IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, **p**-ISSN: 2278-8719
Vol. 3, Issue 12 (December. 2013), ||V3|| PP 40-45

Automatic Construction of Finite Automata for Online Recognition of User Activities in Smart Environments

H. Karamath Ali, Dr. D. I. George Amalarethinam
[1]*(Research Scholar in Computer Science, Jamal Mohamed College(Autonomous), Tiruchirappalli, India.*
[2]*(Associate Professor & Director-MCA, Jamal Mohamed College(Autonomous), Tiruchirappalli, India.*

Abstract: - Recognizing user activities unobtrusively is very important for offering context aware services in pervasive computing environments. Because of their ability to classify given inputs in a simple and efficient manner, Finite State Machine(FSM)s have good potential to be used for recognizing user activities. However, the manual construction of FSMs in the existing methods require a domain expert with enough insight to dissect the activity of interest into an FSM. Obviously, this requirement imposes huge burden on the experts when constructing FSMs for recognizing activities that are not so repetitive in nature. In this paper, we propose a method for automatic construction of an FSM that can be used to recognize user activities in real time. The proposed method gives promising results when tested with publicly available smart home datasets.

Keywords: - *Activity Recognition, Activities of Daily Living, Finite Automata,Online recognition, Pervasive Computing.*

I. INTRODUCTION

To offer context-aware services in pervasive computing environments, it is essential to recognize the activities performed by the user in real time[1]. Pervasive computing environments have almost every possible object embedded with some kind of sensor. Besides these, the user may be wearing sensors like accelerometers and gyroscopes. Each sensor generates a stream of data as the user proceeds through his activities by interacting with the objects in the environment. The streams of data get collected by a dedicated node by means of wireless network. The activity currently being performed by the user has to be identified using these time series data of sensor readings. To accomplish this, researchers have proposed many probabilistic and structural methods[1][2]. Use of finite state machines(FSMs) is one among them. Though relatively less, considerable research has been done to recognize user activities using finite state machines. All the methods that use FSMs, require the structure of the performed activities to be defined by an expert. This requirement can be fulfilled easily for simple repetitive activities like walking and jogging. But for other activities of daily living (ADL) like cooking, washing, cleaning, etc. this requirement puts a lot of burden on the expert. This is because the number, order, and duration of the different steps involved in such activities vary significantly, even when the activities are performed by the same user at different times. So, in our previous work[3] we proposed a method for constructing a fuzzy finite automata to recognize user activities directly from the streams of sensor data, with no user intervention. In this paper we propose to enhance it with the ability to recognize user activities in real time. The proposed enhancement provides encouraging results, when tested with publicly available data sets collected in smart home environments.

The remainder of this paper is organized as follows: Section II presents an overview of the related work in activity recognition; section III defines the problem statement; section IV explains the proposed method; section V discusses the dataset used and the experiment conducted; and section VI presents conclusion.

II. RELATEDWORK

Among the various models developed by researchers for solving the problem of activity recognition the Hidden Markov Model(HMM) is one of those most frequently used . Tim van Kasteren et al.[4] recorded a dataset consisting of 28 days of sensor data in a smart home environment and through a number of experiments showed how the HMMs perform in recognizing activities. Sanchez, D. et al.[5] trained a discrete HMM to map contextual information to a user activity. The model was trained and evaluated using data captured from 200 hours of detailed observation and documentation of hospital workers. An approach for multi-person activity recognition in an office environment using audio as well as video features, gathered from a simple setup, are used to employ a multilevel HMM framework by Christian Wojek et al.[6]. Human activities embedded in video sequences acquired in an archeological site are automatically recognized using Discrete HMMs by Marco Leo, et al.[7]. Weiyao Huang, et al.[8] used Discrete HMM for human posture training, modeling and activity matching to recognize the human motion. Other HMM variations such as Coupled HMM, Fuzzy HMM, Hierarchical HMM and Reconfigurable HMM are used by other researchers[9][10][11][12].

By considering the direction change between frames to track the motion of several limbs and to recognize different activities, Rocío Díaz de León et al.[13] used a Bayesian network for recognition of continuous activities. E.M. Tapia et al. [14] illustrated a system that uses naïve Bayesian classifier for recognizing activities in the home setting using a set of small and simple state-change sensors. The temporal aspects of the activities of daily living(ADL) of elders were modelled by Tim van Kasteren and Ben Kr¨ose[15] using a Dynamic Bayesian Network. According to them, adding more sensors does not necessarily increase accuracy. A d-level dynamic Bayesian network was used to perform complex event recognition by Justin Muncaster and Yunqian Ma[16]. The higher levels of the network are used to enforce state hierarchy while the duration of simplest events are modelled using the dth level.In the work presented by Somboon Hongeng, et al.[17] several layers of naive Bayesian classifiers are used to gather properties of mobile objects from a sequence of image frames. These properties are then used to calculate the occurrence probability of a scenerio.

A framework for an intelligent bedside monitor that derived the current status of a patient using fuzzy state transitions on pre-processed input obtained from clinical instrumentation was presented by Friedrich Steimann and Klaus-Peter Adlassnig [18]. This is one of the earlier works that used fuzzy finite automata for activity recognition. To recognize activities like working in the desk room, crossing the corridor, having a meeting, etc., A. Alvarez-Alvarez, et al.[19] used a Fuzzy Rule based Classifier and two Fuzzy Finite State Machines(FFSM)s. The classifier approximated the position of the user at the level of discrete zones such as office, corridor and meeting room. One of the FFSM was used for human body posture recognition and the other FFSM combined the localization and posture recognition. The use of fuzzy finite state systems was demonstrated in [20] for human gait modeling. Gonzalo Bailador and Gracián Trivièo[21] proposed a syntactic pattern recognition approach based on temporal fuzzy automata, to cope with the variability of patterns by defining imprecise models. The approach was called temporal fuzzy automata as it allowed the inclusion of time restrictions to model the duration of the different states. This approach was then used for recognizing hand gestures. Thiago Teixeira et al.[22] presented an activity recognition system for assisted living applications and smart homes. The user was located using camera nodes placed on the ceiling. Direction and motion information was obtained using an inertial sensor placed on the subject's arm, an accelerometer, a gyroscope and a magnetometer which are worn by the user. These four measurements were parsed using a lightweight hierarchy of finite state machines. The finite state machines were manually constructed by "a field expert - someone with enough insight to be able to dissect the activity of interest into an FSM". Based on a hierarchical finite state machine(FSM) Hoang Trinh et al.[23] proposed a technique to detect check-out related primitive activities in a retail store. Their technique used visual features and predefined spatial constraints on the hand motion to capture particular motion patterns performed in primitive activities. The technique is applied to the problem of retail fraud detection. The FSM was constructed manually as the number of component activities of hand motion in a check-out counter is very small, and are often repeated in the same order. Dhruv Mahajan et al.[24] presented a framework for activity recognition and detection of unusual activities in video data. The framework was based on a model of physical, logical and event layers of finite state machines. The finite state machine layers learned, in an unsupervised mode, usual patterns of activities in a scene over long periods of time. Then, in the recognition phase, usual activities are accepted as normal and all other activity patterns are set as abnormal.

Though finite automata are used in the above works to recognize human activities, an expert has to manually construct the states and the transitions between them. This will be onerous, since users perform activities in different ways at different times. So in this paper it is proposed to extend the algorithm suggested by Henning Fernau [25] to automatically construct a DFA to recognize a given set of sensor event input sequences. The algorithm is extended by storing the class of activity corresponding to the transition path in which a state appears, in that state. So, as soon as a state is reached, it is possible to decide which activity the user is performing, without waiting for reaching some final state. This ability is useful to recognize activities in real time environments. The proposed method is tested with a publicly available data set, and the results are comparable to other often used methods.

III. THE PROBLEM

The objective is to recognize activities from sensor readings in a smart environment. For this, the time series data of sensor readings is divided into time slices of constant length. Each time slice is labeled with the activity performed during that time slice [4]. A vector $\vec{x}_t = (x_t^1, x_t^2, \ldots, x_t^N)^T$is used to represent the sensor readings at time slice t, where x_t^i represents the input from the sensor x^i during the time slice and 'N' is the number of sensors. The activity performed during time slice 't' is represented by y_t. So, the task of the activity recognition system is to find an association between a sequence of observation vectors **x** = $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$ and a sequence of activity labels **y** = $\{y_1, y_2, \ldots, y_n\}$. Figure 1 given below illustrates the above setup, with Δt representing a time slice.

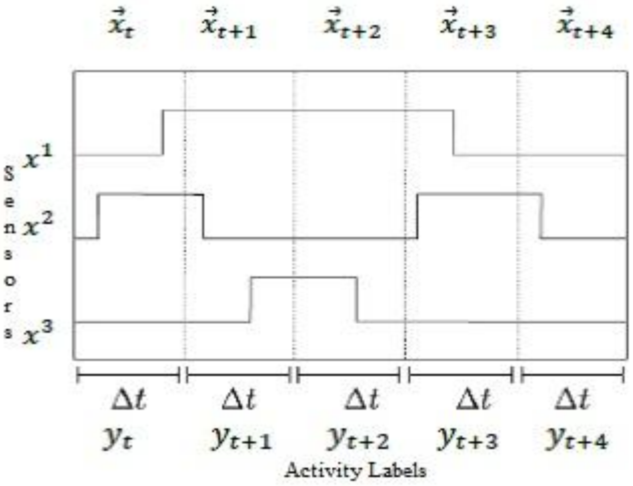


Figure 1. Sensor readings and time slices

IV. THE METHOD

As explained in section III, the time series data of sensor readings is divided into time slices of equal length. Each time slice is labeled with the activity performed during that time slice. A group of subsequent observation vectors $(\vec{x}_t, \vec{x}_{t+1}, \vec{x}_{t+2}, \ldots, \vec{x}_{t+m})$ for which the corresponding activity labels $y_{t+i} = k,\ 0 \le i \le m$, is represented by X_k. As an example, one such group for $m = 8$ and $k\ =\ 3$ is given in fig.2. Here it is assumed that there are six different streams of sensor inputs.

$\vec{x}_t$	$\vec{x}_{t+1}$	$\vec{x}_{t+2}$	$\vec{x}_{t+3}$	$\vec{x}_{t+4}$	$\vec{x}_{t+5}$	$\vec{x}_{t+6}$	$\vec{x}_{t+7}$	$\vec{x}_{t+8}$
0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	1	1
0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0
y_t	y_{t+1}	y_{t+2}	y_{t+3}	y_{t+4}	y_{t+5}	y_{t+6}	y_{t+7}	y_{t+8}
3	3	3	3	3	3	3	3	3

Figure 2. Observation vectors and activity labels

By omitting the repeated occurrences of a vector in more than two of the subsequent time slices, the above sequence may be condensed as given in fig.3.

The condensed group shown in fig.3 is called the *condensed vector sequence (*CVS) ofX_k and is represented by C_k[26]. In the CVS the subsequent occurrence of a vector implies that the vector gets repeated a certain number of times in the actual data set. The maximum number of times the vector consequently appears in the data set can be used to decide the maximum duration the DFA may be allowed to remain in the corresponding state.

0	1	1	1	0	0
0	0	0	1	0	0
0	0	0	0	1	1
0	1	1	1	0	0
0	0	0	0	1	1
0	0	0	0	0	0

Figure 3. Condensed Vector Sequence

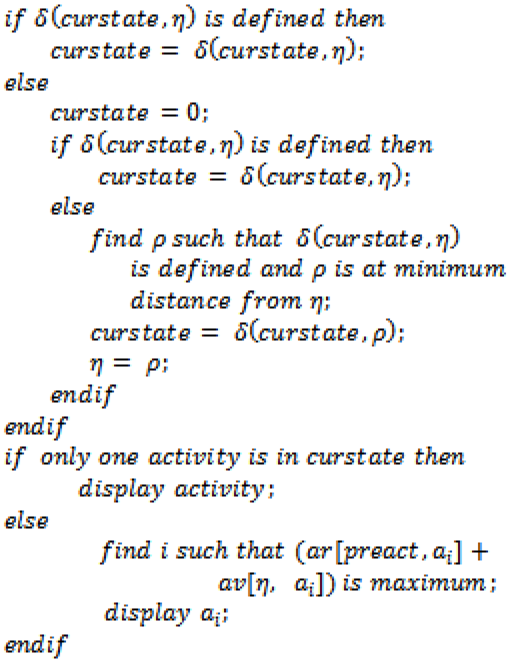
The set of CVSs of all such sequences of observation vectors for activity k is denoted by $\mathcal{X}_k$, that is, $\mathcal{X}_k = \{C_{k,1}, C_{k,2}, \ldots, C_{k,p}\}$, where p is the number of occurrences of activity k in the data set. For each observed activity k, the corresponding $\mathcal{X}_k$ is extracted from the data set. By assigning a unique numeric code for each vector in $\mathcal{X}_k$, the equivalent set of code sequences $\mathcal{S}_k$ is formed. That is, $\mathcal{S}_k = \{s_{k,1},\ s_{k,2}, \dots, s_{k,p}\}$, where $s_{k,i}$ is the string equivalent of $C_{k,i}$ for $1 \le i \le p$.

Hence, a single set $\mathbb{S}$, consisting of all possible code sequences for all the observed activities is formed as the union of all$\mathcal{S}_k$s. That is, $\mathbb{S} = \bigcup_{1\le k\le n} \mathcal{S}_k$ where n is the number of activities. For $\mathbb{S}$, a DFA is constructed using the SL-infer algorithm suggested by Henning Fernau [25].

Usually in a finite automaton, the class or set to which an input sequence belongs, is decided depending upon the final state reached after scanning the input sequence. This is not suitable for the problem of real time activity recognition. Therefore in each state I^a of the SL-infer algorithm, the corresponding activity (or activities) of the code sequence in which 'a' appears is stored. So given the current state of the DFA, the activity corresponding to the input sequence that led to the state can be decided by accessing the activity data stored in that state.

To resolve ambiguity when more than one activity is stored in a state, two tables namely av and ar, are maintained. av is of size $v \times n$, where v is the number of different vectors encountered in the training data. $av(i,j)$ is the number of times vector $i$appeared in activity j. ar is of size $n \times n$. $ar(i,j)$ is the number of times activity i is followed by activity$j$ in the training data. The way the information in these two tables is used is explained in the algorithm given in fig. 4.

To decide the corresponding activity for a vector $\vec{x}$ in a given sequence of vectors, its numeric equivalent 'η', as assigned earlier, is found and the following algorithm given in fig. 4 is used. In the algorithm $curstate$and preact represent the current state of the DFA and previous activity respectively. Initially curstate = 0 representing the initial state, and preact = -1 indicating there is no previous activity. As given in the algorithm, when there is no transition from the current state on input η, the DFA is taken to the initial state on the assumption that a new activity begins. If there is no transition from the initial state also then an input ρ that has minimum distance from η and for which a transition is defined, is found. This is based on the reason that input η might not have occurred in the test data and the activity for an input that closely resembles η, can be the activity for η.



```
if $\delta(curstate, \eta)$ is defined then
    $curstate = \delta(curstate, \eta)$;
else
    $curstate = 0$;
    if $\delta(curstate, \eta)$ is defined then
        $curstate = \delta(curstate, \eta)$;
    else
        find $\rho$ such that $\delta(curstate, \eta)$
            is defined and $\rho$ is at minimum
            distance from $\eta$;
        $curstate = \delta(curstate, \rho)$;
        $\eta = \rho$;
    endif
endif
if only one activity is in curstate then
        display activity;
else
        find i such that $(ar[preact, a_i] +$
                    $av[\eta,\ a_i])$ is maximum;
        display $a_i$;
endif
```

Figure 4. Algorithm for deciding activity in a state

In the algorithm a_i represents the ith activity stored in the current state. At the end of the algorithm, the activity corresponding to the given vector code η is displayed.

V. DATA AND EXPERIMENT

A number of activity datasets, collected in experimental smart environments are publicly available. We have used one such data set collected and made public by Tim van Kasteren, et al.[27]. The datasets have been collected by observing the behavior of inhabitants inside the homes using wireless sensor networks. Output of the binary sensors have been annotated with the activities performed by the subjects during predefined time intervals. A dataset so collected for 25 days of 10 activities such as preparing dinner, using washroom and sleeping, is used by us to conduct the experiment. Three different representations, namely, raw, change point and last-fired, of the sensor data are used[27]. The raw sensor representation uses the sensor data directly as it was received from the sensors. The change point representation indicates when a sensor event takes place. The last-fired sensor representation indicates which sensor fired last. For each of the representations, six different time slices (600, 300, 60, 30, 10 and 1 seconds) have been used. Sensor input vector sequences for all the 10 activities, with each representation and time slice, were culled from the data set.

To study the effectiveness of the algorithm explained in the previous section, we followed the 'leave one day out' approach[26]. In this approach, one full day of sensor readings is used for testing and the remaining days are used for training. This is repeated for all the days, for each representation with each time slice. So the experiment was repeated 450 (= 25 days * 3 representations * 6 time slices) times, and the average performance was measured by calculating $recall,\ precision$, $F_measure$and $accuracy$as follows.

$$recall\ =\ tp\ /\ (tp\ +\ fn)precision\ =\ tp\ /\ (tp\ +\ fp)$$

$$F\_measure\ = \frac{2(recall * precision)}{(recall\ +\ precision)} accuracy = (tp+tn)/(tp+tn+fp+fn)$$

where tp, fp and fn represent the number of true positives, false positives and false negatives.

The average F-measure obtained by our method for the raw, change point and last-fired representations are summarized in Ttable 1for 60 seconds time slice. As given in the table the proposed method produces better result for the last fired feature representation. This is followed by raw data. The result for change point also is promising. The average measure values are comparable to the performance measures obtained by the methods described by Kasteren et al.[26]. So, the method proposed by us can be further explored for online recognition of user activities.

Table 1. Performance Measures

	Precision	Recall	F-measure	Accuracy
RawData	57.31±17.6	56.62±16.9	56.51±16.5	85.19±10.4
Change	51.32±20.1	49.41±18.2	49.88±18.7	64.20±16.9
LastFired	66.58±14.6	61.18±16.1	63.22±14.4	93.79±5.50

VI. CONCLUSION

In this paper, we have proposed a method for automatic construction of a DFA for online recognition of user activities in a smart environment. Since the results obtained are promising, we are planning to test the method with other activity data sets. The number, order, and duration of the different steps involved in an activity vary significantly, even when the activity is done by the same user at different times.So, fuzziness in the constructed DFA can be incorporated in order to deal with these variations as explained in our earlier work[3].

REFERENCES

[1] Liming Chen and Ismail Khalil, Activity Recognition : Approaches, Practices and Trends, Chapter 1, *Activity Recognition in Pervasive Intelligent Environments*, (Atlantis Press, 2011*)*, pp. 1-29,.

[2] Oscar D Lara and Miguel A Labrador, A Survey of Human Activity Recognition using wearable sensors, *IEEE Communications Survey and Tutorials, Volume 15, Issue 3*, November, 2012.

[3] H. Karamath Ali and D.I. George Amalarethinam, Activity Recognition with Multi tape Fuzzy Finite Automata, *International Journal of Modern Education and Computer Science*, pp 60-67, June 2013.

[4] Tim van Kasteren, AthanasiosNoulas, Gwenn Englebienne and Ben Kr¨ose, Accurate Activity Recognition in a Home Setting, *Proceedings of the 10th international conference on Ubiquitous computing*, pp 1-9, 2008.

[5] Sanchez, D.; Tentori, M.; Favela, J., Activity Recognition for the Smart Hospital,*IEEE Intelligent Systems, Volume: 23, Issue: 2*, pp. 50 – 57, March-April 2008.

[6] Christian Wojek, Kai Nickel, Rainer Stiefelhagen, Activity Recognition and Room-Level Tracking in an Office Environment, *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 25 – 30, Sept. 2006.

[7] Marco Leo, Paolo pagnolo, TizianaD'Orazio, and ArcangeloDistante, Human Activity Recognition in Archaeological Sites by Hidden Markov Models, *Advances in Multimedia Information Processing - PCM 2004, Lecture Notes in Computer Science Volume 3332*, pp. 1019-1026, 2005.

[8] Weiyao Huang, Jun Zhang, and Zhijing Liu, Activity Recognition Based on Hidden Markov Models, *Proceedings of the 2nd international conference on Knowledge science, engineering and management*, pp. 532-537, 2007.

[9] Matthew Brand, Nuria Oliver, and Alex Pentland, Coupled Hidden Markov Models for complex action recognition, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp.994, 1997.

[10] Xucheng Zhang, FazelNaghdy, Human Motion Recognition through Fuzzy Hidden Markov Model, *Proceedings of the 2005 International Conference on Computational Intelligence for Modeling, Control and Automation, and International Conference onIntelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)*, pp.450-456, 2005.

[11] SerafeimPerdikis, DimitriosTzovaras, Michael GerasimosStrintzis, Recognition of human activities using Layered Hidden Markov Models, *ZL50 digital Library, http://en.zl50.com*, 2006

[12] Md. KamrulHasan, HusneAraRubaiyeat, Yong-Koo Lee, Sungyoung Lee, A Reconfigurable HMM for Activity Recognition, *10th International Conference on Advanced Communication Technology, Volume: 1*, pp. 843-846, 2008.

[13] RocíoDíaz de León, L. Enrique Sucar, Recognition of Continuous Activities, *Proceedings of the 8th Ibero-American Conference on AI: Advances in Artificial Intelligence*, pp. 875-881, 2002.

[14] Emmanuel Munguia Tapia Stephen S. Intille Kent Larson, Activity Recognition in the Home Using Simple and Ubiquitous Sensors, *Proceedings of Second International Conference on Pervasive Computing*,pp.158-175, 2004

[15] Tim van Kasteren and Ben Kr¨ose, Bayesian Activity Recognition in Residence for Elders, *3rd IET International Conference on Intelligent Environments (IE 07)*, pp. 209 – 212, 2007.

[16] Justin Muncaster, Yunqian Ma, Activity Recognition using Dynamic Bayesian Networks with Automatic State Selection, *Proceedings of the IEEE Workshop on Motion and Video Computing*, pp. 30, 2007.

[17] SomboonHongeng, Francois Br´emond and RamakantNevatia, Bayesian Framework for Video Surveillance Application, *Proceedings of the 15th International Conference on Pattern Recognition*, 2000.

[18] Friedrich Steimann and Klaus-Peter Adlassnig, Clinical Monitoring with Fuzzy Automata, *Journal of Fuzzy Sets and Systems, Volume 61 Issue 1*, pp. 37-42, Jan. 10, 1994.

[19] A. Alvarez-Alvarez, J. M. Alonso, G. Trivino, N. Hern´andez, F. Herranz, A. Llamazares and M. Oca˜na, Human Activity Recognition applying Computational Intelligence techniques for fusing information related to WiFi positioning and body posture, *2010 IEEE International Conference onFuzzy Systems (FUZZ)*, pp. 1 – 8, 18-23 July 2010

[20] Alberto Alvarez-Alvarez, GracianTrivino, Oscar Cord´on, Human Gait Modeling Using a Genetic Fuzzy Finite State Machine, *IEEE T. Fuzzy Systems 20(2)*, pp.205-223, 2012.

[21] Gonzalo Bailador, GraciánTriviòo, Pattern recognition using temporal fuzzy automata, *Journal of Fuzzy Sets and Systems, Volume 161 Issue 1*, pp. 37-55, January, 2010

[22] Thiago Teixeira, Deokwoo Jung, Gershon Dublon, Andreas Savvides, Recognizing Activities from Context and Arm Pose using Finite State Machines, *Third IEEE/ACM International Conference on Distributed Smart Cameras*, pp. 1-8. 2009.

[23] Hoang Trinh, Quanfu Fan, Jiyan Pan, Prasad Gabbur, Sachiko Miyazawa, Sharath Pankanti, Detecting Human Activities In Retail Surveillance Using Hierarchical Finite State Machine, *IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, pp. 1337-1340, May 2011.

[24] Dhruv Mahajan, Nipun Kwatra, Sumit Jain, Prem Kalra, Subhashis Banerjee, Framework for Activity Recognition and Detection of unusual Activities", 2004

[25] Henning Fernau, Algorithms for learning regular expressions from positive data, *Journal of Information and Computation, Volume 207, Issue 4*, pp. 521-541, April 2009

[26] H. Karamath Ali and D. I. George Amalarethinam, Activity Recognition with Fuzzy Finite Automata", to appear in *Proceedings of World Congress on Computing and Communication Technology*, 2014.

[27] T.L.M. van Kasteren, G. Englebienne, and B.J.A. Kröse, Human Activity Recognition from WirelessSensor Network Data: Benchmark and Software, *Chapter 8, Activity Recognition in Pervasive Intelligent Environments, Atlantis Press*, pp. 165-185, 2011.