Process Model Discovery from Sensor Event Data

Dominik Janssen[1], Felix Mannhardt[2,3], Agnes Koschmider[1], and Sebastiaan J. van Zelst[4,5]

[1] Group Process Analytics, Kiel University, Germany
doj|ak@informatik.uni-kiel.de
[2] Dept. of Technology Management, SINTEF Digital, Norway
felix.mannhardt@sintef.no
[3] Dept. of Computer Science, NTNU, Norway
[4] Fraunhofer Institute for Applied Information Technology, Fraunhofer Gesellschaft, Germany
sebastiaan.van.zelst@fit.fraunhofer.de
[5] Chair of Process and Data Science, RWTH Aachen University, Germany

Abstract. Virtually all techniques, developed in the area of process mining, assume the input event data to be discrete, and, at a relatively high level (i.e., close to the business-level). However, in many cases, the event data generated during the execution of a process is at a much lower level of abstraction, e.g., sensor data. Hence, in this paper, we present a novel technique that allows us to translate sensor data into higher-level, discrete event data, thus enabling existing process mining techniques to work on data tracked at a sensory level. Our technique discretises the observed sensor data into activities by applying unsupervised learning in the form of clustering. Furthermore, we refine the observed sequences by deducing imperative sub-models for the observed discretised data, i.e., allowing us to identify concurrency and interleaving within the data. We evaluated the approach by comparing the obtained model quality for several clustering techniques on a publicly available data-set in a smart home scenario. Our results show that applying our framework combined with a clustering technique yields results on data that otherwise would not be suitable for process discovery.

Keywords: Process mining · sensor data · event correlation · IoT.

1 Introduction

The rise of the Internet-of-Things (IoT), i.e., interconnected devices, mechanical and digital machines, gradually digitalises the day-to-day operations of modern-day enterprises. More-and-more devices are interconnected and store valuable traces of behavioural data, generated during their interaction with humans, as well as other interconnected devices. For example, consider the concepts of *autonomous production* and the adoption of *robotics in healthcare*, in which operational processes are gradually digitised and automated, utilising interconnecting and communicating devices and machines.

Whereas the design of a single device, connected to a larger network of devices, remains manageable (though it is complex in its own right), deficiencies in inter-device communication or handover of work-packages, easily lead to global process under-performance. Hence, a clear understanding of the general flow of work, as well as an understanding of bottlenecks and synchronisation points is of utmost importance to further improve the efficiency of the executed processes. *Process mining* techniques aim to exploit behavioural data, stored in the information systems to support the execution of processes and to distil process models [1]. In particular, they can derive-and-construct process models based on tracked event data, i.e., in a *completely automated* fashion.

In general, process mining relies on discrete event data, typically assumed to be tracked at *the business level*, i.e., the event data directly relates to high-level business process concepts. However, often, the level at which the event data is tracked within information systems is at a much lower level.

Possible application scenarios are settings where the movement of objects or people (entities) is tracked by motion sensors, light barriers or similar types of sensors that only detect absence and presence of a person or object and cannot distinguish between different observed entities. Those sensors can be found in smart home settings, smart factories and healthcare-related applications. If in these possible settings, it is of interest to discover frequent behaviour patterns or abnormal behaviour, our proposed method provides a novel approach that translates sensory data, into a process model. In particular, unlabelled raw sensor events are aggregated and clustered by an unsupervised learning technique to identify activities through clusters of related event sequences. To identify the activities, we discover a process model for each identified cluster. The activities, labelled by a domain expert, serve as input for process mining-based model discovery, which allows to identify concurrent and interleaving behaviour in sensor event data. We evaluate our approach on the publicly available CASAS dataset [2] and compare two clustering methods. The obtained results show promising results, hinting towards a better result by using clustering based on a self-organising map (SOM) in comparison to basic k-means in this context based on our methodology.

To the best of our knowledge, this paper suggests the first activity and process discovery technique for unlabelled sensor event data using SOM as model and addressing the challenges of concurrent behaviour between activities and multiple residents.

The remainder of this paper is structured as follows. The next section presents related work. Subsequently, section Section 3 presents our approach, which has been evaluated using a real-life data-set. The evaluation is summarised in Section 4. The paper concludes with an outlook on future work in Section 5.

2 Related Work

A large body of research exists that partially addresses the discovery of events and activities at different levels (see Fig. 1). In the following we consider related approaches that use sensor data aiming to translate it into higher-level, discrete event data or applying process mining on raw sensor data. Our focus of related approaches also lays in smart homes as we used position data of smart home sensors for evaluation.

Activity recognition in smart home has been widely addressed relying the recognition on different sensor types like motion or video [3–5] or analysing data from wearables [6,7] or reference sensors [8]. Recently, Deep Learning (DL) methods for detecting and predicting activities in IoT environments have been increasingly explored [9]. Unlike classical machine learning techniques, DL networks automatically derive features from the data and produce promising results in different domains. Particularly in the field of smart homes or ambient assisted living, there are first approaches that recognise activities based on sensor event data [10–13]. Activity recognition is predominantly used for a situational prognosis [14]. Also these kinds of approaches identify simple activities [15,16]. Complex activities like people's daily activities can only be identified using extra sensors [15,17]. Although our method for process model discovery from raw location sensor data also requires a manual labelling of clusters of high-level events, we believe that the process model view on raw sensor data advances existing approaches and is beneficial in terms of evaluating the quality of data aggregations, which DL-based approaches are not capable of.

Mapping low-level events to activities for process mining is still a challenge [18]. The current status-quo is that approaches indicate only likelihoods of mappings, since there is often more than one possible solution [19]. Our approach for event aggregation in combination with unsupervised learning aims to bridge this gap. Related literature for activity discovery for process mining either use supervised techniques [6,20] or visualise human habits [21] in order to accurately identify activities. Some works exist that detect activities from high-level events through unsupervised techniques [20,22,23], which have been compared in this paper. These related works [20,22,23] use patterns or local process models to aggregate event data towards higher abstraction levels. But they did not allow to discover meaningful activities for our data set. For unlabelled training sets, related approaches suggest to use a time-based label refinement [24] or locations [25] as characteristics in order to segment the event log and to abstract activities out of it. However, the methods already expects particular representations of traces. Given our scenario, the application of local process models did not allow to identify useful process fragments.

3 Translating Sensor Data to High-Level Traces

Our method for process model discovery from raw location sensor data assumes a location sensor event log E_L as input derived from a set of sensors S e.g., networks of WiFi-access points, or motion sensors in smart homes. We expect events $e \in E_L$ to satisfy some minimal requirements: For each event we can retrieve a timestamp $time(e)$ inducing a partial order on the events, a sensor label $sensor(e) \in S$ indicating which sensor was activated and some form of informa-

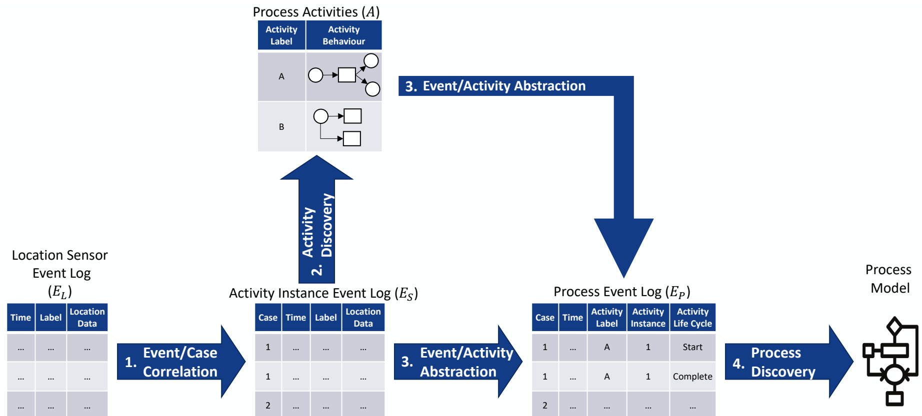

Fig. 1: Process discovery approach for location sensor event data.

tion that either implicitly or explicitly refers to a location (i.e., $location(e) \in L$). The location information can be explicit in the form of coordinates (e.g., latitude, longitude) or implicit by providing labelled locations together with a distance function providing pairwise distances between them. Throughout the paper we assume that E_L was generated by one or more *entities* $n \in N$. An entity may be a person or an object in the observed area.

Events in a location sensor log do not necessarily have a unique identifier attached to identify by which entity they were triggered. Often data contains overlapping and concurrent activities by multiple entities. In smart homes or factories, multiple entities can be present at the same time. It has to be ensured that the analysed activities are all associated with the correct entity, to obtain a meaningful process model on a by-entity-level. Our method targets such scenarios where a sensor cannot identify entities utilising a unique identifier as it is the case in WiFi networks, for example.

Figure 1 gives an overview of the proposed approach, which consists of the following four steps that are explained in the following sections:

1. *Event Correlation*: Correlation of events from a location sensor event log E_L to (unlabelled) activity instances yielding an instance log E_I.
2. *Activity Discovery*: Discovery of process activities A together with their labels and sensor-level process models describing the expected behaviour on a sensor level.
3. *Event Abstraction*: Abstraction of the instance log E_I to a process event log E_P where events are directly related to the *start* or *completion* of process activities A.
4. *Process Discovery*: Process discovery based on the process event log E_P resulting in an activity-level process model defined over activities A.

3.1 Event Correlation

The first step towards process mining on raw sensor events is to group the input data according to a set of numbered activity instances by correlating each individual location sensor event $e \in E_L$ to an activity instance $i \in \mathbb{N}$. This results in an *instance log* E_I in which, beyond the requirements for E_L, each event $e_i \in E_I$ is additionally assigned an activity instance that can be retrieved with $instance(e) \in \mathbb{N}$. The main goal of this step is to produce traces such that each trace can be associated with an activity instance. We assume every recorded event in the raw data is caused by an entity. In order to determine which entity $n \in N$ caused which event, the raw event location data is assigned to the respective entities. Eventually the trace of each entity is divided into smaller sub-traces (cases) that contain only one single activity: we denote this as *sensor case slicing*. Here, also an approach for *entity detection* is required.

Entity detection. In a setting with sensors providing only information whether an object is present or absent, a distinction between entities is not possible. However, if we know the relative location of the sensors to each other, our weighted average distance approach can be implemented and distinguish between multiple entities. The very first time any of the sensors detects the presence of an entity is the beginning of the first entity's trace. For every subsequent sensor activation, we have to decide which entity caused the activation of a sensor. Each time a sensor is activated, we calculate which already registered entity is closest to the current sensor activation, based on the entities' last known position. If no entity is close enough, the algorithm assumes that a new entity has entered the observed area and creates a new trace for this new entity. Both the proximity threshold and the maximum number of entities are parameters that can be manually adjusted based on the scenario. This straightforward implementation works well if entities always keep a certain distance to others. But as soon as various entities cross paths in a spot that is only covered by a single sensor, this method will not be able to correctly assign the sensor activations after the entities moved on, since the newly activated sensor has the same distance to every entity in that single spot. This limitation can be overcome, by assuming, entities will preserve their direction of motion and predict where entities are headed by also considering the entities' previous locations combined with a decay function in the distance function.

Sensor case slicing. During its presence in the observed area, the entity executes most likely more than a single activity. To identify meaningful activities from the continuous recording (what is called a "long trace"), an appropriate separation into smaller sub-traces, called cases, is required. We have to divide the traces here, because we are identifying and clustering activities by their sensor-activation-signature, therefore the sub-traces can only contain one single activity.

In concrete terms, in our approach, a long trace is cut into sub-traces of a predefined fixed length. Depending on the application, the optimal fixed length might be different. Our implementation incorporates a grid search, comparing the results for different sub-trace lengths, to maintain flexibility. The challenge is to avoid sub-traces that are too short and contain too little sensor-data to extract meaningful activities. But at the same time, the sub-traces cannot be too long, as a too-long sub-trace may consist of multiple activities.

3.2 Activity Discovery

Having obtained the instance log E_S, we aim to infer a set of process activities A that are likely to have generated the raw sensor events. The outputs of this step in our approach are a set of activities A. Each activity $a \in A$ has both an activity label $label(a)$ as well as a process model describing the low-level behaviour of that activity a in terms of events on the sensor-level. The main challenge in this part of the approach is to determine a good division of activity instances into clusters, i.e., an *activity clustering* where each of the clusters should represent a distinct activity on the process level. This refers not only to the clustering itself but also to finding a good number of clusters. Furthermore, a suitable *activity labelling* needs to be found.

Activity clustering. Independent of the implemented clustering technique, the objective remains the same: Find similar sub-traces and group them. For this, we used a *Self-Organising Map* (SOM) clustering and k-means. The challenge with the discovery of similarities is to find a criterion to define the similarity between sub-traces. Usually, in SOM this is achieved by calculating the euclidean distance between vectors. However, this is challenging if sensors have arbitrary label names. We experimented with three alternative representations of the traces: First, we counted how often each sensor is activated in a trace. Second, we counted for how long each sensor is activated for in a trace. And third, we combined both the quantity method and time method in one vector. The third representation retains the most information of the original trace and is, therefore, the preferred choice.

Activity labelling and Validation. Having discovered clusters of similar traces corresponding to distinct activities, we still lack insights into the kind of activity that may be represented by each cluster. Also, it may be challenging to judge the quality of the obtained clustering. We assume that activity labelling generally requires a human-in-the-loop with appropriate domain knowledge. Thus, we propose to discover a process model based on the events of each cluster by using, e.g., Inductive Miner. Then, the quality of the process model is evaluated based on the F1-score combination of the common *fitness* and *precision* measure. The core idea is that these interpretable process models make our method suitable for complex processes and the quality measure can be used to validate the clustering result. Having access to the process models and their quality evaluation a domain expert can interpret, validate and label each cluster with an appropriate activity label $a \in A$.

3.3 Event Abstraction

The third step of our approach combines the sub-traces yielded by the *event correlation* step (Section 3.1) and the activity clusters detected in the *Activity Discovery* step (Section 3.2). This results in a process event log E_P that groups together events from the original location sensor event log E_L to process events $e_p \in E_P$. For each process event e_p we can obtain the following attributes: $time(e_P) \in \mathbb{N}$, $activity(e_P) \in A$, $entity(e_P) \in T$, and $transition(e_P) \in \{start, completed\}$. Thus, each process event refers to a specific high-level process activity and indicates a transition in the transactional life-cycle, i.e., whether the activity instance has been started or completed.

3.4 Process Discovery

Having promoted the raw location sensor events E_L to the level of activity instances, our process event log E_P fulfils almost all requirements for high-level process discovery. Anyway, still missing are process cases that are meaningful to our analysis goal. Identifying process cases is highly dependent on the particular circumstance. In our application scenario, we propose to focus on re-occurring behaviour of an entity starting with a specific activity (e.g., entering the smart home). Based on our event correlation step (Section 3.1), we build a separate trace for each entity. Then, the potentially very long trace referring to a single entity is subdivided into multiple traces by dividing it into separate traces each time the activity of interest occurs. To discover a meaningful process model, we have to assume that regular and routine behaviour is observable. As a starting point, an activity has to be selected that most likely will be the origin of the routine behaviour such as *entering the observed area*. Finally, an overall process model is discovered using a standard technique, e.g., Inductive Miner [26]. The final output is a process model reflecting the observed behaviour of the entities aggregated only from raw sensor data.

4 Evaluation

4.1 Set-up

We evaluated our approach on the publicly available CASAS data-set, which contains raw sensor data from a smart home environment [2]. The CASAS data fulfils the two requirements of our approach: it contains the timestamps and location information of sensor events. The data was recorded in a smart home test-bed with two residents and a house equipped with 51 motion sensors. Figure 2 shows the house plan and the positions of the motion sensors. Each motion sensor generates low-level events, where each sensor entry is tagged with a timestamp, the sensor ID and the binary sensor value (active / not active). We extracted sensor data from 7 consecutive days (02/05–09/02/2010) from the `20-Kyoto-2-Daily life, 2010-2012` data set.

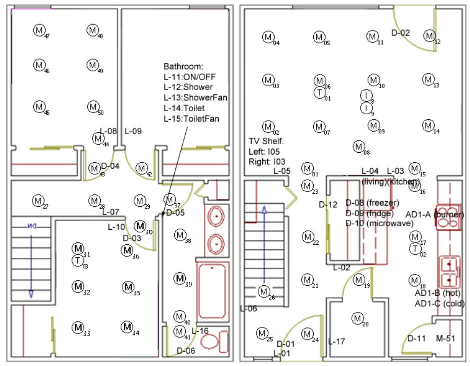

Fig. 2: Sensor layout of an apartment in the CASAS project [2].

We applied our method for different values of parameters such as sub-trace length, number of clusters and the similarity measure used. We used grid search to identify best parameter values for the clustering based on the average combined fitness [27] and precision [28] (F1-score) obtained for the process models discovered for each cluster of high-level events. We employ standard filtering techniques (most frequent traces and activities) used in process mining to focus on the dominant behaviour in each cluster. We compared the proposed SOM clustering with k-means clustering based on the same similarity measures. The implementation of step 1 and 2 is openly accessible [6] We used PM4Py 1.1.1 and heuristicmineR for the process discovery and evaluation.

Having discovered activities and obtained traces based on the idea to discover re-occurring behaviour starting with the same activity (Section 3.4), we applied Heuristics Miner to discover a process model of the behaviour. Based on the spatial layout of the smart home (Figure 2), we choose to create traces that start with the activity *Walk entrance/stairs/storage* as the entry point into the house. Heuristics Miner was selected as we expect the inhabitants of the smart home environment to show a lot of infrequent behaviour, for which Heuristics Miner has shown to be appropriate [29].

4.2 Results & Discussion

Figure 3 shows the results of our grid search. We experimented with trace lengths ranging from four to twelve. Shorter trace lengths generally lead to a better F1-score. However, we need to impose a minimal trace length since traces consisting only of a single event would trivially lead to the discovery of process models with perfect fitness and precision. In our case, less than four events did not allow to infer a set of meaningful activities.

Evaluating sample data has shown that considering both the frequency of activation as well as the duration of the activations as a similarity measure (the

[6] `https://github.com/d-o-m-i-n-i-k/Process-Model-Discovery-public`.

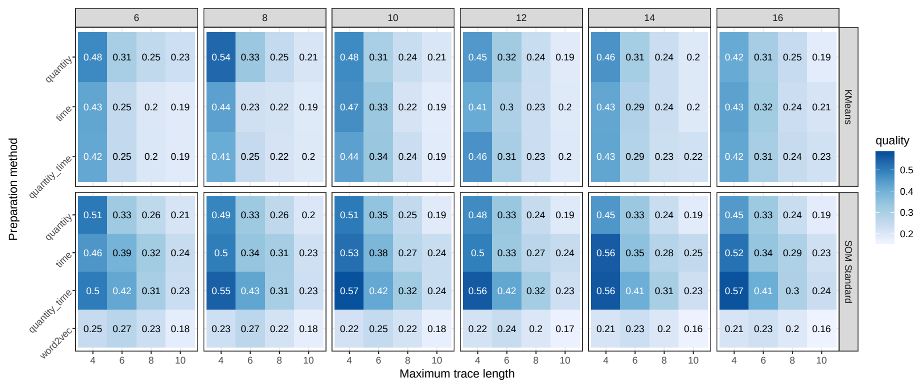

Fig. 3: Average F1-score for process models discovered for the clusters based on six different cluster sizes (6-16), five different maximum trace lengths (4-12), four vector preparation methods and two clustering algorithms.

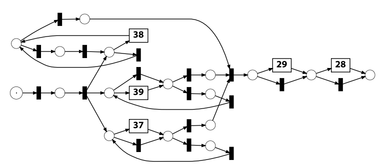

Fig. 4: Example of a Petri net discovered using Inductive Miner for a cluster in the *Activity Discovery* step.

method *quantity_time*) yields superior results, compared to only regarding one aspect. When choosing too few clusters or too many, the quality score decreases. In turn, choosing too many clusters may lead to several clusters representing the same activities, which should have been grouped. We also qualitatively evaluated the clustering by manually inspecting and labelling some of the results.

For example, the Petri net discovered by Inductive Miner on a cluster shown in Figure 4 is a reasonable candidate. The three sensors that can be activated simultaneously are all located in the bathroom. The subsequent sensors `M29` and `M28` are located in the hall with `M28`, which is furthest from the bathroom. From this example process model, it is reasonable to infer that this cluster refers to activities where the entity spends some time in the bathroom and then left the room. Overall, the similar results are obtained for 10 and 16 clusters with a trace length of 4 and using our proposed *quantity&time* vectorisation approach.

We grouped the activity instances of the best clustering results (16 clusters) into traces at the level of process instances. Afterwards we filtered the resulting event log to only retain traces of a length in the range of 5 to 25 events. This yields a log with 5898 events grouped in 273 traces with an average length of 21.6. The application of Heuristics Miner with a dependency threshold of 0.8 and a frequency threshold of 10 returns the Causal net dependencies shown in

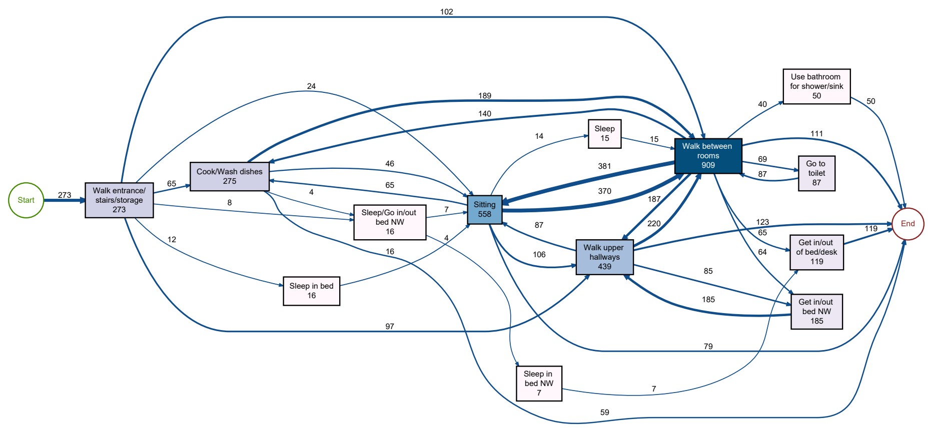

Fig. 5: Causal net discovered with the Heuristics Miner on the obtained process event log.

Figure 5. The activity in entrance area of the house marks the starting point of our Causal net. The activities that can mostly be observed after the entry activity are *walking between the rooms*, *walking in the upper hallways* and *going to the kitchen to cook or wash the dishes.* After cooking the dishes it often occurs that the resident would walk between the rooms to sit down, presumably to eat in the living room.

4.3 Limitations

A drawback of our method is the assumption of continuous movement in the event correlation step (Section 3.1). As soon as the motion at the rendezvous location is more than just a mere passing by, our approach might not return the desired results. Additionally, the entity recognition could be improved by using more sophisticated methods, e.g. hidden Markov models that have already shown promising results in differentiating people from one another [30]. Moreover, the sensor case slicing mechanism could take variable sub-trace length into consideration, i.e., depending on the activity, the number of involved events, and therefore the sub-trace length may vary. For example, the activities *sleeping*, *cooking* and *washing hands* are activities with a distinctive difference in the number of involved events.

5 Conclusion

IoT environments generate a large amount of data, predestined for further analysis. Process mining can give valuable insights into how real-life activities perform when extracting meaningful activities instances from raw sensor events. This paper combined unsupervised learning in the form of clustering and process mining, to discover activities and process models from motion sensors. We evaluated our approach by comparing the obtained model quality for several clustering techniques on a publicly available data-set in a smart home scenario and found it to be superior. To fully relieve domain experts from process modelling and to automate the process of model discovery, we believe that an accurate approach for entity centricity is imperative. For this, future tasks are to fuse heterogeneous sensor events as input for high-level aggregation, to take into account other vectorisation methods such the shortest path distance between sensors (i.e., relational or pair-wise distances only) to better disambiguate between residents and to apply non-end-to-end process discovery methods such as Local Process Model discovery [22]. In further research, we plan to include spatial information, like room layouts in smart homes, into our approach as well as implement variable trace lengths and experiment with other machine-based learning techniques to further improve the discovered process models.

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016)
2. Rashidi, P., Youngblood, G.M., Cook, D.J., Das, S.K.: Inhabitant guidance of smart environments. In: Lecture Notes in Computer Science. Volume 4551 of LNCS. Springer Berlin Heidelberg (2007) 910–919
3. Nentwig, M., Stamminger, M.: A method for the reproduction of vehicle test drives for the simulation based evaluation of image processing algorithms. In: 13th International IEEE Conference on Intelligent Transportation Systems, IEEE (September 2010) 1307–1312
4. Zhang, M., Sawchuk, A.A.: Motion primitive-based human activity recognition using a bag-of-features approach. In: Proceedings of the 2nd ACM SIGHIT symposium on IHI '12. IHI '12, ACM Press (2012) 631–640
5. Diete, A., Sztyler, T., Weiland, L., Stuckenschmidt, H.: Improving motion-based activity recognition with ego-centric vision. In: PerCom Workshops 2018, IEEE Computer Society (2018) 488–491
6. Sztyler, T., Carmona, J., Völker, J., Stuckenschmidt, H.: Self-tracking reloaded: Applying process mining to personalized health care from labeled sensor data (2016)
7. Civitarese, G.: Human activity recognition in smart-home environments for healthcare applications. In: 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). (March 2019) 1–1
8. Larue, G.S., Rakotonirainy, A., Pettitt, A.N.: Predicting reduced driver alertness on monotonous highways. IEEE Pervasive Computing **14**(2) (April 2015) 78–85
9. Weerdt, J.D., vanden Broucke, S., Vanthienen, J., Baesens, B.: Active trace clustering for improved process discovery. IEEE Transactions on Knowledge and Data Engineering **25**(12) (December 2013) 2708–2720
10. Choi, S., Kim, E., Oh, S.: Human behavior prediction for smart homes using deep learning, IEEE (August 2013) 173–179
11. Gallicchio, C., Micheli, A.: Experimental analysis of deep echo state networks for ambient assisted living. In Bandini, S., Cortellessa, G., Palumbo, F., eds.: Proceedings of the Third Italian Workshop on AI for Ambient Assisted Living. Volume 2061 of CEUR Workshop Proceedings., CEUR-WS.org (2017) 44–57

12. Wang, A., Chen, G., Shang, C., Zhang, M., Liu, L.: Human activity recognition in a smart home environment with stacked denoising autoencoders. Volume 9998. (June 2016) 29–40
13. Wang, J., Chen, Y., Hao, S., Peng, X., Hu, L.: Deep Learning for Sensor-based Activity Recognition: A Survey. arXiv e-prints (2017)
14. Lee, S., Lin, F.J.: Situation awareness in a smart home environment. In: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT). (December 2016) 678–683
15. Dernbach, S., Das, B., Krishnan, N.C., Thomas, B.L., Cook, D.J.: Simple and complex activity recognition through smart phones. In: Intelligent Environments, IEEE (2012) 214–221
16. Sztyler, T.: Sensor-based human activity recognition: Overcoming issues in a real world setting. PhD thesis, University of Mannheim, Germany (2019)
17. Nguyen, H.D., Tran, K.P., Zeng, X., Koehl, L., Tartare, G.: Wearable sensor data based human activity recognition using machine learning: A new approach. ArXiv **abs/1905.03809** (2019)
18. Koschmider, A., Mannhardt, F., Heuser, T.: On the contextualization of event-activity mappings. In: BPM 2018 Workshops. LNBIP, Springer (September 2018)
19. van der Aa, H., Leopold, H., Reijers, H.A.: Checking process compliance on the basis of uncertain event-to-activity mappings. In: Advanced Information Systems Engineering. Springer International Publishing (2017) 79–93
20. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.: Mining process model descriptions of daily life through event abstraction. In: Intelligent Systems and Applications. Springer International Publishing, Cham (December 2017) 83–104
21. Leotta, F., Mecella, M., Sora, D.: Visual process maps: a visualization tool for discovering habits in smart homes. Journal of Ambient Intelligence and Humanized Computing **11**(5) (January 2019) 1997–2025
22. Mannhardt, F., Tax, N.: Unsupervised event abstraction using pattern abstraction and local process models. In: RADAR+EMISA@CAiSE. Volume 1859 of CEUR Workshop Proceedings., CEUR-WS.org (2017) 55–63
23. Alharbi, A., Bulpitt, A., Johnson, O.: Towards unsupervised detection of process models in healthcare. Studies in health technology and informatics **247** (January 2018) 381–385
24. Tax, N., Alasgarov, E., Sidorova, N., Haakma, R., van der Aalst, W.M.P.: Generating time-based label refinements to discover more precise process models. Journal of Ambient Intelligence and Smart Environments **11**(2) (2019) 165–182
25. Brzychczy, E., Trzcionkowska, A.: Process-oriented approach for analysis of sensor data from longwall monitoring system. In: Advances in Intelligent Systems and Computing. Springer International Publishing (August 2018) 611–621
26. Leemans, S.J.: Robust Process Mining With Guarantees. PhD thesis (2017)
27. van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.F.: Replaying history on process models for conformance checking and performance analysis. WIREs Data Min Knowl Discovery **2**(2) (2012) 182–192
28. Munoz-Gama, J., Carmona, J.: A general framework for precision checking. International Journal of Innovative Computing, Information and Control (IJICIC) **8**(7) (2012) 5317–5339
29. Augusto, A., Conforti, R., Dumas, M., Rosa, M.L., Maggi, F.M., Marrella, A., Mecella, M., Soo, A.: Automated discovery of process models from event logs: Review and benchmark. IEEE TKDE **31**(4) (April 2018) 686–705
30. Guo, P., Miao, Z.: Multi-person activity recognition through hierarchical and observation decomposed HMM. In: 2010 IEEE International Conference on Multimedia and Expo, IEEE, IEEE (July 2010) 143–148