tolerate these variations to a certain degree.

4 Case-based Activity Detection from IoT Data

In Fig. 3, we present an overview of the data flow architecture that we propose for learning (top part) as well as for retrieval and reuse (bottom part).

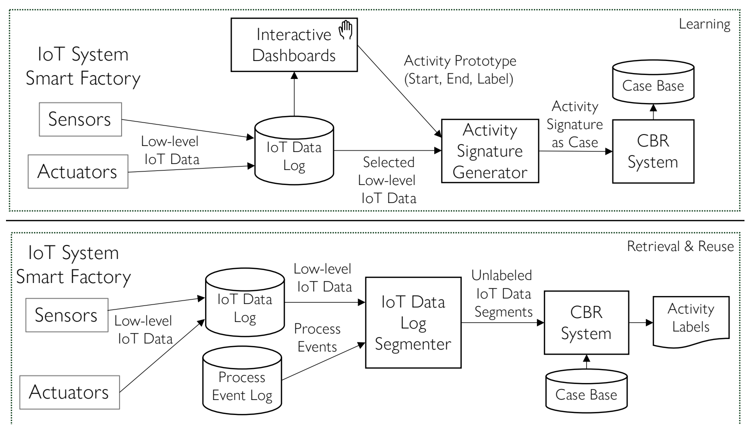

Fig. 3. Data flow Architecture for Learning and CBR Application.

4.1 Learning Phase

In the *Learning Phase*, the data from the IoT system's sensors and actuators (cf., e.g., Listing 1) recorded during the exemplary execution of the processes is persisted in an IoT data log (e.g., a time series database). This data is visualized in an interactive dashboard (cf. Fig. 2) where the data analyst identifies and annotates a prototypical execution of an activity according to the method described in [39]. In the learning phase, we assume that a PAIS for process monitoring is not available and thus, the analyst needs to provide the data annotations for one prototype of an activity execution. The provided data–the activity label, its start and end timestamps–is then used together with the corresponding IoT data from the data log (aligned based on the timestamps) to generate the activity signature [39]. This signature represents the activity execution's manifestation in the IoT data, which is converted to a case in the CBR system. A vocabulary exists for this purpose, which provides the storage of the relevant time series at local level and as the attributes relating to the process on global level (e.g., the executing station in the factory). The signatures are converted into the format specified by the vocabulary so that the signatures are contained in the case base.

To apply the TCBR approach with this vocabulary and case base, local similarity measures must be modeled by domain experts to be able to compare the time points of the time series and the corresponding values (e.g., coordinates or Boolean conditions), and aggregate these for each time series entry appropriately. The comparison of the time series itself is performed using DTW [33] based on the calculated similarity values for the individual entries [34]. This similarity measure aligns entries of two time series by computing an optimal mapping based on those local similarities. Therefore, DTW has the potential of recognizing the time series lengths without penalizing the subsequent similarity values (cf. Table 1). This enables the signatures used in activity detection, only containing a minimal number of time points, to be compared with the significantly longer time series from the query and recognized as similar. For aggregation on a global level, a similarity measure is used that learns class-specific weights (cf. [4, p. 122]) from the case base. Due to the number of sensors in smart factories, not all recorded time series are of relevance to the respective activity. Only the relevant time series are stored in the case itself, while other attributes are ignored. In contrast, the query can contain significantly more attributes with associated time series, as all sensors are continuously producing data. For similarity assessment, therefore, an aggregation function with class-specific weights is required, whereby the weights are only binary and indicate whether an attribute is relevant or not. This similarity knowledge can be derived from the case base by extracting the respective activity as a label from each case and only considering through an equal weight higher than 0.0 the attributes for which the case contains time series. All other attributes not contained in the case are weighted with 0.0 and thus omitted from the similarity calculation. This means that the global similarity measure is automatically learned based on the case. From these relevant, identified attributes, the average is calculated as global similarity value. As adaptation knowledge, only null adaptation is used for this domain because there is only one case derived from the activity prototype in the case base for each activity, and it is therefore assumed that the most similar case contains the correct label.

4.2 Application of the CBR Cycle

The application of the CBR cycle [1] focuses on *Retrieval and Reuse Phases*. We aim to classify unknown data from the IoT system, which is stored in the IoT data log, according to the activity executions it represents. Our goal is to only classify the IoT data in a first step using TCBR, not to segment it (i.e., to identify specific start and end times). For the segmentation task, we use the process event log recorded by a WfMS during the execution of the processes. This event log contains start and end times of activity executions to be used for segmentation, and their labels to be used as ground truth for evaluations. Note that the process event log has been created automatically by executing process models that describe the sequence of activities (e.g., as described in [41]). This event log does not have to be created by a WfMS or PAIS, as we assume that these systems might not be available in the IoT system. It can also be created by other activity monitoring approaches–even a manual tracking of start and end events–while recording the corresponding IoT data. The unlabeled and segmented low-level IoT data is fed into the CBR system for classification.

In the CBR system, a retrieval is triggered that works based on the similarity measures described in Sect. 4.1. The activity from the most similar case is reused and, therefore, returned as an identified activity label including the similarity value. This value serves as a confidence value for the data analyst, as it can be used to assess whether the label is determined with a high degree of certainty (high similarity value) or based on the lack of suitable alternative cases (low similarity value). In addition, all cases with similarity values can be returned to check whether other cases have similarly high values, indicating potential ambiguities [14]. These results help the data analyst to assess the quality of detecting activity executions from the selected prototypical activity signature, and thus, if a modification of the signature or even an adjustment of the IoT system (e.g., by adding new sensors [27]) is necessary. Once the activity signature is confirmed to be suitable, it can serve as an input for automatically generating a corresponding activity detection service based on the approach presented in [38].

5 Evaluation

The activity detection approach is implemented in the *ProCAKE* framework [5][1] which has already been used for TCBR in previous research. In this section, we present the data collection and analysis setup, and discuss the results.

5.1 Setup

To evaluate the activity detection, we modeled the two IoT processes described in Sect. 2.1. The *storage* process consists of 6 activities, the *production* process consists of 10 activities; both to be executed in sequence. We used a WfMS to execute several instances of these processes over 27.63 minutes to record an IoT data log for the learning phase. In this log, 12 distinct activities were annotated from six production stations by an expert data analyst. From these annotations, we generated the corresponding 12 activity signatures to be stored as 12 cases in the case base. To test the TCBR approach, 36 instances of the two processes were executed in the factory, with a total duration of 178.44 minutes and 359 activity executions. These activities are used as queries for the evaluation, each

[1] Available at: https://gitlab.rlp.net/procake/publications/procake-activity-detection

based on the executing stations. The IoT data logs were sampled at a frequency of 0.5 Hz. The process event log recorded by the WfMS was used as ground truth and to segment the activity execution. The time required to execute the retrieval was 24.89 seconds on a state-of-the-art computer (12 cores@3.2 Ghz, 32 GB RAM). The dataset in [42] contains all relevant input and result data.

5.2 Results and Discussion

The confusion matrix for each distinct activity with its executing station is displayed in Table 2. We see that the TCBR-based activity detection works very well in this manufacturing scenario, achieving high accuracy, precision, and F1 scores. As a reference, the work in [40] detects activities in the same scenario based on change patterns encoded as rules and exact matches in the IoT data, achieving an average F1 score of less than 0.5. Better results for detecting the manufacturing activities are achieved in [17]. However, this approach relies on manual adjustment, and it is not able to handle variations in the IoT data.

Table 2. Performance Metrics for all Distinct Detected Activities.

Activity (Station)	Accuracy	Precision	Recall	F1 Score
Transport to sink (VGR)	0.988	0.991	0.800	0.883
Calibrate VGR (VGR)	0.856	0.723	0.507	0.598
Transport to oven (VGR)	0.992	0.947	0.935	0.941
Sort (SM)	0.922	0.421	1.0	0.592
Mill (MM)	0.981	0.761	1.0	0.864
Burn (OV)	1.0	1.0	1.0	1.0
Transport to DPS (VGR)	0.991	0.977	0.915	0.945
Store (HBW)	0.980	1.0	0.757	0.861
Unload (HBW)	0.983	1.0	0.714	0.833
Get Workpiece from DPS (VGR)	0.983	0.898	0.884	0.891
WT Transport (WT)	1.0	1.0	1.0	1.0
Calibrate HBW (HBW)	0.902	0.585	0.693	0.635
Average	0.965	0.859	0.850	0.837

For production stations that are capable of executing only one type of activity (MM, OV, WT) the metrics are especially good. A notable difference is the *Sort* activity executed by the *Sorting Machine* (SM). The sorting activity shows variations in the associated IoT data (cf. Sect. 2.1) as it involves different light barrier sensors and valves to sort a workpiece according to its color. One activity signature as case for this activity is not sufficient here. We expect that multiple signatures and thus cases for the different sorting behavior will improve the detection quality. The VGR is a central component for transportation, which executes different types of activities that are mostly distinguished by the start and end position parameters of the crane (cf. Sect. 2.1). The distinction of the individual activities here is more difficult and results in slightly decreased performance. Similar results can be observed with the store and unload activities in the HBW station. They show similar behavior in the IoT data, which may lead to ambiguities in the detections [14]. In the case of the HBW, an additional sensor indicating if the container that is being retrieved and then stored is full or empty would already suffice to make the detection more accurate. Finally, the *calibrate* activities executed by HBW and VGR show rather poor performance of the activity detection, as these are very short and show movement patterns in the IoT data that are also part of activity types executed by the two stations.

Overall, the metrics seem acceptable for the detection of automated activities in the small-scale production setting. The investigations of activities in real-world production settings and domains with less automated, more manual activities (e.g., smart healthcare) that are often characterized by stronger variations in the IoT data, are subject to future work. With our proposed data flow architecture, we can support the data analyst with selecting a suitable prototype signature for activities to be detected based on IoT data. The TCBR-based approach provides feedback about the detection qualities for a specific type of activity given a signature selected by the data analyst. Apart from handling variations quite well, the approach is also able to identify potential quality issues in the detection which might indicate the need of adjusting an activity signature, providing more signatures for an activity, or adding new sensors. Ultimately, the data analyst has to decide about an acceptable activity detection quality based on given use cases and requirements. When the quality is acceptable, the signature can be used as basis for the automatic implementation of a light-weight detection service as described in [38], removing the need of any heavy-weight PAIS to monitor activity executions. In this work, we assume that the unknown IoT data is already segmented, and the task is to classify these segments according to the activity executions. The investigation of using TCBR to also segment the data based on the activity signatures is subject to future work. The execution times of the retrieval for the size of the case based specified above seem reasonably fast to move the activity detection also to online detection settings where streams of IoT data are being analyzed at runtime in edge computing scenarios [36].

6 Conclusion and Future Work

This paper proposes a novel architecture for detecting process activity executions based on low-level IoT sensor data, which requires an event abstraction step to lift this time series data to the process level. For this purpose, TCBR is applied as a methodology to support data analysts in selecting suitable prototypes of activity executions together with the corresponding IoT data as cases. These cases are then used to detect similar occurrences of the activity executions in unknown IoT data. The proposed architecture is evaluated using a prototype implementation with smart factory data. The evaluation suggests that the approach works very well, even in the presence of variations in the IoT data, which related approaches do not address. In contrast to related work, our proposed solution is rather light-weight, works already with a small number of cases, and successfully applies CBR to time series data from IoT systems as a novel application domain.

In future work, we will investigate an extension of the approach and architecture with the support for parallel activity executions and a segmentation component. In its current form, the architecture also supports streaming of IoT data in real time [41]. The segmentation component will recognize when a new activity starts and an existing one ends. This allows the sensor data to be converted into queries and classified using the TCBR approach. Such a segmentation component can be investigated on a case-based basis, but can also be realized using other methods (e.g., [23]). The pipeline can also be evaluated on this basis for real-time applications in IoT. Furthermore, an explanation component will be examined, with the help of which data analysts are shown transparently how the similarity between the query and the case containing the signature comes about. We will also investigate the integration of additional semantic information about dependencies between time series in the cases and the extent to which this contributes to the optimization of the similarity value.

Acknowledgments. This work has received funding from the Swiss National Science Foundation under Grant No. IZSTZ0_208497 (*ProAmbitIon* project [13]). This work is also partly funded by the German Federal Ministry for Economic Affairs and Climate Action under grant No. 01MD22002C *EASY* [36].

References

1. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Commun. **7**(1), 39–59 (1994)
2. Ariza, D.S.L., Sánchez-Ruiz, A.A., González-Calero, P.A.: Time Series and Case-Based Reasoning for an Intelligent Tetris Game. In: 25th ICCBR Proc. LNCS, vol. 10339, pp. 185–199. Springer (2017)
3. Beerepoot, I., Di Ciccio, C., Reijers, H.A., Rinderle-Ma, S., Bandara, W., et al.: The biggest business process management problems to solve before we die. Computers in Industry **146**, 103837 (2023)
4. Bergmann, R.: Experience Management: Foundations, Development Methodology, and Internet-Based Applications, LNCS, vol. 2432. Springer (2003)
5. Bergmann, R., Grumbach, L., Malburg, L., Zeyen, C.: ProCAKE: A Process-Oriented Case-Based Reasoning Framework. In: 27th ICCBR Workshop Proc. vol. 2567, pp. 156–161. CEUR-WS.org (2019)
6. Beyel, H.H., Makke, O., Pourbafrani, M., Gusikhin, O., van der Aalst, W.M.: Analyzing data streams from cyber-physical-systems: A case study. SN Computer Science **5**(6), 706 (2024)
7. Borck, H., Johnston, S., Southern, M., Boddy, M.S.: Exploiting Time Series Data for Task Prediction and Diagnosis in an Intelligent Guidance System. In: 31st ICCBR Workshop Proc. vol. 1815, pp. 132–141. CEUR-WS.org (2016)
8. Cornacchia, M., Ozcan, K., Zheng, Y., Velipasalar, S.: A survey on activity detection and classification using wearable sensors. IEEE Sensors **17**(2), 386–403 (2016)
9. Di Federico, G., Burattin, A.: Cvamos—event abstraction using contextual information. Future Internet **15**(3), 113 (2023)
10. Diba, K., Batoulis, K., Weidlich, M., Weske, M.: Extraction, correlation, and abstraction of event data for process mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **10**(3), e1346 (2020)
11. Elsayed, A., Hijazi, M.H.A., Coenen, F., García-Fiñana, M., Sluming, V., Zheng, Y.: Time Series Case Based Reasoning for Image Categorisation. In: 19th ICCBR Proc. LNCS, vol. 6880, pp. 423–436. Springer (2011)

12. Esposito, L., Leotta, F., Mecella, M., Veneruso, S.: Unsupervised segmentation of smart home logs for human habit discovery. In: 18th IE Proc. pp. 1–8. IEEE (2022)
13. Franceschetti, M., Seiger, R., González, M.J.G., Garcia-Ceja, E., Flores, L.A.R., García-Bañuelos, L., Weber, B.: Proambition: Online process conformance checking with ambiguities driven by the internet of things. In: CAiSE Research Projects Exhibition. pp. 52–59 (2023)
14. Franceschetti, M., Seiger, R., López, H.A., Burattin, A., García-Bañuelos, L., Weber, B.: A characterisation of ambiguity in bpm. In: International Conference on Conceptual Modeling. pp. 277–295. Springer (2023)
15. Fritsche, L., Schlaefer, A., Budde, K., Schröter, K., Neumayer, H.: Recognition of Critical Situations from Time Series of Laboratory Results by Case-Based Reasoning. J. Am. Medical Informatics Assoc. **9**(5), 520–528 (2002)
16. Funk, P., Xiong, N.: Case-Based Reasoning and Knowledge Discovery in Medical Applications with Time Series. Comput. Intell. **22**(3-4), 238–253 (2006)
17. García-Bañuelos, L., González González, M.J., Seiger, R., Franceschetti, M., Silva Trujillo, A.G.: A semi-automated approach to detecting process-level activities from sensor data. In: 8th EDI40 Proc. (2025)
18. Jære, M.D., Aamodt, A., Skalle, P.: Representing Temporal Knowledge for Case-Based Prediction. In: 6th ECCBR Proc. LNCS, vol. 2416, pp. 174–188. Springer (2002)
19. Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A., et al.: The Internet-of-Things Meets Business Process Management. A Manifesto. IEEE Syst. Man Cybern. Mag. **6**(4), 34–44 (2020)
20. Janssen, D., Mannhardt, F., Koschmider, A., van Zelst, S.J.: Process model discovery from sensor event data. In: 2nd ICPM Proc. pp. 69–81. Springer (2020)
21. Klein, P., Weingarz, N., Bergmann, R.: Enhancing Siamese Neural Networks Through Expert Knowledge for Predictive Maintenance. In: 2nd IoT Streams Workshop Proc. CCIS, vol. 1325, pp. 77–92. Springer (2020)
22. López, B.: Case-Based Reasoning: A Concise Introduction. SLAIML, Springer (2013)
23. Lovrić, M., Milanović, M., Stamenković, M.: Algoritmic methods for segmentation of time series: An overview. JCEBI **1**(1), 31–53 (2014)
24. Lupiani, E., Juarez, J.M., Palma, J.T., Marín, R.: Monitoring elderly people at home with temporal Case-Based Reasoning. Knowl. Based Syst. **134**, 116–134 (2017)
25. Malburg, L., Schultheis, A., Bergmann, R.: Modeling and Using Complex IoT Time Series Data in Case-Based Reasoning: From Application Scenarios to Implementations. In: 31st ICCBR Workshop Proc. vol. 3438, pp. 81–96 (2023)
26. Malburg, L., Seiger, R., Bergmann, R., Weber, B.: Using Physical Factory Simulation Models for Business Process Management Research. In: BPM 2020 Workshop Proc. LNBIP, vol. 397, pp. 95–107. Springer (2020)
27. Mangler, J., Seiger, R., Benzin, J.V., Grüger, J., Kirikkayis, Y., et al.: From internet of things data to business processes: Challenges and a framework. arXiv preprint arXiv:2405.08528 (2024)
28. Mannhardt, F., Bovo, R., Oliveira, M.F., Julier, S.: A taxonomy for combining activity recognition and process discovery in industrial environments. In: 19th IDEAL Proc. pp. 84–93. Springer (2018)
29. Nakanishi, T.: Semantic Waveform Model for Similarity Measure by Time-series Variation in Meaning. In: 10th IIAI-AAI Proc. pp. 382–387 (2021)

30. Nilsson, M., Funk, P.J., Olsson, E.M.G., von Schéele, B., Xiong, N.: Clinical Decision-Support for Diagnosing Stress-Related Disorders by Applying Psychophysiological Medical Knowledge to an Instance-Based Learning System. Artif. Intell. Medicine **36**(2), 159–176 (2006)
31. Rebmann, A., Emrich, A., Fettke, P.: Enabling the discovery of manual processes using a multi-modal activity recognition approach. In: BPM 2019 International Workshops, Vienna, Austria, September 1–6, 2019. pp. 130–141. Springer (2019)
32. Richter, M.M.: Knowledge Containers. In: Readings in Case-Based Reasoning. Morgan Kaufmann Publishers (2003)
33. Sakoe, H., Chiba, S.: Dynamic Programming Algorithm Optimization for Spoken Word Recognition. IEEE Trans. Aco. Sp. and Sig. Proc. **26**(1), 43–49 (1978)
34. Schake, E., Grumbach, L., Bergmann, R.: A Time-Series Similarity Measure for Case-Based Deviation Management to Support Flexible Workflow Execution. In: 28th ICCBR Proc. LNCS, vol. 12311, pp. 33–48. Springer (2020)
35. Schlaefer, A., Schröter, K., Fritsche, L.: A Case-Based Approach for the Classification of Medical Time Series. In: 2nd ISMDA Proc. LNCS, vol. 2199, pp. 258–263. Springer (2001)
36. Schultheis, A., Alt, B., Bast, S., Guldner, A., Jilg, D., et al.: EASY: Energy-Efficient Analysis and Control Processes in the Dynamic Edge-Cloud Continuum for Industrial Manufacturing. Künstliche Intelligenz (2024)
37. Schultheis, A., Malburg, L., Grüger, J., Weich, J., Bertrand, Y., Bergmann, R., Serral Asensio, E.: Identifying Missing Sensor Values in IoT Time Series Data: A Weight-Based Extension of Similarity Measures for Smart Manufacturing. In: 32nd ICCBR Proc. LNCS, vol. 14775, pp. 240–257. Springer (2024)
38. Seiger, R., Franceschetti, M., Weber, B.: Data-driven generation of services for iot-based online activity detection. In: ICSOC. pp. 186–194. Springer (2023)
39. Seiger, R., Franceschetti, M., Weber, B.: An interactive method for detection of process activity executions from iot data. Future Internet **15**(2), 77 (2023)
40. Seiger, R., Kurz, A.F., Franceschetti, M.: Online detection of process activity executions from iot sensors using generated event processing services. Available at SSRN 5165943 (2025). https://doi.org/10.2139/ssrn.5165943
41. Seiger, R., Malburg, L., Weber, B., Bergmann, R.: Integrating Process Management and Event Processing in Smart Factories: A Systems Architecture and Use Cases. J. Manuf. Syst. **63**, 575–592 (2022)
42. Seiger, R., Schultheis, A., Bergmann, R.: Dataset from a Smart Factory for Activity Detection using Temporal Case-based Reasoning (Mar 2025). https://doi.org/10.5281/zenodo.14998532
43. Shahar, Y.: A Framework for Knowledge-Based Temporal Abstraction. Artif. Intell. **90**(1-2), 79–133 (1997)
44. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. JMB **147**(1), 195–197 (1981)
45. Stahl, A.: Learning of knowledge-intensive similarity measures in case-based reasoning. Ph.D. thesis, University of Kaiserslautern (2004)
46. Szczepanski, T., Bach, K., Aamodt, A.: Challenges for the Similarity-Based Comparison of Human Physical Activities Using Time Series Data. In: 31st ICCBR Workshop Proc. vol. 1815, pp. 173–177 (2016)
47. Weber, B., Abbad-Andaloussi, A., Franceschetti, M., Seiger, R., Völzer, H., Zerbato, F.: Leveraging digital trace data to investigate and support human-centered work processes. In: Int. Conf. ENASE. pp. 1–23. Springer (2023)

48. Weich, J., Schultheis, A., Hoffmann, M., Bergmann, R.: Integration of Time Series Embedding for Efficient Retrieval in Case-Based Reasoning. In: 33rd ICCBR Proc. LNCS, Springer (2025), Accepted for Publication.