IoT-Based Activity Recognition for Process Assistance in Human-Robot Disaster Response

Adrian Rebmann[1,2], Jana-Rebecca Rehse[1,3], Mira Pinter[1,2], Marius Schnaubelt[4], Kevin Daun[4], and Peter Fettke[1,2]

[1] German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
{firstname.lastname}@dfki.de
[2] Saarland University, Saarbrücken, Germany
[3] University of Mannheim, Mannheim, Germany
rehse@uni-mannheim.de
[4] Technical University of Darmstadt, Darmstadt, Germany
{lastname}@sim.tu-darmstadt.de

Abstract. Mobile robots like drones or ground vehicles can be a valuable addition to emergency response teams, because they reduce the risk and the burden for human team members. However, the need to manage and coordinate human-robot team operations during ongoing missions adds an additional dimension to an already complex and stressful situation. BPM approaches can help to visualize and document the disaster response processes underlying a mission. In this paper, we show how data from a ground robot's reconnaissance run can be used to provide process assistance to the officers. By automatically recognizing executed activities and structuring them as an ad-hoc process instance, we are able to document the executed process and provide real-time information about the mission status. The resulting mission progress process model can be used for additional services, such as officer training or mission documentation. Our approach is implemented as a prototype and demonstrated using data from an ongoing research project on rescue robotics.

Keywords: Rescue Robotics · Process Assistance · Internet of Things · Activity Recognition · Emergency Process Management

1 Introduction

In any natural or man-made disaster, first response teams have to save and protect humans, animals, assets, and the environment. These responsibilities, along with a lack of real-time information, public interest, the need to act quickly, and an often hectic and confusing situation, put officers under a high amount of pressure [3]. To reduce both the mental burden and the physical danger to the officers, rescue robots are increasingly used to support disaster response operations [17]. Ground vehicles, for example, can transport heavy equipment or search for victims in areas that would be too dangerous for human officers to enter, while drones may take aerial pictures of a disaster site, which can help in identifying victims or finding access routes in unsafe territory [18].

However, integrating rescue robots into first response teams adds an additional dimension to an already complex and stressful situation, increasing the cognitive load of the officers in charge [16]. One way to support the officers in this situation is to provide them with real-time information about the status of the mission [29]. For this purpose, Business Process Management (BPM) approaches are particularly interesting, because disaster response missions are organized in a process-oriented way [7]. The actions of both officers and robots during an emergency mission follow predefined processes that are well documented and frequently practiced. Therefore, BPM methods can be used to capture and visualize ongoing processes for live-mission monitoring and to document them for after-mission reporting and debriefing. However, since the execution of disaster response processes depends on the concrete circumstances of a mission, any useful BPM solution requires real-time data about the executed activities.

For human first responders, we designed a system that interprets verbal communications, recognizes activities, structures them into processes, and visualizes the processes to provide assistance to the officers [30]. Since radio communication does not provide a complete status of the mission, there is a need for additional data sources. (Teleoperated) Rescue robots are equipped with sensors that record data about themselves and their surroundings, such as the robot's position, acceleration, or direction of movement. This sensor data can be used to recognize the activities that the robot operator executes during a mission.

In this paper, we design, develop, and evaluate an approach for the real-time recognition of robot activities that can be used for process assistance during robot-assisted disaster response. We define an ad-hoc process with a set of activities, which are executed as the mission status requires. Using a machine learning approach, we automatically recognize the ongoing activity from sensor data collected during execution. These activities are mapped to an instance of the ad-hoc process and stored in an event log, which is visualized to provide real-time information on the mission status to the officer in charge. In addition, this system supports the robot operator, as it documents all their activities during a mission. The collected data can afterwards be used for automatically generating a mission documentation report, as illustrative material for training new robot operators, or to improve the processes by means of post-mission process mining.

Our research is objective-centered, motivated by the opportunity to use available sensor data to provide process assistance. In this paper, we focus on detecting activities by a teleoperated ground robot in a specific application scenario described in Sect. 2. Sections 3 and 4 describe the design and development of our activity recognition approach, which is demonstrated and evaluated in Sect. 5. We report on related work in Sect. 6, before concluding the paper in Sect. 7.

2 Application Scenario

Automatic activity recognition depends on the capabilities of the robot and the situation in which it is used. In this paper, we focus on the application scenario of a medium-sized unmanned ground vehicle (UGV) in an industrial fire, where

Name	Search for unconscious human and take gas probe after industrial fire
Description	After an industrial fire, the robot UGV is sent into the factory to inspect a gas tank. Once the tank is found, the robot takes a gas probe to find out whether it was damaged by the fire and might therefore be leaking dangerous gases. After taking the probe, the robot searches for a missing human that could be unconscious due to the fire or any dangerous fumes. Once the human is detected, the robot returns to the base. Based on the result of the probe, first responders know if they can safely enter the building to rescue the unconscious human or if any additional steps must be taken beforehand.
Actors	UGV, Unconscious victim
Preconditions	A factory has burned down. A potentially damaged gas tank is located inside a building and a human is lying unconsciously on the floor of the same building.
Postconditions	The human is detected, a gas probe was taken, and UGV has arrived back at the base.
Main Success Scenario	1. UGV is navigated across the factory site to approach the entrance in a direct path. Once there, it enters the building. 2. UGV is navigated inside the room to search for the gas tank. 3. The operator detects the gas tank by classifying it on the camera picture. 4. UGV is navigated to the gas tank. 5. A gas probe is taken next to the tank, by raising UGV's sensor arm and holding it close to the tank. After taking the probe, the arm is lowered into the standard position. 6. UGV is navigated through the building to search for an unconscious human. 7. The operator detects an unconscious human by classifying it on the camera picture. 8. UGV is navigated towards the human so the operator can determine his/her health status. 9. UGV is navigated back to the base.

Fig. 1. Application scenario of an industrial fire formulated as a use case

using a robot is particularly beneficial. Since production processes often require highly explosive or flammable materials, they are stored within the plants, making industrial fires both dangerous and frequent. The American National Fire Protection Association reports that between 2011 and 2015 municipal fire departments in the U.S. responded to an average of 37,910 industrial fires every year. Annual losses from these fires are estimated at 16 civilian deaths, 273 civilian injuries, and $1.2 billion direct property damage [2]. At the same time, those fires can be challenging for first responders. For instance, gas tanks stored within a plant might be damaged by a fire, making it dangerous to enter the building without proper safety measures. In this case, a robot can be sent into the building to locate the gas tank of interest and take a gas probe to determine if any gas has leaked. Furthermore, the robot can search for any victims that might be trapped or unconscious to communicate their position to a rescue team.

To illustrate the application possibilities of the concept and implementation of the artifact presented in this paper, we propose the use case depicted in Fig. 1 in the form of a modified version of the Use Case Template according to [4].

3 Conceptual Design

3.1 Outline

Our approach for automatic activity recognition consists of several steps to overcome an abstraction gap from low-level sensor data to a top-level process. These sensor data are obtained from sensing devices, which can communicate over communication networks, including the Internet, i.e. the Internet of Things (IoT)[8].

Given the set of available activities, we can use machine learning to automatically recognize the ongoing activity from fine-grained sensor data acquired from UGV's on-board sensor unit. Therefore, we train a classifier to assign a sequence of sensor data to one of the activity types, as described in Sect. 3.3.

Conceptually, the set of relevant robot activities constitutes an ad-hoc process, i.e., a process whose control flow cannot be modeled in advance, but instead is defined at run-time [5]. This is because the robot operator has to adapt his actions to the terrain and the needs in a new and unforeseeable emergency situation. This ad-hoc process is the basis for our process assistance tool, as described in Sect. 3.4, where we log and visualize each process execution as a sequential process instance. As a post-processing step, process discovery methods can be applied to the recorded event log, allowing insights into how the robot operators are performing their tasks and enable process improvements based on that.

3.2 Activity Type Definition

For our approach, we assume activity types that are defined a priori. Those activities are derived from the robot's capabilities within the application scenario and based on the human operators' understanding of high-level activities (e.g. "Search") instead of the low-level capabilities of the robot (e.g. "Drive Forward"), which are components of the higher-level activities. We aim at directly recognizing the activities that are relevant to the process. In this way, a direct mapping between activity and task in a process model is achieved. A different approach would be recognizing the robot's capabilities and deriving higher-level activities, which are only subsequently mapped onto a process model. Both approaches however, require prior knowledge of higher-level activities.

3.3 Recognizing Activities from IoT data

Sensor data, like those acquired from the UGV, are frequently recorded representations of some kind of state. A state could be a temperature, a velocity, a level of humidity, a position, or a distance to a specific point in a space. The main challenge is to assign those fine-grained state representations to activities, more precisely to instances of activity types on a process-relevant abstraction level. Those activities commonly occupy larger time frames than the ones between two sensor captures. Therefore, time frames of sensor data have to be assigned to activity instances. This can be modeled as

$$\langle s_1, \ldots, s_n \rangle \rightarrow a \tag{1}$$

where $s_{1,\ldots,n}$ are n data points captured by an individual sensor s and a is one of the given activity types. This solves an individual assignment problem between sensor data and activity type. In order to train a machine learning model, we need a sufficient amount of labeled input data, which is considered as a great challenge in detecting activities, due to its high effort [11]. In our case, we manually label the sensor data that belong to certain activity instances with the corresponding activity name. In the described application scenario, it was sufficient to rely on a single sensor as a data source to distinguish all relevant activity types. Limiting ourselves to one sensor with low payload per measurement, can also prevent problems such as transfer speeds from the mobile robot to other processing components, especially since data transmission can become difficult in an emergency situation. A scenario, where multiple sensors of the same type or even multiple types of sensors are necessary to fully capture all relevant activities, would require a mechanism for synchronizing, aggregating, filtering, and merging sensor data, making a reliable process assistance much more challenging.

3.4 Providing Ad-Hoc Process Assistance

After training a machine learning classifier for activity recognition, we can use it to recognize, document, and visualize the robot's actions in real time. Therefore, we model UGV's behavior as an ad-hoc process, as shown in Fig. 2. Within an instance of this sub-process, all tasks can be executed arbitrarily many times and in any order. Readers should note that the application scenario and the activity names indicate a partial ordering, which is not represented in the process model. For example, *Navigate Back to Base* appears to be the last activity in a trace. A prior activity filtering based on a more structured process model is also interesting and will be addressed in future work.

The process model is the basis for our process assistance system. We connect the activity recognition to a process model repository and process execution engine, where the model is instantiated at the beginning of a robot mission. The process execution engine is controlled by the activity recognition, such that the process status is continuously captured and automatically recorded in an event log. For example, whenever the recognition approach detects an activity instance that differs from the currently active activity instance, it triggers the completion of the active one and starts the newly detected one. With a continuous pipeline from the robotic sensors to the activity recognition and process execution engine to an event log, we are now able to provide process assistance to the first responders. The active process instance is visualized, providing process assistance to the operator, as well as real-time information on the mission status to the officer in charge. After completing a mission, the logged event data captures all actions of the robot in sequence and can be used for post-mission process mining, debriefing the mission participants, improving the mission processes and protocols, as well as training new officers [30].

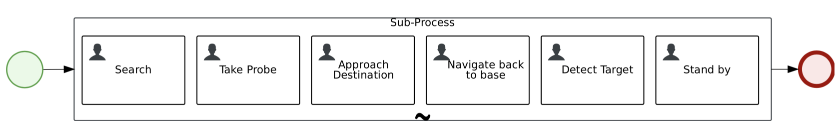

Fig. 2. Ad-hoc process for the behavior of UGV, modelled as a BPMN diagram

4 Implementation

In order to implement the conceptual design, we created an overall system architecture that is easily extendable by new components, shown in Fig. 3.

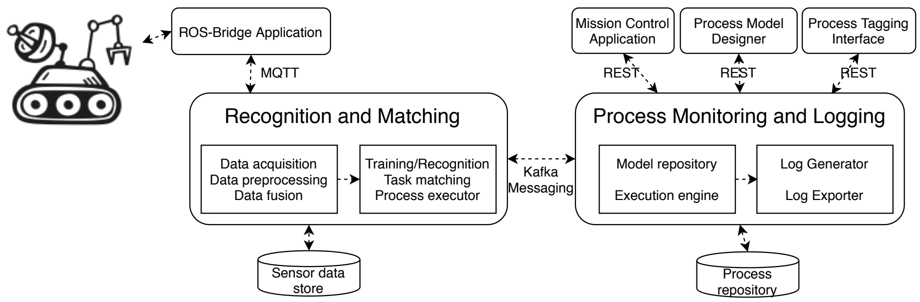

Fig. 3. System architecture and relevant connected components.

Data Acquisition. The UGV collects its IoT data using the Robot Operating System (ROS)[5], which uses a proprietary data format and is organized in topics, similar to Kafka[6]. Both systems implement a publisher subscriber pattern. *Topics* or *Channels* are categories or feed names to which records are published and subscribed. We implemented a bridge application that consumes sensor data from topics relevant for task recognition, translates them into a format, which is simpler to process by our system, and re-publishes them to an MQTT[7] broker. From there, other components of the system, in particular recognition and matching, can subscribe to the data channels relevant to them.

The robot collects a variety of different sensor data. For our goal of recognizing tasks on the process-level of an operator, we assume motion data to be most discriminative. Therefore, we base our implementation on the data collected by the Inertial Measurement Unit (IMU) sensor, attached to the base of the robot. Using a combination of accelerometer and gyroscope, the IMU measures linear accelerations, angular velocities, and the orientation for each 3D-axis at high frequency. Each of the three quantities has three dimensions yielding a 9D-measurement in total. Even though the IMU is located at the base of the robot, operating actuators induce small motions, which are recognized by the IMU and allow for conclusions about the state of the manipulator arm.

Data Pre-processing. The recognition approach essentially consists of a continuous multi-class classification using sliding windows of pre-processed IoT data streams. These classification results need to be mapped to pre-defined activity types. Therefore, we define a fixed window size and compute feature vectors

[5] `https://www.ros.org`
[6] `https://kafka.apache.org`
[7] `http://mqtt.org`

based on segments of sensor time series. Then, we classify each segment individually. To ensure that no boundaries of activity instances are missed, we use segments overlapping by 50%, as proposed by [26]. Thus, given an individual sensor s and segments consisting of t points in time when sensor data is captured (segment size), the sensor time series data are segmented as follows:

$$\begin{array}{lllllll} segment_1: & s_1 & ... & s_t & & & \\ segment_2: & & s_{t/2} & ... & s_{3t/2} & & \\ segment_3: & & & s_t & ... & s_{2t} & \\ \vdots & & & & \ddots & \ddots & \ddots \end{array} \tag{2}$$

In order to achieve accurate classification results, we compute representative features of the raw sensor data and use them as input for the classification. We consider each dimension of the IMU measurements as a separate time series and compute the mean, standard deviation, root mean squares, kurtosis, skewness, interquartile range, and mean absolute deviation [10,1]. Besides those statistical features, frequency domain features are used as additional input because they provide additional insights. Specifically, we include the energy, entropy, and signal magnitude area for each time series as features. Moreover, for all time series within each quantity, we include the pairwise correlation.

Training. After the sensor data is pre-processed, segmented, and labeled with the activities from Fig. 2, it can be used to train the activity recognition. Therefore, we need a machine learning approach that allows for multi-class classification. We selected a Random Forest, as it produced promising results in past applications [21] and does not require vast amounts of training data, as opposed to, e.g., neural networks. The Random Forest classifier we finally deployed was trained as an ensemble of 200 trees with a maximum tree depth of 16. As entropy measure, we used the Gini impurity. The sensor data used for training is stored in a relational database as shown in the system architecture.

Recognition. While a disaster response process is executed, the trained model is queried continuously. Whenever a segment of the defined size is completed, we apply the same pre-processing as for the training data. Since segments are overlapping by 50%, the last half of the previous segment becomes the first half of the current segment for classification. To complete the activity recognition, the classification outcome needs to be mapped to the process model tasks, as stored in the process model repository. The available activity instances as published by the process execution engine are matched with the classification results of the activity classifier. This is done on the basis of the activity label, which in our case is identical to the respective activity label in the process model. Based on the activity label, life-cycle transitions of activity instances of the ad-hoc process can be controlled. For example, if an activity instance is currently active but the classification yields a different activity, the active activity instance is completed and the detected activity instance is started. This enables the process assistance.

Process Monitoring and Assistance. The process assistance is based on the deployed process execution engine, which instantiates processes at the beginning of a mission and monitors their execution depending on the results of the activity recognition. Therefore, the process engine bridges the gap between the activity recognition and the ad-hoc process execution. We used the open source process engine from jBPM[8] for this purpose. Accessing the results of all previous processing steps, we implemented a status monitoring visualization as a web-based application, which can be shown on any display. The app depicts an integrated view of the current mission status with all active and completed disaster response processes. Aggregating information about each process gives the officers in charge a better overview of the overall situation, as well as information about the progress in critical processes, such as reconnaissance.

5 Evaluation

5.1 Evaluation Approach

To ensure repeatable experiments and efficient data capturing, we train and evaluate our approach using data generated by a physics-based simulation of the UGV. The simulation is based on the open-source framework Gazebo[9], which includes accurate simulation of multi-body dynamics and realistic sensor data generation for common robotic sensors. We use a digital twin of the Telerob Telemax Hybrid UGV, which is deployed by the German Rescue Robotics Center.[10] The simulation exactly mirrors the real robot with respect to the software interface and the available sensors.

Based on this, we performed a two-stage evaluation. For the first step, we gave the use case description from Fig. 1 to ten case study participants and asked them to execute the described reconnaissance process using the simulation application. During the execution of each individual process instance, sensor data of the simulated robot were produced and consumed by our system. Simultaneously, the performed activities were manually labeled with the respective activity type. Based on this data, we trained the machine learning model as introduced in Sect. 4 and derived per-class accuracies as well as overall accuracies of the recognition. Using the trained and evaluated model, we performed a second case study to validate the process assistance system in real-time. Five study participants were asked to execute a reconnaissance sub-process in the same setting as the first case study. In order to measure the validation accuracy, we compared the actually executed activities to the ones the system detected and visualized in the developed mission control application. The use case executed in the simulation application with the running mission control app can be viewed in [22].

5.2 Case Study Implementation

While the study participants controlled the robot, we used a web-based tool (also depicted in Fig. 3) to tag the start and completion of process and activity

[8] `https://jbpm.org`
[9] `http://gazebosim.org/`
[10] `https://rettungsrobotik.de`

instances. Available instances were determined according to the respective process model and the process instance's current state. This way, labelling was done implicitly via timestamps. Every recorded sensor data point was saved with the timestamp of its creation. Then, the activity time frames were used to label the data. We acquired a total of 45 process instances for training and testing including different variants of activity orders. Next, the data was prepared according to Sect. 3. The recorded and labeled data were used to train a random forest classifier, which is capable of performing multi-class classification as well as producing probability distributions over all possible classes for each unlabeled feature vector. The model was trained using different segment sizes (4-8-16). Segment size 16 yielded slightly better results, but since larger segments take longer to complete, this resulted in longer intervals between classifications and therefore a longer latency in the process assistance. We therefore chose to use a rather small segment size of 4, which performed comparably well and has the advantage of generating more training data. We initially evaluated each trained model using 20% of the collected data as test data. This classifier was then used for the overall evaluation in the second case study. Participants controlled the robot, performing both the same use case as for the training data acquisition and a different variant to show that the classifier can handle different orders of activities. For that second variant, we asked them to execute the original scenario, but since the human could be detected immediately, participants skipped the search step and directly approached the human. Afterwards, they searched for the gas tank, took a probe, and approached the tank to examine it visually. This was followed by another search activity to detect any other possible victims. 14 evaluation instances were captured, while the system was running. The efficacy of the approach can be shown by comparing the process instances produced by the system to the actual process execution.

5.3 Results

Activity Classifier Performance. Fig. 4 shows the results of the individual performance evaluation of the activity classifier, which was trained on the sensor data of the previously recorded training process instances. We split the data into 80% training data and 20% test data. Additionally, Fig. 4 shows the evaluation metrics (precision, recall and f1-score) for the classifier evaluation. The results indicate that the classifier is capable of differentiating the robot's activities very accurately. A close look at the individual mis-classification revealed that confusions mainly occur because *Approach destination* is often classified mistakenly and therefore produces false positives. Also, the classifier mostly fails to correctly classify *Standby*, however, there are no false positives. Even in this case, the performance is sufficient to accurately determine the activities given the selected segment size and overlap. It should also be considered that standby times hardly occurred in the process executions of our case study, which is why this activity is underrepresented in training and test data. The activity *Detect target* is not considered in the classifier evaluation, as it is a point-based activity, which can be recognized using image data rather than motion data.

	precision	recall	f1-score
A	0.84	0.95	0.89
B	0.70	0.74	0.84
C	1.00	0.51	0.68
D	0.73	0.87	0.79
E	0.97	0.81	0.88

Fig. 4. Normalized confusion matrix (left) and classification metrics (right) for the activity classifier per class (A=Take probe; B=Approach Destination; C=Stand by; D=Search; E=Navigate back to base)

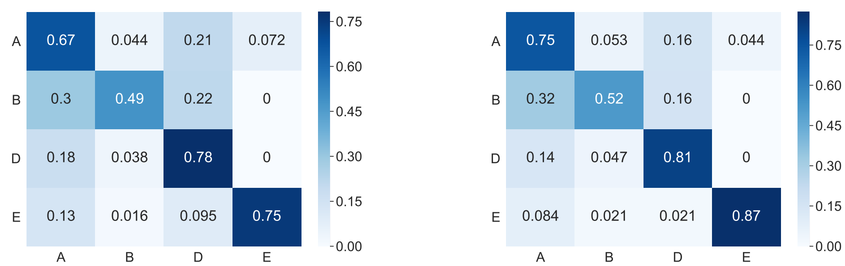

Fig. 5. Normalized confusion matrices showing all classified segments, including (left) and excluding (right) activity transitions (A=Take probe; B=Approach Destination; D=Search; E=Navigate back to base)

Overall System Performance. To evaluate the overall efficacy of the approach and the implemented artifact, we compare instances produced by the system in the form of an event log with the actual activities the robot performed, while being controlled by the case study participants. Fig. 5 shows normalized confusion matrices of the classifications of the overall evaluation. The left matrix shows the performance of all classifications throughout all use case instances, whereas the right one displays all classifications that were executed as long as no activity transition took place. Label C (Stand by) is not depicted, as this activity was never executed during the overall system evaluation.

5.4 Discussion

The evaluation shows that our approach can monitor and capture process execution. The individual classifier performs well for all activities in our use case, despite limited training data, using a simple random forest classifier, and a single sensor as data source. Based on this classifier, we can map the results acquired

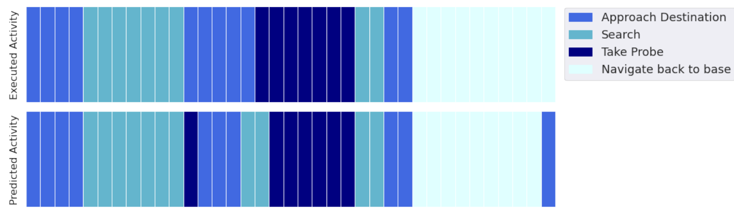

Fig. 6. Single overall execution instance of the scenario, with actual labels in the top row and predicted labels in the bottom row

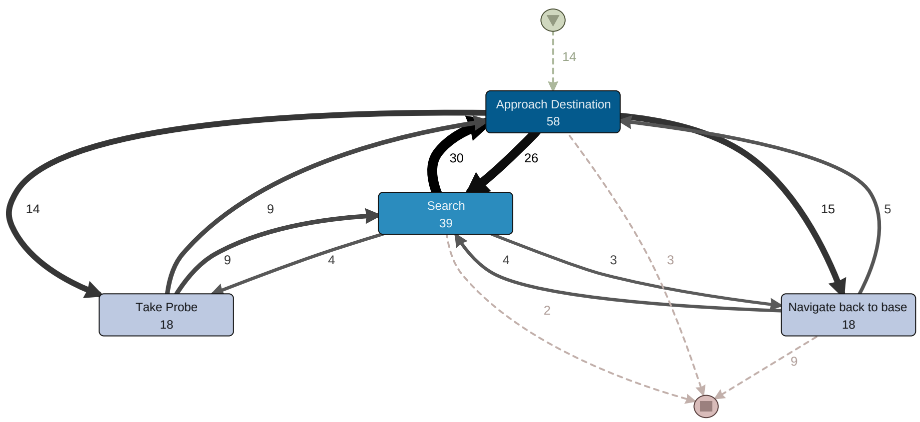

Fig. 7. Process map generated from the process instances that were recorded by the execution engine during the overall system evaluation (100% activities; 100% paths)

in an online mode to tasks of our reconnaissance sub-process and control the process execution. As expected, the recognition accuracy is better for segments of the sensor data that are completely contained in an activity time frame. To further support this observation, Fig. 6 shows a single execution instance of our scenario comparing actual labels with predicted ones, illustrating that misclassifications mainly occur at activity transitions. To demonstrate post-mission process mining, we exported the event log produced during the overall system evaluation and used Disco to generate a process map, as shown in Fig. 7. For this log, subsequent identical events were merged, such that each recognized activity was logged only once. The process map shows that despite the ad-hoc process model, there is a clear structure to the executed process. For example, the first activity was always *Approach Destination* and *Take Probe* is always an intermediate activity. Other than expected, *Navigate back to base* was not always the last activity in the recorded cases. Overall, this discovered process model might give first hints to first responders regarding the recommended execution of a reconnaissance process.

Limitations The conducted evaluation is limited due to rather few cases available for training. However, regarding the number of training samples, the availability of the simulation application allowed for obtaining far more data than we could have obtained using the actual robot. Moreover, in this case study, only two scenarios have been carried out. Future work will address these limitations.
Reproducibility. Our data repository[11] contains an export of the PostgreSQL database, which stores the IMU data as well as the process instance and respective activity instance data that we collected in the case study and used for the evaluation. This data can be used to reproduce our classification approach. Unfortunately, the simulation environment and our process assistance implementation cannot be made available due to licensing problems.

6 Related Work

Human Activity Recognition. The robot used in this paper could operate at least semi-automated, but currently, all actions are remotely controlled by a human operator, because disaster response missions require absolute dependability. Hence, our approach is related to Human Activity Recognition (HAR) in disaster response, as for example Lieser et al. present [12]. They describe an approach for situation detection based on activity recognition using wearable devices. Specifically, they detect people in disaster related situations by recognizing their physical activities using two smartwatches and three smartphones. Although the approach uses machine learning for HAR, an integration with a process-aware assistance system is not provided. The authors of [20] apply HAR based on data from wearable sensors with the goal of providing workers with ergonomic recommendations during the execution of manual commissioning processes. The approach allows for real-time monitoring, however, it uses predefined process sequences and does not consider ad-hoc behaviour. In their literature analysis, Mannhardt et al. give a comprehensive overview of approaches that employ activity recognition on sensor data, with the goal of enabling process discovery [13]. However, the focus is specifically on industrial settings.
Event abstraction. Our approach aims at abstracting low-level events in the form of sensor data to tasks in a disaster response process. Therefore, it is related to a line of work in process mining that aims to identify high-level activities from low-level event logs. For example, Tax et al. present an approach for the abstraction of events in an XES event log based on supervised learning [27]. It consists of a feature representation of an XES event and a learning step based on a conditional random field. The authors evaluated their approach using, among others, a data set of sensor data from a smart home. Compared to our approach, however, no online monitoring is performed. In [28], a machine learning framework for log abstraction is proposed. It consists of two phases, log segmentation and classification. The approach is evaluated using a low-level SAP log and focuses on existing event logs in a narrower sense. Therefore, the applicability in

[11] `https://github.com/a-rebmann/iot-processassistance`

IoT scenarios needs to be investigated. Mannhardt et al. present a method for event abstraction based on behavioral patterns [14]. Their goal is to create an abstracted log by matching low-level event logs with activity patterns. The user has to create a model for each high-level activity to be able to use the approach. Futhermore, analyses are conducted on historical data only. Compared to this approach, we focus on real-time monitoring of ad-hoc processes, with the aim of giving timely process assistance to relevant stakeholders. An approach that transforms sensor data, in particular real-time location system data (RTLS), into event logs is presented by the authors of [24]. Interactions are used as an additional knowledge layer to close the gap between sensor data and process instances. The approach was implemented and evaluated using simulated event logs and is focused on RTLS. Process assistance is not targeted.

Process Management in Rescue Robotics. From a process management view, disaster response missions are an interesting but challenging field. On the one hand, both the team structure and the processes, including communications and actions, are predefined and strictly followed. In many practice sessions, team members internalize all rules such that they are the most efficient during a mission. Therefore, there have been many attempts at applying business process management (BPM) methods in disaster response, including, e.g., [7,6,9]. On the other hand, each disaster situation is unique on its own. No one can foresee which processes have to be executed in which kind of environment. These decisions have to be made on-site and in real time, often without knowing the entire situation. Researchers have argued that this unforeseeable nature of disaster missions made conventional process management methods ineffective [19]. Our approach deviates from previous research, as we do not attempt to plan disaster response processes, but instead take a data-driven bottom-up approach for managing them in real-time [30].

BPM and IoT. The authors of [8] postulate a series of challenges for integrating BPM and IoT. The paper at hand addresses the specific challenge of *Dealing with unstructured environments*, in addition to the challenges *Detecting new processes from data* and *Monitoring of manual activities*. Related work exists in the discovery of models of daily routines, where activity recognition is embedded into a BPM context. The authors of [25] use an existing sensor log from a smart space to mine processes of human habits, which they depict using process modeling techniques. Presence Infrared sensors were used as a single data source. In [21] the development and evaluation of an approach which recognizes and logs manually performed assembly and commissioning activities is presented. The goal is to enable the application of process discovery methods. The presented system uses a body area network, image data of the process environment and feedback from the executing workers in case of uncertainties. In [23] an architecture is proposed that integrates IoT and BPM. The authors present a provenance framework for IoT data and consider IoT objects in the process model. The system architecture is implemented and demonstrated, e.g. in a use cases in the production industry using wearable devices. The authors of [15] present an autonomous monitoring service for inter-organizational processes. Using this approach, human operators e.g. from service providers do not need to notify about the execution of their tasks. An extension of the Guard–Stage–Milestone (GSM) makes it possible to monitor the process even if the control flow is not strictly followed.

7 Conclusion and Outlook

In this paper, we presented an approach for process assistance based on IoT data collected during a robot's reconnaissance run during a disaster response mission in the application scenario of an industrial fire. We presented the design, implementation, and evaluation of a system for the real-time recognition of robot activities. The robots' set of capabilities defines a set of activities, which can be mapped to tasks in an ad-hoc process models. Based on this, the process assistance system captures process instances of the disaster response operation.

As our evaluation shows, process behavior can be monitored accurately using a single activity classifier. The classifier itself is trained on sensor data of rather few process instances, captured from a single sensor mounted on the robot. It can therefore be said that we presented a light-weight approach with respect to training effort as well as to the required capacity for the transmission of sensor data. We showed that the approach allows for the IoT-based capturing of unstructured processes, which can otherwise not be captured by information technology. Yet, there are many possible starting points for future research. First, the machine learning model used for activity recognition can be improved either by further optimizing segmentation and overlapping ratios of the sliding windows or by training different model types. In order to preserve an efficient data transmission, edge computing scenarios could be investigated. All data pre-processing or even model training and querying could be done on a machine deployed on the robot. Then, only the results need to be sent to a server to enable process assistance. Second, our system has been tested and evaluated using a simulation application to mimic real-life reconnaissance processes. Although the application is an accurate virtual representation, the results need to be validated in the field. This would also allow to test the approach with respect to environmental factors and phenomena such as lossy data streams happening due to connection problems between the robot and the other components.

Furthermore, the evaluation as it was conducted here, aimed at showing the efficacy of the presented IoT-based process monitoring and logging approach. We did not evaluate the acceptance of target users, which is required before introducing the system to real disaster response missions. In addition, incorporating further application scenarios would allow to show the generalizability of the approach. This should not only be limited to including other types of robots into the scenario, but different types of missions should be considered as well.

Acknowledgements The research results presented here were partially developed within the research project A-DRZ (Grant No.: 13N14856), funded by the German Ministry for Education and Research (BMBF).

References

1. Bulling, A., Blanke, U., Schiele, B.: A tutorial on human activity recognition using body-worn inertial sensors. ACM Computing Surveys **46**(3), 1–33 (2014)
2. Campbell, R.: Fire in Industrial or Manufacturing Properties. Tech. rep., National Fire Protection Association (2018), `https://www.nfpa.org/News-and-Research/Data-research-and-tools/Building-and-Life-Safety/Fires-in-US-Industrial-and-Manufacturing-Facilities`
3. Carver, L., Turoff, M.: Human-computer interaction: the human and computer as a team in emergency management information systems. Communications of the ACM **50**(3), 33–38 (2007)
4. Cockburn, A.: Basic use case template. Humans and Technology, Technical Report **96** (1998)
5. Dorn, C., Burkhart, T., Werth, D., Dustdar, S.: Self-adjusting recommendations for people-driven ad-hoc processes. In: Hull, R., Mendling, J., Tai, S. (eds.) Business Process Management. pp. 327–342. Springer (2010)
6. Gašparín, M.: Identification and description of processes at the operational and information centre of the fire and rescue service of the czech republic. Quality Innovation Prosperity **19**(1), 1–12 (2015)
7. Hofmann, M., Betke, H., Sackmann, S.: Process-oriented disaster response management: A structured literature review. Business Process Management Journal **21**(5), 966–987 (2015)
8. Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A., Di Ciccio, C., Gal, A., Kannengiesser, U., Mannhardt, F., Mendling, J., Oberweis, A., Reichert, M., Rinderle-Ma, S., Song, W., Su, J., Torres, V., Weidlich, M., Weske, M., Zhang, L.: The Internet-of-Things Meets Business Process Management: Mutual Benefits and Challenges. arXiv preprint arXiv:1709.03628 (2017)
9. Kittel, K., Sackmann, S., Betke, H., Hofmann, M.: Achieving flexible and compliant processes in disaster management. In: Hawaii International Conference on System Sciences. pp. 4687–4696. IEEE (2013)
10. Lara, O.D., Labrador, M.A.: A Survey on Human Activity Recognition using Wearable Sensors. IEEE Communications Surveys & Tutorials **15**(3), 1192–1209 (2013)
11. Lavania, C., Thulasidasan, S., LaMarca, A., Scofield, J., Bilmes, J.: A weakly supervised activity recognition framework for real-time synthetic biology laboratory assistance. In: International Joint Conference on Pervasive and Ubiquitous Computing. pp. 37–48. ACM (2016)
12. Lieser, P., Alhamoud, A., Nima, H., Richerzhagen, B., Huhle, S., Böhnstedt, D., Steinmetz, R.: Situation detection based on activity recognition in disaster scenarios. In: International Conference on Information Systems for Crisis Response and Management (2018)
13. Mannhardt, F., Bovo, R., Oliveira, M.F., Julier, S.: A taxonomy for combining activity recognition and process discovery in industrial environments. In: Yin, H., Camacho, D., Novais, P., Tallón-Ballesteros, A. (eds.) Intelligent Data Engineering and Automated Learning. pp. 84–93. Springer (2018)
14. Mannhardt, F., de Leoni, M., Reijers, H., van der Aalst, W., Toussaint, P.: From low-level events to activities - a pattern-based approach. In: La Rosa, M., Loos, P., Pastor, O. (eds.) Business Process Management. pp. 125–141. Springer (2016)
15. Meroni, G., Di Ciccio, C., Mendling, J.: An artifact-driven approach to monitor business processes through real-world objects. In: Maximilien, M., Vallecillo, A., Wang, J., Oriol, M. (eds.) Service-Oriented Computing. Springer (2017)

16. Mirbabaie, M., Fromm, J.: Reducing the cognitive load of decision-makers in emergency management through augmented reality. In: European Conference on Information Systems (2019)
17. Murphy, R.R.: Trial by fire [rescue robots]. IEEE Robotics & Automation Magazine **11**(3), 50–61 (2004)
18. Murphy, R.R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., Erkmen, A.M.: Search and rescue robotics. Springer handbook of robotics pp. 1151–1173 (2008)
19. Peinel, G., Rose, T., Wollert, A.: The myth of business process modelling for emergency management planning. In: International Conference on Information Systems for Crisis Response and Management (2012)
20. Raso, R., Emrich, A., Burghardt, T., Schlenker, M., Gudehus, T., Sträter, O., Fettke, P., Loos, P.: Activity monitoring using wearable sensors in manual production processes - an application of cps for automated ergonomic assessments. In: Multikonferenz Wirtschaftsinformatik. pp. 231–242. Leuphana Universität Lüneburg (2018)
21. Rebmann, A., Emrich, A., Fettke, P.: Enabling the discovery of manual processes using a multi-modal activity recognition approach. In: Di Francescomarino, C., Dijkman, R., Zdun, U. (eds.) Business Process Management Workshops. pp. 130–141. Springer (2019)
22. Rebmann, A., Rehse, J.R., Pinter, M., Schnaubelt, M., Daun, K., Fettke, P.: IoT-Based Activity Recognition for Process Assistance in Human-Robot Disaster Response (Video) (6 2020). https://doi.org/10.6084/m9.figshare.12409577.v1
23. Schönig, S., Ackermann, L., Jablonski, S., Ermer, A.: Iot meets bpm: a bidirectional communication architecture for iot-aware process execution. Software and Systems Modeling pp. 1–17 (2020)
24. Senderovich, A., Rogge-Solti, A., Gal, A., Mendling, J., Mandelbaum, A.: The ROAD from Sensor Data to Process Instances via Interaction Mining. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) CAiSE 2016. pp. 257–273. Springer (2016)
25. Sora, D., Leotta, F., Mecella, M.: An habit is a process: A bpm-based approach for smart spaces. In: Teniente, E., Weidlich, M. (eds.) Business Process Management Workshops. pp. 298–309. Springer (2018)
26. Tapia, E.M., Intille, S.S., Haskell, W., Larson, K., Wright, J., King, A., Friedman, R.: Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In: International Symposium on Wearable Computers. IEEE (2007)
27. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.P.: Event abstraction for process mining using supervised learning techniques. In: Bi, Y., Kapoor, S., Bhatia, R. (eds.) Intelligent Systems Conference. pp. 251–269. Springer (2018)
28. Tello, G., Gianini, G., Mizouni, R., Damiani, E.: Machine learning-based framework for log-lifting in business process mining applications. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) Business Process Management. pp. 232–249. Springer (2019)
29. Weidinger, J., Schlauderer, S., Overhage, S.: Analyzing the potential of graphical building information for fire emergency responses: Findings from a controlled experiment. In: Internationale Tagung Wirtschaftsinformatik (2019)
30. Willms, C., Houy, C., Rehse, J.R., Fettke, P., Kruijff-Korbayová, I.: Team communication processing and process analytics for supporting robot-assisted emergency response. In: nternational Symposium on Safety, Security, and Rescue Robotics (2019)