ISSN:0975-9646

H. Karamath Ali et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 2453-2457

Detecting Abnormality in Activities Performed by People with Dementia in a Smart Environment

H. Karamath Ali[1#], Dr. D. I. George Amalarethinam[2#]

[1] *Research Scholar in Computer Science,* [2] *Director-MCA & Associate Professor,*
[#] *Jamal Mohamed College(Autonomous), Trichirappalli, TamilNadu, India*

Abstract – Recognition of abnormalities in the activities performed by users in smart environments has applications in many such fields as surveillance and medical care for differently abled and elderly persons. There are many causes for abnormality in user activities. Missing out some events of an activity, not maintaining the proper order of events and occurrence of irrelevant events are the major causes for abnormality in activities performed by persons with early stages of dementia. In this paper a simple and novel method for identifying missed out events, out of order events and irrelevant events is proposed. The method can be used for measuring how well an activity is performed by the user and for offering corrective suggestions. One of the possible applications of this method is enabling elderly people to lead an independent life. The proposed method was tested using a publicly available test data after introducing abnormalities manually, and is found to produce promising results.

Keywords – Pervasive Computing, Activity Recognition, Abnormal Activities, Aging, Dementia, Smart Environments.

I. Introduction

The main aim of activity recognition in pervasive environments is to ascertain the activity being pursued by the user in order to take proactive actions to help the user. The help provided to the user may be for effectively completing the activity. This kind of help is necessary for users with cognitive difficulties either due to old age or other disabilities. For this the activity recognition system, besides being able to identify the activity performed by the user, needs to be able to decide if there is any abnormality in the activity. There are different causes for abnormality in user activities. Missing out one or more constituent steps of an activity is one of the causes of abnormality[1][2]. The constituent steps of an activity may be called events for the sake of clarity. Users perform different sequences of events to accomplish the same activity at different times. Nevertheless there may be some events without which an activity may be incomplete. Such events that are indispensable to an activity may be called core events. Events other than the core ones of an activity may be called sub events. For example, 'opening shower' is a core event of the activity 'bathing', whereas 'using shampoo' is a sub event of the activity. Missing out a core event results in abnormality whereas a missed out sub event does not. Obviously, bathing without opening the shower is an abnormality whereas not using shampoo is not so.

Ordering of events also is important for successful completion of an activity. Not following the order of events is another reason for abnormality[2]. Depending upon the activity of interest, ordering may involve both core and sub events. For example, in the activity 'TV watching', 'switch on TV' is a core event; 'surfing channels using remote control' is a sub event. But the ordering between these two events is important because without switching the TV on, it is not possible to surf channels using the remote control. Also it is obvious that every occurrence of the activity 'TV watching' may not have the event 'surfing channels'.

Earlier methods used for detecting abnormality require extensive computations. In this paper a simple and novel method for representing the structure of normal activities from sensor data i.e. activities without any abnormalities, is proposed. The method is inspired by the simplicity of the work of Xin Hong, et al.[3] which proposed a method of identifying activities using two similarity measures to compare a sensor event sequence and a generic sensor model of an activity. While the order of events is not considered in their work, the method proposed in this paper suggests a simple and novel method of detecting core and sub events of an activity and the different pairs of events between which ordering is important. For dealing with activities which do not contain any core events a simple similarity measure that can be used to decide how well an activity is performed, is also proposed. The proposed method is based on the observation that users perform different sequences of events to accomplish the same activity at different times. This method makes it easier to detect abnormalities in performed activities due to missing out of events or changing order of events. The proposed method can be combined with that of Xin Hong, et al.[3] so that abnormal activities due to the three reasons can be identified even from event sequences without any annotation. In this work it is assumed that problems such as noise in sensor data are handled by appropriate methods, and reliable stream of annotated sensor data is available.

The remainder of this paper is organized as follows. Section II describes the related work. The proposed method is illustrated in section III. Section IV provides details of data and experiment and section V provides conclusion.

II. RELATED WORK

Paul Cuddihy, et al.[4] designed an algorithm named Automatic Inactivity Detection (AID) algorithm that was based on inactivity duration data collected from a smart home over several weeks. The algorithm creates a threshold for acceptable elapsed inactive time for each predefined duration of a day, using the elapsed inactivity data. If the resident is inactive for longer than the threshold duration, an alert is sent to the caregiver.

V. Jakkula and D.J. Cook[5] identified the frequently occurring sequential patterns in the raw sensor data using the Apriori algorithm. Then the occurrences of the temporal patterns defined by Allen, in the output of the algorithm were identified. This information was used to calculate the probability for the occurrence of a given event. This formed the basis of the anomaly detection. The method was tested on the dataset collected from MavLab.

Jie Yin, et al.[6] used a one-class support vector machine (SVM) to filter out the activities with very high probability of being normal. Then abnormal activity models were derived from a general normal model using a kernel nonlinear regression (KNLR) to reduce false positive rate in an unsupervised manner. According to them this approach provided a good tradeoff between abnormality detection rate and false alarm rate.

To assist Alzheimer patients to carry out some activities in a smart home, Patrice C. Roy, et al.[2] presented a formal activity recognition framework based on possibility theory and description logics. The activity recognition process was separated into three agents: the environment representation agent, the action recognition agent, and the behavior recognition agent. The environment recognition agent inferred the current probable context of the environment; the action recognition agent inferred the most probable low-level action that was carried out by the incumbent of the home; The behavior recognition agent inferred whether the user was performing the intended activity in an abnormal manner.

A Hierarchical Dirichlet Process Hidden Markov Model (HDP-HMM), Derek Hao Hua, et al.[7] to automatically find an appropriate number of states for recognizing an activity. Activities that were likely to be normal were filtered out by incorporating a Fisher Kernel into One-Class Support Vector Machine (OCSVM) model. Finally, an abnormal activity model was derived from the normal activity models to reduce false positive rate in an unsupervised manner.

D.J. Cook and M. Schmitter-Edgecombe[1] designed algorithms to automatically learn Markov models for each class of activity. These models were used to recognize activities that were performed in a smart home and to identify errors and inconsistencies in the performed activity.

Irina Mocanu and Adina Magda Florea[8] performed emergencies detection in a smart home using a stochastic contextfree grammar with attributes and a domain activity ontology for modeling the daily programme of the supervised person.

Vikramaditya Jakkula and Diane J. Cook[9] used OCSVM to detect anomalous events in a smart home environment.

All these methods involve relatively expensive computations to detect abnormalities. In this paper a simple and novel method of detecting abnormalities, that can be used for generating prompting messages for such people as those with dementia, from annotated data is presented. When combined with some method for segmenting un-annotated data such as that in [3], the requirement for annotated data for the proposed method can be done away with.

III. THE METHOD

Consider the activity 'watching TV'. The list of possible events that may occur in this activity in a common household may be as follows:

$\boldsymbol{e_1}$: Switch on light in the room

$\boldsymbol{e_2}$: Switch on fan in the room

$\boldsymbol{e_3}$: Get some snacks from the fridge
(to nibble during watching TV)

$\boldsymbol{e_4}$: Switch on TV

$\boldsymbol{e_5}$: Sit on the sofa

$\boldsymbol{e_6}$: Change channel using TV remote

$\boldsymbol{e_7}$: Watch for some time

$\boldsymbol{e_8}$: Switch off TV

$\boldsymbol{e_9}$: Switch off fan

$\boldsymbol{e_{10}}$: Switch off light

Obviously, every event in the above list need not occur each time the user watches TV. For example the user may not like to eat snacks during watching TV or he may not need to switch on the light. So the performance of the activity 'watching TV' at different times by the user may result in different sequences of events in the above list. Some of the possible valid sequences of events at different times of TV watching may be as given below.

$\boldsymbol{e_3\ e_2\ e_1\ e_4\ e_5\ e_6\ e_7\ e_8\ e_9\ e_{10}}$

$\boldsymbol{e_4\ e_2\ e_5\ e_7\ e_9\ e_8}$

$\boldsymbol{e_4\ e_5\ e_6\ e_3\ e_7\ e_8}$

$\boldsymbol{e_3\ e_2\ e_4\ e_9\ e_6\ e_5\ e_2\ e_9\ e_7\ e_8}$

$\boldsymbol{e_3\ e_4\ e_1\ e_5\ e_6\ e_7\ e_8\ e_{10}}$

$\boldsymbol{e_4\ e_5\ e_2\ e_1\ e_5\ e_7\ e_{10}\ e_9\ e_8}$

$\boldsymbol{e_1\ e_{10}\ e_2\ e_4\ e_3\ e_6\ e_5\ e_6\ e_7\ e_9\ e_8}$

To analyze and derive useful information about the structure of the activity, the 'happens after' ordering of the events are entered in a table $\boldsymbol{m}$ which contains one row and one column for each of the possible events that may occur in the activity. If an event '$\boldsymbol{e_i}$' occurs in a sequence, then $\boldsymbol{m}[\boldsymbol{i},\boldsymbol{i}]$ is incremented by 1. So, the number of sequences in which event '$\boldsymbol{e_i}$' occurs, is recorded in $\boldsymbol{m}[\boldsymbol{i},\boldsymbol{i}]$. If event '$\boldsymbol{e_i}$' happens after event '$\boldsymbol{e_j}$' in a sequence then $\boldsymbol{m}[\boldsymbol{i},\boldsymbol{j}]$ is incremented by 1. In general, for each $\boldsymbol{e_i}$ and every $\boldsymbol{e_j}$ that happens after $\boldsymbol{e_i}$ in a sequence, $\boldsymbol{m}[\boldsymbol{i},\boldsymbol{j}]$ is incremented by 1, where $\mathbf{1} \le \boldsymbol{i},\boldsymbol{j} \le \boldsymbol{n}$, $\boldsymbol{i} \neq \boldsymbol{j}$ and $\boldsymbol{n}$ is the number of possible events in the activity. The table created in this way for the above sequences is shown in table 1. For example, $\boldsymbol{e_1}$ occurs in four of the sequences. Hence, $\boldsymbol{m}[\mathbf{1},\mathbf{1}]$ = 4 in the table. Since $\boldsymbol{e_4}$ is followed by $\boldsymbol{e_5}$ in all the seven sequences, $\boldsymbol{m}[\mathbf{4},\mathbf{5}]$ = 7.

TABLE 1. EVENT ORDERING COUNTS

events	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
1	4	2	2	2	3	3	4	4	3	4
2	1	5	2	3	4	3	5	5	5	2
3	1	1	5	3	4	4	5	5	3	2
4	2	2	2	7	7	5	7	7	4	3
5	1	1	1	-	7	3	7	7	3	3
6	-	-	1	-	2	5	5	5	1	2
7	-	-	-	-	-	-	7	7	3	3
8	-	-	-	-	-	-	-	7	-	2
9	-	-	-	1	2	2	2	5	5	1
10	-	1	1	1	1	1	1	2	2	4

From table 1 the following observations can be made.

i) If $\boldsymbol{m}[\boldsymbol{i},\boldsymbol{i}]$ = $\boldsymbol{n}$(the number of event sequences given), it means $\boldsymbol{e_i}$ occurs in every event sequence. That is $\boldsymbol{e_i}$ is a core event. Otherwise $\boldsymbol{e_i}$ is a sub event. So in this example, events $\boldsymbol{e_4}$, $\boldsymbol{e_5}$, $\boldsymbol{e_7}$ and $\boldsymbol{e_8}$ are core events. The remaining are sub-events.

ii) Following possibilities are there as far as the 'happens after' ordering of two events $\boldsymbol{e_i}$ and $\boldsymbol{e_j}$ is concerned.

a) If $\boldsymbol{m}[\boldsymbol{i},\boldsymbol{i}] \ge \boldsymbol{m}[\boldsymbol{i},\boldsymbol{j}] = \boldsymbol{m}[\boldsymbol{j},\boldsymbol{j}]$ and $\boldsymbol{m}[\boldsymbol{j},\boldsymbol{i}] = \mathbf{0}$, then $\boldsymbol{e_j}$ must happen only after $\boldsymbol{e_i}$. That is, if $\boldsymbol{e_j}$ happens after $\boldsymbol{e_i}$ whenever it appears in a sequence, then it can be deciced that $\boldsymbol{e_j}$ must always happen after $\boldsymbol{e_i}$ in the activity. For example in the table, it can be seen that $\boldsymbol{e_7}$ can happen only after $\boldsymbol{e_4}$.

b) Otherwise, ordering of events $\boldsymbol{i}$ and $\boldsymbol{j}$ is immaterial.

As per the above table and the observations, $\boldsymbol{e_{10}}$ must happen only after $\boldsymbol{e_1}$. $\boldsymbol{e_9}$ can happen only after $\boldsymbol{e_2}$. Similarly the prerequisite for the happening of $\boldsymbol{e_5}$ and $\boldsymbol{e_6}$ is the happening of $\boldsymbol{e_4}$. $\boldsymbol{e_7}$ can happen only after $\boldsymbol{e_4}$ and $\boldsymbol{e_5}$. $\boldsymbol{e_4}$, $\boldsymbol{e_5}$ and $\boldsymbol{e_7}$ must have happened before $\boldsymbol{e_8}$. The validity of these observations is self-explanatory from the description of the events.

Table 2 summarises the 'happens after' relationship between the events, extracted from table 1, according to the above observations. 'T' which stands for 'true', in a cell in table 2 indicates that the event corresponding to the column must happen after the event corresponding to the row.

TABLE 2.
SUMMARY OF 'HAPPENS AFTER' RELATIONSHIPS

	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
1	-	-	-	-	-	-	-	-	-	T
2	-	-	-	-	-	-	-	-	T	-
3	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	T	T	T	T	-	-
5	-	-	-	-	-	-	T	T	-	-
6	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	T	-	-
8	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-

To decide if an activity performed by the user has any abnormality, whenever a new event occurs, the following need to be checked.

1. whether the event is irrelevant to the currently pursued activity
2. whether all events that need to precede the event have occurred earlier
3. whether every core event has occurred

By maintaining a events-already-occurred list for the activity, the above checkings can be done easily. Suppose the latest occurred event is $\boldsymbol{e_j}$. If $\boldsymbol{e_j}$ is not in the list of events that may occur in the current activity, then obviously it is an irrelevant event. So, appropriate prompting message may be generated. Otherwise, the $\boldsymbol{j^{th}}$ column of the relationships table for the activity is to be searched for 'T' entries. If an event $\boldsymbol{e_i}$, for which there is a 'T' in the column, is not in the events-already-occurred list, an 'event $\boldsymbol{e_i}$ is not done' message may be generated. To check if every core event has been performed, the system may wait for the maximum duration of the activity which can be collected from the temporally ordered test event sequences. Then from the events-already-occurred list, the missed out core events can be decided and appropriate corrective message can be generated.

Some activities may not have any core events. Deciding how well such activities are performed can be done by calculating a value which will be proportional to how similar the activity performed and the intended activity are. Such a value can be called similarity measure. Similarity measure is defined as the sum of the total weights of the performed events. Weight of an event $\boldsymbol{e_i}$, denoted by $\boldsymbol{w(e_i)}$, is calculated as

$$\boldsymbol{w(e_i) = \frac{n_{e_i}}{\sum_{j=1}^{n} n_{e_j}} \times \frac{n_{e_i}}{n}}$$

where

$\boldsymbol{n_{e_i}} -$ the number of occurrences of $\boldsymbol{e_i}$ in the training event sequences

and

$\boldsymbol{n} -$ the number of training event sequences.

The above formula is based on the following reasons:

- weight of an event must be proportional to the number of times the event occurs relative to other events, in the training event sequences
- weight of an event must be proportional to the number of training event sequences in which the event occurs

The similarity measure($\boldsymbol{sm}$) of the performed activity is then calculated as

$$\boldsymbol{sm = \sum_{k=1}^{m} w(e^k) \times e_t^k}$$

where $\boldsymbol{e^k} \in \{\boldsymbol{e_1, e_2, \ldots, e_n}\}$, $\boldsymbol{e_t^k}$ is the number of occurrences $\boldsymbol{e^k}$ in the given event sequence and $\boldsymbol{m}$ is the number of different events in the sequence.

The method of calculating $\boldsymbol{sm}$ is based on the fact that the more number of times an event of an activity occurs, the more the given sequence resembles the activity. If the value of $\boldsymbol{sm}$ is more than a given threshold value, then it can be decided that the given sequence is adequately similar to the activity in question. Also, the 'happens after' relations calculated as above can be used to decide if the events have occurred in proper order.

IV. Data and Experiment

To test the proposed method, the temporally ordered data set collected and made public by Tim van Kasteren, et al.[10] was used. The data set consists of 28 days of data collected using 14 different sensors on seven different activities of a person in a smart home environment. The sensors generated binary outputs whenever the corresponding objects were handled by the subject. The outputs were recorded in temporally sequential order and annotated. The sensor event sequences for each activity were extracted from the data set and used to identify the core events, sub events and the ordering relationship of the events. There were no core events for the two activities 'use toilet' and 'prepare drink'. The list of relevent events with the corresponding weights calculated as explained in section 3, for these two activities are given below.

'<u>Use Toilet</u>'

Hall-ToiletDoor	: 0.3632
ToiletFlush	: 0.3828
HallBedroomDoor	: 0.0437

'<u>Prepare Drink</u>'

Fridge	: 0.4629
CupsCubboard	: 0.3214
Freezer	: 0.0014
Dishwasher	: 0.0014

To test the performance of the system, one day of sensor events were used as test data. The three causes of abnormalities, viz. irrelevent events, out of order events and missing out events, were introduced manually in the test data. The system located all of these abnormalities and generated appropriate text messages.

The effectiveness of the proposed method mainly depends on the training event sequences. The training event sequences should reflect all the possible ways in which an activity can be performed without any abnormality. There should be at least one event sequence for each of the possible ways of peforming the activity.

V. Conclusion

In this paper a simple and novel way of detecting abnormalities in the activities perfomed by people with early stages of dementia is proposed. Such people tend to skip some events of an activity or perform events in improper order besides introducing irrelevant events while pursuing activities of daily living. The proposed method can be used to effectively identify such abnormalities and to offer appropriate corrective suggestions. Since the relevant event and their ordering information is captured and stored for each normal activity separately, the proposed method can be used for monitoring interleaved activities also.

References

[1] D.J. Cook and M. Schmitter-Edgecombe, "Assessing the Quality of Activities in a Smart Environment", Methods of Information in Medicine, volume 48, issue 5, 2009, pp. 480-485.

[2] Patrice C. Roy, Sylvain Giroux, Bruno Bouchard, Abdenour Bouzouane, Clifton Phua, Andrei Tolstikov, and Jit Biswas, "A Possibilistic Approach for Activity Recognition in Smart Homes for Cognitive

Assistance to Alzheimer's Patients", Activity Recognition in Pervasive Intelligent Environments, Atlantis Ambient and Pervasive Intelligence, Volume 4, 2011, pp 33-58.

[3] Xin Hong, Chris D. Nugent, Maurice D. Mulvenna, Suzanne Martin, Steven Devlin, Jonathan G. Wallace, "Dynamic similarity-based activity detection and recognition within smart homes", International Journal of Pervasive Computing and Communications, Vol. 8 Iss: 3, 2012, pp. 264 – 278.

[4] Paul Cuddihy, Jenny weisenberg, Catherine Graichen, Meena Ganesh, "Algorithm to Automatically Detect Abnormally Long Periods of Inactivity in a Home", Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments, 2007, pp. 89 - 94.

[5] V. Jakkula and D.J. Cook, "Anomaly Detection Using Temporal Data Mining in a Smart Home Environment", Methods of Information in Medicine, volume 47, issue 1, 2008, pp. 70-75.

[6] Jie Yin, Qiang Yang and Jeffrey Junfeng Pan, "Sensor-Based Abnormal Human-Activity Detection", IEEE Transactions on Knowledge and Data Engineering, Volume:20, Issue: 8, Aug. 2008, pp. 1082 – 1090.

[7] Derek Hao Hua, Xian-Xing Zhangb, Jie Yinc, Vincent Wenchen Zhenga and Qiang Yang,"Abnormal Activity Recognition Based on HDP-HMM Models", Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence, 2009, pp. 1715-1720.

[8] Irina Mocanu and Adina Magda Florea, "A Model for Activity Recognition and Emergency Detection in Smart Environments", AMBIENT 2011 : The First International Conference on Ambient Computing, Applications, Services and Technologies, 2011, pp. 13-19.

[9] Vikramaditya Jakkula and Diane J. Cook, "Detecting Anomalous Sensor Events in Smart Home Data for Enhancing the Living Experience", Association for the Advancement of Artificial Intelligence, 2011, pp. 33-37.

[10] van Kasteren, T., Noulas, A., Englebienne, G. and Kro¨se, B., "Accurate activity recognition in home setting", Proc. 10th Int. Conf. Ubiquitous Computing, ACM 2008, pp. 1-9.