No algorithm details. Activity recognition based on known pattern definition

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 96 (2016) 672 – 681

Procedia Computer Science

20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems

Activity detection in smart home environment

Pavle Skocir*, Petar Krivic, Matea Tomeljak, Mario Kusek, Gordan Jezic

University of Zagreb, Faculty of Electrical Engineering and Computing, Internet of Things Laboratory, Unska 3, HR-10000 Zagreb, Croatia

Abstract

Detection of human activities is a set of techniques that can be used in wide range of applications, including smart homes and healthcare. In this paper we focus on activity detection in a smart home environment, more specifically on detecting entrances to a room and exits from a room in a home or office space. This information can be used in applications that control HVAC (heating, ventilation, and air conditioning) and lighting systems, or in Ambient Assisted Living (AAL) applications which monitor the people's wellbeing. In our approach we use data from two simple sensors, passive infrared sensor (PIR) which monitors presence and hall effect sensor which monitors whether the door is opened or closed. This installation is non-intrusive and quite simple because the sensor node to which sensors are connected is battery powered, and no additional work to ensure power supply needs to be performed. Two approaches for activity detection are proposed, first based on a sliding window, and the other based on artificial neural network (ANN). The algorithms are tested on a dataset collected in our laboratory environment.

Peer-review under responsibility of KES International

Keywords: event recognition; activity detection; data filtering; Internet of Things

1. Introduction

Rapidly increasing number of physical devices that are connected to the Internet enables accelerated development of Internet of Things (IoT) applications that can improve our quality of lives[1]. By the end of 2020 it is expected that 20 billion connected devices will be deployed, while in 2016 the number of connected devices in use is 6.4 billion[2]. Applications that use connected devices can be grouped into three main domains[3]: industrial domain, smart cities domain, and health & well-being domain.

Smart homes as a part of smart cities domain are often mentioned in the surveys that focus on IoT[1,3,4]. By connecting devices as thermostats to the Internet, home automation systems enable remote control of HVAC (heating, ventilation, and air conditioning) systems via web or mobile applications. Additionally, within smart grid devices can suggest optimizations of energy consumption by creating a schedule by which home appliances are turned on at the time of more favorable tariffs. Certain applications enable even more advanced capabilities, such as controlling home appliances according to user location. For instance, turning on the heating when user is on her/his way home,

* Pavle Skocir, MSc. Tel.: +385-1-6129-838; fax: +385-1-6129-832.
E-mail address: pavle.skocir@fer.hr

Peer-review under responsibility of KES International
doi:10.1016/j.procs.2016.08.249

or turning on the lights and multimedia system when a user enters the room. Advancements in sensing technologies, embedded processors and communication systems also enhanced integration of independent living services from the health domain within a smart home setting[5]. These services are often referred to as Ambient Assisted Living (AAL) and their main goal is to determine the wellness of elderly people, people with disabilities, or people with acute or chronic pathologies living independently in their home[6].

Wellness of people can be inferred by monitoring activities of daily living (ADL). These activities can be detected by capturing and analyzing time series sensor data originating from various smart sensors of AAL, e.g. motion detectors, heart rate monitors or similar. Activities that could be detected include sleeping, eating, toileting, relaxing, watching TV[5,7], or entering and exiting a home[8]. Detected activities can further be analyzed to detect patterns in behavior by applying machine learning techniques. During regular operation, algorithms detect changes in pattern behaviour which might be an indication of a problem and a trigger for alerting emergency services.

This work focuses on one particular activity within a smart home that can be used for AAL purposes, detection of entrances and exits to and from a room in a home or an office space. By having this information, HVAC or lighting systems can be controlled, or wellness of users can be monitored (e.g. if user stays for too long in a bed in her/his bedroom might imply a problem with user's health). We want to detect these activities by using two simple sensors, passive infrared sensor (PIR) which monitors presence and hall effect sensor which monitors whether the door is opened or closed. The sensors are connected to a battery-powered sensor node, that can be referred to as M2M device[9]. This installation is non-intrusive, it allows residents to stay in their home or office without any intrusion. Furthermore, the installation of the M2M device with sensors is quite simple because the M2M device is battery powered, and no additional work to ensure power supply needs to be performed. This is convenient especially for implementation in homes of elderly people who reluctantly accept even minor construction works in their homes. However, the fact that devices do not have access to unlimited power supply requires taking energy efficiency into account. This particular topic has been in focus of our previous work[9,10,11].

Section 2 describes related work in connection with activity detection. Section 3 presents our two approaches for detecting a particular type of activities, entrances and exits to and from a room. First approach is based on a sliding window, and the other on artificial neural network (ANN). Section 4 evaluates the effectiveness of the proposed approaches, while Section 5 presents concluding remarks.

2. Related work

Activity detection techniques have been widely researched and some of the findings focusing on smart home domain will be presented in this section. Before finding algorithms that could recognize activities in real time, research activities were focused on offline mechanisms which use static data sets, in which all the data is firstly stored and then analyzed. Hong and Nugent[12] focus on segmenting sensor data to extract each segment of consecutive sensor events associated with a complete activity. They detect using the toilet, taking a shower, leaving the house, going to bed and preparing meals. By taking into account correlations of locations, objects and sensors with activities being monitored, they propose three approaches to sensor stream segmentation: location-based approach, model-based approach and dominant centered model-based approach. All three algorithms showed similarly good performances for segmentation and activity classification. However, they point out that the increased prevalence of pervasive technologies such as mobile phones, tablet computers and wireless sensor networks could have an impact on these algorithms since they are all based on mappings between objects and activities, and between locations and activities.

Tao Gu et al.[13] present a way to avoid usual supervised learning phase in the machine learning process for activity recognition. They base their algorithm on object-use fingerprints and test it on various everyday activities such as: making coffee, making phone calls, washing clothes, taking pills, reading books, just to mention a few. The main idea is to retrieve objects used in a specified activity from the Web and identify the relevance weight for each retrieved object. Since activities may share common objects, it is also necessary to mine a set of contrast patterns from object terms and their relevance weights for each activity class. Segmenting data is done using the sliding window combined with "MaxGap" and "MaxGain" segmentation heuristic algorithms to determine the beginning and ending of activity. The result shows that this recognition algorithm achieves precision of 91.4%, which is almost as good as hidden Markov model algorithm which includes a learning phase (93.5%).

Jie Wan et al.[14] implement a way to process sensor data and recognize activities in real-time. Most of the algorithms perform analysis offline, by using stored datasets which are good for researches, but in real-world environment data should be instantly processed so that the proper actions can be taken if needed. The authors concentrate on data segmentation in real time by using sensor and time correlation. Observed activities in this work were also usual daily activities in a smart home environment as listed earlier in this section. Different algorithms were tested for activity recognition, such as Bayesian network, decision trees, and Hidden Markov Models (HMM). It was proved that selection of the algorithm had a great influence on final results. Additionally, it was proved that segmentation has great impact on the capabilities of activity recognition algorithms.

Reducing energy consumption on M2M devices (sensor nodes) is a very important aspect that needs to be considered especially when the devices are battery powered. Special attention towards energy consumption should be paid when deploying software on the devices. Wang et al.[15] present a distributed event detection approach using self-learning threshold. Along with reliable detection, the authors state that energy saving is another major challenge on resource-constraint sensor nodes when designing such system. To tackle the issue with energy consumption, within their work they propose a timer-based node sleep scheduling to prolong network lifetime during the detection process.

In most cases when solutions for activity detection in smart home environment were presented, two level event detection approach was proposed. First level refers to activity detection on a sensor node, and the second level refers to gateway reaching a consensus among individual sensor node decisions about the activity[16]. Also, activity detection made great improvements in the area of elderly people assisted living[17].

In this paper, we focus on recognizing entrances and exits to and from a room. Firstly two different algorithms were proposed, sliding window and artificial neural network and tested on a static offline dataset to analyse their performances for activity recognition. These techniques were chosen since we already have experience with them[18,19], and because they are applicable for activity detection problems[20,21]. The algorithm with better performances showed on an offline dataset was then implemented on a Libelium Waspmote device to recognize the mentioned activities in real time.

3. Activity detection algorithms

Two approaches were observed for detection of entrances and exits within smart home environment. First approach is based on sliding window, while the other approach is based on artificial neural network (ANN). The algorithms were tested offline, on a static dataset collected over a period of five days. The approach based on sliding window was implemented on an M2M device to monitor the desired activities, i.e. entrances and exits to and from a room, in real-time. Results of the detection were compared to activities that actually occurred and that were jotted down by employees in the lab, and along with offline analysis are presented in Section 4.

Recognition of room entrances and exits is performed by using two simple sensors: a passive infrared sensor (PIR) which monitors presence, and hall effect sensor which monitors if the door is opened or closed. PIR sensor activates when a movement is detected within its field of view. Hall effect sensor consists of two parts, one is placed on the door, while the other is placed on the door frame. The sensor produces a different output based on the magnetic field created by those two parts. Both sensors are connected to one M2M device. PIR sensor is directed in a way which enables capturing all the movements within the door frame. Different outputs are produces when the parts are close together or apart.

Presence sensor has two possible values: ”1” when person is present in front of the sensor, and ”0” when no person is present in front of the sensor. The hall effect sensor also has also two possible values: ”3” when the door is closed, and ”0” when the door is opened.

To monitor room entrances and exits, readings of both sensors are necessary. In a single moment two sensors, each with two possible values, can have four different combinations of sensor readings. How those sensor readings can be used to conclude whether an entrance to the room or exit from the room occurred will be presented in the remainder of the section.

3.1. Sliding window algorithm

Two sensors that were used for infer whether an entrance or exit occurred can each have two outputs. By taking this into account, we can define four states that are shown in Figure 1.

Fig. 1. States with sensor values

Furthermore, we define activities that can be detected when monitoring room entrances and exits with these two sensors. After studying all possible cases, we identified six different activities:

1. entrance when the door is closed beforehand, with closing the door afterwards,
2. entrance when the door is closed beforehand, without closing the door afterwards,
3. exit when the door is closed beforehand, with closing the door afterwards,
4. exit when the door is closed beforehand, without closing the door afterwards,
5. entrance/exit when the door is opened beforehand, with closing the door afterwards,
6. entrance/exit when the door is opened beforehand, without closing the door afterwards.

For activities 1-4 it was possible to discern whether the activity was entrance or exit. Unfortunately, for activities 5 and 6 when door is opened beforehand, it was impossible to discern the type of activity.

In order to recognize each activity, we manually pre-segmented sensor data and assigned a sequence of state changes that should occur to define a given activity, as shown in Table 1. We used a sliding window to track last five state changes in a given moment, which is a commonly used approach in human activity detection[20]. The algorithm was implemented to determine the state from read sensor values, and then the state is added at the end of the queue if it is different from the last state in it. Afterwards, if the state is added to the queue, the algorithm checks if the sequence of states within the sliding window matches some of listed activities.

Table 1. Sequences of state changes that define activities

Activity	Sequence of state changes
1	s1→s2→s3→s4→s1
2	s1→s2→s3→s2
3	s1→s4→s3→s2→s1 s1→s4→s3→s1 s1→s4→s3→s4→s1
4	s1→s4→s3→s2
5	s2→s3→s4→s1
6	s2→s3→s2

Figure 2 shows the sequence of states which occur for activity 1. The monitoring of this activity starts with the state when door is closed, and no presence is detected (state s1). First change in sensor readings is when a user opens the door (state s2). Slightly afterwards, as the user enters the room and approaches the M2M device which is located near the door on the inner side of the room, presence sensor detects user's movement. At that time, the door is still opened (state s3) and user is entering the room while closing the door. Afterwards, user closes the door and heads to a certain destination within the room (state s4). When user walks away from the door and M2M device, the M2M device reports the reading as in the beginning (state s1).

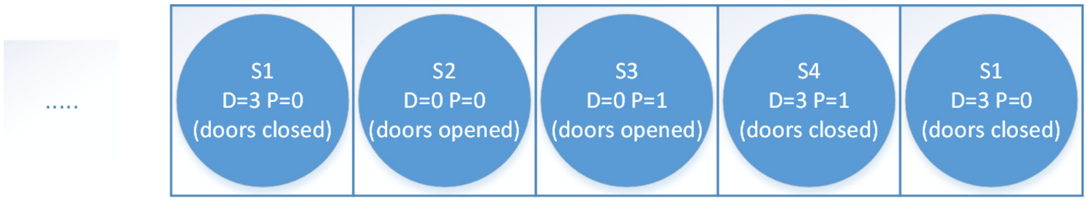

Fig. 2. Example of states in the sliding window for activity 1 - entrance when the door is closed beforehand with closing the door afterwards

As for activity 1, state sequences are defined for other five activities. Activity 3 has three possible sequences, which were detected by monitoring sensor readings while entering and exiting the room. For activities 5 and 6 it is impossible to discern whether the activity is an entrance or exit because both in cases of entrance and exit first sensor to change state is presence (change from state s2 to s3). When entering the room, a user passes the door and firstly comes into the field of view of the presence sensor. After that, the user can close the door (state s4) and walk away from the presence sensor. The same state sequence can be detected when the user exits the room. Firstly the user approaches the door and comes into the field of view of presence sensor (state s3), exits the room and closes the door while presence sensor is still active because of all the movements (state s4), and in the end the presence sensor is deactivated (state s1). For activity 6, when door is opened before and after the activity, only presence sensor participates in activity detection and it cannot be distinguished whether the user activated the sensor when entering or leaving the door. To be able to do that, another presence sensor would be needed.

Algorithm 1 shows the algorithm for detecting the desired activities. Firstly the sensor values are read from the database. Based on their values, the state is inferred. If the current state is different from the last one, then this state is added to the queue. Sliding window monitors the last five entries within the queue. The states in the sliding window are compared to the sequence of states shown in Table 1. If the sequence in the sliding window matches one of the sequences from the table, the code of the activity is generated as the output.

Algorithm 1 Sliding window algorithm

```
procedure FIND ACTIVITY
loop:
    presenceValue ←value of presence sensor
    doorValue ←value of hall effect sensor
    State ←(presenceValue, doorValue)
    if State! = lastStateInQueue then
        Queue ←State
        if queueSequence = activityPattern then
            print Activity
    goto loop.
```

3.2. Algorithm based on Artificial Neural Network

Data from sensors introduced at the beginning of this section was also used by artificial neural network (ANN) to determine whether a person entered or exited the room. Artificial Neural Network (ANN) is a branch of machine learning based on replicated biological neural network that can also be applied to human activity detection[21]. It consists of nodes (neurons) and connections (synapses) between them. Through connections neurons are sending each other signals, and each bond is determined by its weight. During learning phase, ANN adapts connection weights according to learning samples.

In our case, each sensor data is specific by two important values: time and measurement value. Each measurement value is actually a combination of values from two sensors (presence sensor and hall effect sensor). Additionally, it is not enough to know only one separate measurement, but it is necessary to monitor changes of sensor outputs in time. For our purposes we selected a Time delay network, which does not require development of specific algorithms for learning. It is necessary to adjust the layout of the network to a specific form of data that is being analyzed. The input layer of the network was expanded to accept more than one sample. The disadvantage of this ANN type is that it has a limited number of stored samples. As in the example with the sliding window, we monitor last five readings (the value just obtained from sensors and the last four historical values), which should be enough to detect the wanted activities listed in Section 3.1. Network architecture created for this project is shown in Figure 3.

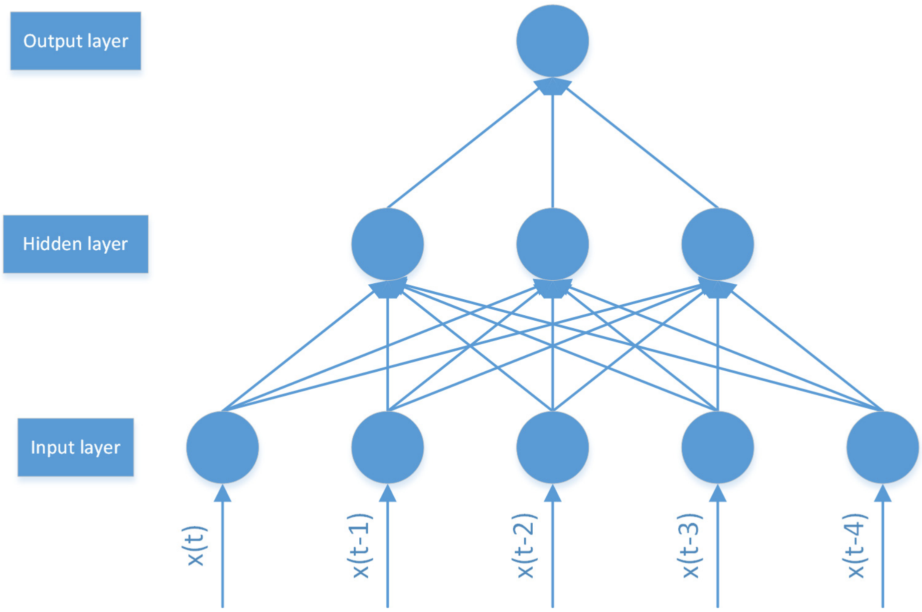

Fig. 3. The neural network with time delay

According to the availability of input and output data, learning in ANN is divided into supervised, unsupervised and reinforcement learning. Within this implementation we consider supervised learning that assumes that the set of input and expected output data is known. For network learning it is necessary to go through a training phase in which weights are being assigned to connections between neurons. Learning algorithms usually assign pseudo-random numbers at the beginning of the process to connections. The weights are changing depending on the validity of results as long as the error can be tolerated. At the end of training phase network is suitable for activity recognition even on data different from the training set.

Input values used in this approach both for training and recognition phase are shown in Table 2, while output values are shown in Table 3. States s1-s4 from the sliding window approach are now mapped to values between 0 and 1. Codes of detected activities (1-6) in sliding window approach are in this approach values between 0 and 1, as for input. The connotation is the same, just the appearance is different. Input values still stand for the measurement values of two sensors used for detection, while output values represent the type of the detected activity. In the learning phase of the algorithm, sequence of events as described in Table 1 is brought to the inputs layer of the neural network. The outputs are written in the form presented in Table 3. The input sample for activity 1 now looks like this: $x(t) = 0$, $x(t-1) = 1$, $x(t-2) = 2/3$, $x(t-3) = 1/3$, $x(t-4) = 0$, while the value of the output sample is 1.

Table 2. ANN input values

Value	Sensor states
0	the door is closed, no person is present in front of the sensor
$\frac{1}{3}$	the door is opened, no person is present in front of the sensor
$\frac{2}{3}$	the door is opened, person is present in front of the sensor
1	the door is closed, person is present in front of the sensor

Table 3. ANN output values that define the activity

Value	Activity
$0 - \frac{1}{7}$	no activity recognized
$\frac{1}{7} - \frac{2}{7}$	entrance/exit when the door is opened beforehand, without closing the door afterwards
$\frac{2}{7} - \frac{3}{7}$	exit when the door is closed beforehand, without closing the door afterwards
$\frac{3}{7} - \frac{4}{7}$	exit when the door is closed beforehand, with closing the door afterwards
$\frac{4}{7} - \frac{5}{7}$	entrance/exit when the door is opened beforehand, with closing the door afterwards
$\frac{5}{7} - \frac{6}{7}$	entrance when the door is closed beforehand, without closing the door afterwards
$\frac{6}{7} - 1$	entrance when the door is closed beforehand, with closing the door afterwards

Algorithm 2 ANN algorithm

```
procedure FIND ACTIVITY
Network ←create Neural Network
    training(Network)
    fileReading ←read presence and door values from file
    loop:
    Line ←one line of fileReading contains presence and door values with same timestamp
    if Line! = null then
        State ←stateSetter(presenceValue[Line], doorValue[Line])
        Queue ←State
        Activity ←activityRecognizer(Queue,Network)
        if Activity! = no activities then
            print Activity
    goto loop.
```

For our implementation we used the program framework Encog (http://www.heatonresearch.com/encog/) which has built-in classes and functions to simulate ANN. A neural network was created by defining layers and nodes. Training phase was executed by using the input samples as described in Table 1, but with input and output values as presented in Table 2 and Table 3.

Algorithm 2 shows the main steps of the neural network algorithm. Firstly, neural network is created based on the defined parameters (i.e. number of layers and nodes). Afterwards, training set is executed. Then a loop starts in which sensor data from text file is brought to the input layer of ANN. *Statesetter* procedure is reading the sensor values and converting them to a format as shown in Table 2. Then the neural network produces the output, after which *activityRecognizer* procedure is analyzing the produced output. For instance, value 0.9 would signify that the activity belongs to the activity entrance when the door is closed beforehand, with closing the door afterwards because it is in the interval that belongs to that particular activity (6/7 - 1). Basically, the *activityRecognizer* procedure tries to find one of the intervals shown in Table 3 to which the produced output value belongs.

4. Evaluation

This section presents the evaluation of proposed algorithms. Firstly, algorithms were tested offline, on a static data set. Afterwards, one algorithm was chosen for implementation on an M2M device what enabled online activity detection, in real-time.

4.1. Comparison of the two proposed algorithms

Sliding window algorithm and the neural network algorithm were both tested offline, on a static data set. All activities described in Section 3.1 were registered manually by the employees in the laboratory, while readings from hall effect and PIR sensor connected to Libelium Waspmote v1.2 device were obtained every second, sent to server and stored in a database. Hall effect sensor monitored whether the door was opened or closed while PIR sensor was directed so that each movement within the door frame could have been captured. The readings were retrieved from database and then analysed by the two algorithms. The output of algorithms was the code of recognized event. Table 4 shows the total number of occurred activities, well detected activities, not detected activities and wrongly detected activities during a period of three days.

The success of predicting the activities is evaluated by using precision and recall metric. Precision is the number of true positives (T_p), i.e. the number of correctly detected activities over the number of true positives plus the number of false positives (F_p), i.e. detected activities which did not occur:

$P = \frac{T_p}{number\ of\ detected\ activities} = \frac{T_p}{T_p+F_p}$.

Recall is defined as the number of true positives over the number of true positives plus the number of false negatives (F_n), i.e. the activities which did occur, but were not detected.

$R = \frac{T_p}{number\ of\ ocurred\ activities} = \frac{T_p}{T_p+F_n}$.

Table 4. Occurred and detected activities - offline analysis

	Sliding window	**Neural network**
Occurred activities	162	162
Well detected activities (true positives, T_p)	98	62
Not detected activities (false negatives, F_n)	64	100
Wrongly detected activities (false positives, F_p)	64	142
precision, $T_p/(T_p + F_p)$	0.605	0.304
recall, $T_p/(T_p + F_n)$	0.605	0.383

As it can be seen in Table 4, precision and recall values are lower for neural network algorithm. Furthermore, implementation of the neural network algorithm is more complex, there has to be a training phase before putting the algorithm into execution. On the other side, sliding window algorithm can be implemented more easily and it showed better results.

Since in our case the sliding window algorithm has shown better performance and is more appropriate for a real-time usage (no training phase needed), we decided to use it in our online data analysis in real-time. The algorithm is similar to the one tested on a static dataset and shown in Algorithm 1. Algorithm code is implemented and uploaded on Libelium Waspmote v1.2 device with connected PIR and hall effect sensors. Every second sensor values are read and if an activity occurs, its code is sent to the back-end system. In a real-world implementation this is a good progress since now M2M devices do not have to send each sensed value towards gateway and back-end system, but only information about activity when it occurs, which is a great energy consumption improvement.

4.2. Evaluation in real-time

As in the Section 4.1, all activities described in Section 3.1 were registered manually by the employees in the laboratory, but in this experiment they were predicted in real-time by the algorithm implemented on the M2M device. The M2M device and sensors were situated as in the setting when analysis was performed offline. But in this experiment, only code of the recognized event was sent to back-end system. The analysis about how effective the algorithm detects the events was then performed.

The number of monitored activities collected during period of five days is shown in Table 5. Each column represents one type of activity, as presented in Section 3.1. The success of predicting the aforementioned six activities is evaluated by using precision and recall metric, as in Section 4.1. As it can be seen from Table 5, during regular operation in our laboratory for five working days, altogether 236 activities occurred, out of which 203 were correctly detected (true positives). Along with that, another 64 activities were detected which did not occur (false positives). More than 50 occurrences happened for each of the entrance and exit activities when the door was left in the same state like they were found (closed for activities 1 and 3 and opened for activity 6).

Table 5. Occurred and detected activities - online analysis

	1	2	3	4	5	6	all activities
Occurred activities	53	33	56	3	33	58	236
Well detected activities (true positives, T_p)	49	30	48	3	18	55	203
Not detected activities (false negatives, F_n)	4	3	8	0	15	3	33
Wrongly detected activities (false positives, F_p)	0	1	2	6	5	50	64
precision, $T_p/(T_p + F_p)$	1	0.968	0.96	0.333	0.783	0.524	0.76
recall, $T_p/(T_p + F_n)$	0.925	0.909	0.857	1	0.545	0.948	0.86

For activity 1, when someone entered the room, finding and leaving the door closed, 92.5% of the activities was detected. These 7.5% of activities were not detected in cases when a user stopped moving for a brief moment before closing the door. In that particular case, the presence sensor would deactivate and state sequence would be different than defined in the algorithm. Precision for this activity was 1, which means that there were no false positives during detection process. Recall for activity 2 (90.9%) was similar to the recall for activity 1. Precision for activity 2 is little smaller, i.e. there was one false positive activity detected. For activity 3 the precision is similar as for activity 2. False positives were detected for cases when a person stands by the door and speaks to someone in the room while holding the doorknob and opening and closing the door. The other case is when someone peeps into the room, i.e. enters and afterwards quickly leaves the room. False negatives occurred when a person exits a bit slower, or talks at someone in the room during the exit. In such cases activities 4 and 5 are often wrongly detected instead of activity 3.

Activity 4 occurred only a small number of times, but recall was 1, which means that all such activities were detected, there was no false negatives. However, precision is the lowest for this activity. The reason for that is slow exit from the room (when this activity is detected instead of activity 3), often accompanied by talking to someone in the room. Recall for activity 5 is the lowest, only 54% of these activities were correctly detected. The reason for that is a bit slower exit, when presence sensor is deactivated sooner than the door is closed. Precision is at 78%, false positives were often detected when door was closed from the inside, without anyone exiting. Recall for the activity 6 was among the highest. However, there was a lot of false positives which caused lower precision. These false activities were detected every time the presence sensor was activated. That happened when someone was walking in the room near the sensor, and there was a lot of such movements. The absolute number of false positives for activity 6 is the highest among all the activities because of that. The reason for false negatives is entrance of multiple persons at the same time (one after another). Also, the occasional malfunction of the presence sensor, i.e. when it was not activated when it should have been, caused false negatives.

5. Conclusion

This paper presented two approaches for activity detection in smart home environment. The activities being monitored, entrances to a room and exits from a room, can be used in applications which monitor the wellbeing of people (Ambient Assisted Living applications, AAL), in HVAC (heating, ventilation and air conditioning) applications for monitoring and regulating the temperature of the room, for regulation of lighting systems, etc. The algorithms were tested offline, on static data set, and online in real-time. In offline analysis, precision and recall for sliding window algorithm were 60%, while for neural network precision amounted to 30% and recall to 38%. Precision and recall values for neural network algorithm could be enhanced if a larger data set was used for training. However, since sliding window algorithm is easier for implementation and showed better performance, it was chosen to be implemented on M2M device for online analysis. In online analysis, the precision was 76%, and recall was 86%. The reason why online analysis showed better results than offline analysis lies in the fact that some of the data packets sent from M2M device to gateway, with data later used for offline analysis, did not reach their destination.

In future work, we plan to update the algorithm which runs on M2M device. In a current version, sensor readings are obtained every second. That is not really necessary, and sensor readings could be triggered by an event, i.e.

readings could be obtained after a sensor changes its output, e.g. when someone opens the door. Sensor triggering is enabled on Libelium Waspmote v1.2 devices. By applying this approach the device could spend more time in low-energy mode and conserve energy when no events occur. Additionally, we plan to enhance the mechanism for detecting entrances and exits so that it could predict the number of people in the room at a certain moment, which could also be applied in AAL applications.

Acknowledgements

This work has been supported by Croatian Science Foundation under the projects 8065 (Human-centric Communications in Smart Networks) and 8813 (Managing Trust and Coordinating Interactions in Smart Networks of People, Machines and Organizations).

References

1. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M.. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials* 2015;**17**(4):2347–2376. doi:10.1109/COMST.2015.2444095.
2. Gartner, . Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015. Press Release; 2015. URL: `http://www.gartner.com/newsroom/id/3165317`.
3. Borgia, E.. The internet of things vision: Key features, applications and open issues. *Computer Communications* 2014;**54**:1 – 31. doi:http://dx.doi.org/10.1016/j.comcom.2014.09.008.
4. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems* 2013;**29**(7):1645 – 1660. doi:http://dx.doi.org/10.1016/j.future.2013.01.010; including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services & Cloud Computing and Scientific Applications — Big Data, Scalable Analytics, and Beyond.
5. Suryadevara, N.K., Mukhopadhyay, S.C.. *Smart Homes: Design, Implementation and Issues*. Springer; 2015. ISBN 978-3-319-13556-4.
6. Grguric, A., Dešić, S., Mošmondor, M., Kušek, M.. Towards medication management in smart homes. In: *Proceedings of the International Conference on Health Informatics (BIOSTEC 2013)*. 2013, p. 228–232. doi:10.5220/0004238002280232.
7. Soulas, J., Lenca, P., Thépaut, A.. Unsupervised discovery of activities of daily living characterized by their periodicity and variability. *Engineering Applications of Artificial Intelligence* 2015;**45**:90 – 102. doi:http://dx.doi.org/10.1016/j.engappai.2015.06.006.
8. Jakkula, V., Cook, D.J.. Detecting anomalous sensor events in smart home data for enhancing the living experience. In: *Proceedings of the 7th AAAI Conference on Artificial Intelligence and Smarter Living: The Conquest of Complexity*; AAAIWS'11-07. AAAI Press; 2011, p. 33–37.
9. Skocir, P., Kusek, M., Jezic, G.. Influence of on-device measurement analysis on energy efficiency in machine-to-machine systems. In: *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*. 2015, p. 1273–1278. doi:10.15439/2015F306.
10. Skocir, P., Zrncic, S., Katusic, D., Kusek, M., Jezic, G.. Energy consumption model for devices in machine-to-machine system. In: *Telecommunications (ConTEL), 2015 13th International Conference on*. 2015, p. 1–8. doi:10.1109/ConTEL.2015.7231224.
11. Skocir, P., Maracic, H., Kusek, M., Jezic, G.. *Agent and Multi-Agent Systems: Technologies and Applications: 9th KES International Conference, KES-AMSTA 2015 Sorrento, Italy, June 2015, Proceedings*; chap. Data Filtering in Context-Aware Multi-agent System for Machine-to-Machine Communication. Cham: Springer International Publishing; 2015, p. 41–51. doi:10.1007/978-3-319-19728-9_4.
12. Hong, X., Nugent, C.D.. Segmenting sensor data for activity monitoring in smart environments. *Personal Ubiquitous Comput* 2013; **17**(3):545–559. doi:10.1007/s00779-012-0507-4.
13. Gu, T., Chen, S., Tao, X., Lu, J.. An unsupervised approach to activity recognition and segmentation based on object-use fingerprints. *Data Knowl Eng* 2010;**69**(6):533–544. doi:10.1016/j.datak.2010.01.004.
14. Wan, J., O'Grady, M.J., O'Hare, G.M.P.. Dynamic sensor event segmentation for real-time activity recognition in a smart home context. *Personal and Ubiquitous Computing* 2014;**19**(2):287–301. doi:10.1007/s00779-014-0824-x.
15. Wang, Y., Wang, D., Chen, F., Fang, W.. Efficient event detection using self-learning threshold for wireless sensor networks. *Wirel Netw* 2015;**21**(6):1783–1799. doi:10.1007/s11276-014-0885-9.
16. Mekikis, P.V., Athanasiou, G., Fischione, C.. A wireless sensor network testbed for event detection in smart homes. In: *2013 IEEE International Conference on Distributed Computing in Sensor Systems*. 2013, p. 321–322. doi:10.1109/DCOSS.2013.36.
17. Benmansour, A., Bouchachia, A., Feham, M.. Human activity recognition in pervasive single resident smart homes: State of art. In: *Programming and Systems (ISPS), 2015 12th International Symposium on*. 2015, p. 1–9. doi:10.1109/ISPS.2015.7244997.
18. Antonić, A., Pripužić, K., Marjanović, M., Skočir, P., Ježić, G., Žarko, I.P.. A high throughput processing engine for taxi-generated data streams. In: *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems*; DEBS '15. New York, NY, USA: ACM; 2015, p. 309–315. doi:10.1145/2675743.2772588.
19. Vukovic, M., Skocir, P., Katusic, D., Jevtic, D., Trutin, D., Delonga, L.. Estimating real world privacy risk scenarios. In: *Telecommunications (ConTEL), 2015 13th International Conference on*. 2015, p. 1–7. doi:10.1109/ConTEL.2015.7231214.
20. Ortiz Laguna, J., Olaya, A.G., Borrajo, D.. *User Modeling, Adaption and Personalization: 19th International Conference, UMAP 2011, Girona, Spain, July 11-15, 2011. Proceedings*; chap. A Dynamic Sliding Window Approach for Activity Recognition. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011, p. 219–230. doi:10.1007/978-3-642-22362-4_19.
21. Zheng, H., Wang, H., Black, N.. Human activity detection in smart home environment with self-adaptive neural networks. In: *Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on*. 2008, p. 1505–1510. doi:10.1109/ICNSC.2008.4525459.