Abnormal Human Behavioral Pattern Detection in Assisted Living Environments

Kyungseo Park, Yong Lin, Vangelis Metsis, Zhengyi Le, Fillia Makedon

Heracleia Human Centered Computing Lab.
Computer Science and Engineering
The University of Texas at Arlington
Arlington, TX 76019
{kpark,ylin, vmetsis, zyle,makedon}@uta.edu

ABSTRACT

In recent years, there is a growing interest about assisted living environments especially for the elderly who live alone, due to the increasing number of aged people. In order for them to live safe and healthy, we need to detect abnormal behavior that may cause severe and emergent situations for the elderly. In this work, we suggest a method that detects abnormal behavior using wireless sensor networks. We model an episode that is a series of events, which includes spatial and temporal information about the subject being monitored. We define a similarity scoring function that compares two episodes taking into consideration temporal aspects. We propose a way to determine a threshold to divide episodes into two groups that reduces wrong classification. Weights on individual functions that consist the similarity function are determined experimentally so that they can produce the good results in terms of area under curve in receiver operating characteristic (ROC) curve.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Assisted living environment, abnormal behavior detection, similarity function, threshold value, classification

1. INTRODUCTION

Abnormal behavior detection can be used in many useful fields such as surveillance systems, network intrusion detection, and healthcare monitoring systems. Especially, in healthcare monitoring systems, most of the research has used images from video cameras and analyzed them with image related techniques. However, the elderly or patients usually do not want to be monitored by the means of video cameras, because of privacy violation issues. Therefore, we propose a human behavior detection system that does not depend on images from video camera, but uses noninvasive wireless sensor techniques.

Most work that is related to an abnormal human behavior detection has focused on recognizing the current situation. That means, they try to figure out what happens currently in terms of high level context. They also have used the Hidden Markov Model (HMM) technique or variations of it to recognize the human behavior through hidden states such as low level sensing data. In this work, we do not focus on human behavior recognition that gives us high level context, but we try to reach the step of determining abnormal behavior using low level similarity comparison. This is based on our assumption that sensor data can be unreliable. We adopt the concept of pattern classification that divides data into multiple classes, each of which includes its own characteristics. In our case, the number of classes that we need is two, one class that includes normal behavior, and the other class that includes abnormal behavior.

We define an event as an outcome of interest from a sensor. We also define an *episode* as a series of events. An event is a 3-tuple, which includes a sensor ID that can represent the location of the sensor or a binary status of an object, a time stamp when a sensor is activated, and a duration that is time difference between two sensors activated consecutively. Thus, an episode can represent the series of basic actions performed by a person, where each action is modeled as an event. To determine abnormal behavior, we consider not only the sequence of events, but also temporal aspects when the individual event happens and how long it lasts.

The purpose of this work is to determine whether an episode taking place is normal or abnormal by low level pattern classification and get a best weight of individual similarity function. If it is abnormal, then the system has to make a warning so that it can grab caregiver's interest. First, we create a similarity function that considers four aspects, sequence of events, the number of common events, time, and duration. We adopt a well-known algorithm, longest common subsequence (LCS) for the similarity function. Second, we determine a threshold value to decide normalcy by using a training set with a sample set of normal episodes and a sample set of abnormal episodes. Third, we determine weights of individual functions that consist the similarity function so that they can produce the best results in terms of area under curve in the receiver operating characteristic

PETRA'10 , June 23-25, 2010, Samos, Greece
Copyright 2010 ACM ISBN 978-1-4503-0071-1/10/06 ...$10.00.

(ROC) curve. The contribution of this paper is to use a similarity function in a series of events to determine abnormal human behavior in the domain of sensor data that may include faulty values.

The rest of this paper is organized as follows: Section 2 reviews the related work. Section 3 defines events, episodes, and abnormal behavior. We propose our similarity function and a threshold based classification scheme in section 4. Section 5 shows our experimental results. Finally, we conclude our work and discuss future work in section 6.

2. RELATED WORKS

Lymberopoulos et el. treated the problem of extracting the spatiotemporal human activity model from an assisted living home environment using sensor networks in [14]. They used their data model, $<$ location, time, duration $>$, to extract human activity patterns and apriori algorithm, which was proposed by Agrawal [2]. They used tracking cameras, door sensors, and passive infrared sensors to detect human behavior and used 30-day actual data set to extract the person's daily pattern. This is useful to find frequent patterns, but it does not handle the issue of detection of outlier or abnormal behavior. Lühr et al. proposed intertransaction association rule (IAR) mining to detect deviations from normal behavior in smart home occupants in [13]. They used Extended Frequent Pattern Tree (EEP-Tree), which is an extension of the Frequent Pattern Tree (FP-Tree) proposed by Han. They applied their algorithm to a real world data set, which was provided by [16]. The data set was collected by two people (30s and 80s) of daily behavior for 16 days. They used state-change sensors that detect on/off status installed at toilet flush, sink faucet, closet, door, drawer, freezer, medicine cabinet, etc. They focus on finding emergent behavior. Data model is $<$ location, event, start time, end time$>$.

There are several previous works that used classification techniques in the application of recognition of human being's behavior or healthcare monitoring systems [16, 10]. Tapia et al. built a system for experiment to recognize human being's activities using low-cost state-change sensors in [16]. They developed simple state-change sensors instead of using multi-purpose sensor motes and cameras in order to minimize intrusion of privacy. Initially, they get pre-knowledge about what the residents are doing at particular moment by the context-aware experience sampling tool (ESM) so that the information can be used as a training set. Then the system calculates the temporal features (exists and before) to generate training examples. A Bayesian network classifier is used to calculate the probability of the current activity. Haghighi et el. proposed an architecture for context-aware adaptive data stream mining in [10]. They used temperature, age, location, time, heart rate, and battery level for context attributes to monitor heart patients. They used K-nearest neighbor algorithm to classify unknown situations based on the pre-defined situations.

In the area of human behavior recognition, variations of HMM techniques have been used in a number of works. [3] gives a model for human activities recognition. They use Bayesian networks to merge heterogeneous data and HMM to detect a particular activity. [5] can detect abnormal behavior. It also uses DHMM and HC-HMM, which considers duration (D-HMM) and hierarchical architecture (HC-HMM). [11] uses HMM to classify daily activities, such as lying, sitting, and standing, which can be classified into stable states. These works are similar to our work in that they treat human behaviors. But they are different from our work in that they try to recognize human behavior through observations.

"Abnormality" has been researched in many areas such as network intrusion, visual surveillance, and human behavior or activity. First, [1, 4, 8] are all about Intrusion Detection System (IDS), where a system tries to detect abnormal traffic or predict the traffic in advance. Second, an application is about visual surveillance, where abnormal behavior can be detected by analyzing images that were taken by video surveillance cameras. These include [9, 17]. Third, one is about abnormal human behavior detection, which recognizes human Activities of Daily Living (ADL) [7, 12].

3. DEFINITIONS

We define an event in section 3.1, an episode in section 3.2, and abnormal behavior that will be used throughout this paper in section 3.3.

3.1 Events

An *event* is a 3-tuple, which includes a sensor ID, a time stamp, and a duration. We let e_i be an event, where i indicates the order of activated sensors.

$$e_i = (S, T, D) \tag{1}$$

where S is a sensor ID that can represent the location of the sensor or an individual action, T is a time stamp when the sensor is activated, and D is a duration, which is time difference after one sensor is activated until the next sensor is activated.

Sensors can be attached on an object or placed on a particular location to detect appropriate actions done by a person. For example, a sensor located on a hallway can detect the change of intensity of light, when a person passes by it (S). In this case, the sensor ID indicates both the location of the sensor and the action (someone is passing by it.) When the sensor detects the change, it can also record the time stamp. This indicates the time, when the sensor is activated (T). The format of time stamp is hh:mm:ss, whose granularity is the second. Time stamp also includes date, whose format is mm/dd/yyyy. If another sensor detects the other action, it also records the activation time. Hence, we can simply calculate a duration between two activation times by subtracting the former one from the latter one (D). Since the granularity of time stamp is the second, we use the same one for duration.

3.2 Episodes

An *episode* is a series of events. We let E_i be an episode, where i indicates the index and define it as a sequence.

$$E_i = (e_1, e_2, ..., e_n) \tag{2}$$

The order of events in an episode is determined by the timestamp T of e_i. For example, when a person walks from a bedroom to a kitchen through a hallway, three sensors may react by detecting change of light intensity. In this case, we have three events, e_1, e_2, and e_3, which are corresponding to a sensor at a bedroom, a sensor on a hallway, and a sensor at a kitchen, respectively. Generally, an episode can represent a meaningful activity, such as ”a person comes to a kitchen to drink a cup of water, through a hallway, from a bedroom”, or ”after watching a TV, a person enters into a bedroom to sleep.” But in our work, since we did not define the episode as such, it is nothing but a series of n events.

3.3 Abnormal Behavior

We define ”abnormal behavior” as ”an episode which was not done before at all, an episode which was rarely done before, or an episode which was not close enough to the one previously done.” But this is not enough to define abnormal behavior since we do not consider temporal aspects in episodes. First, we need to consider time and add it to the definition that ”an episode whose sequence of events are similar to the previous one, but the time that the episode happened is far different from the previous one.” Second, we need to consider the duration of each event. Same sequences of events that happened at similar times can have different duration. An example includes that a person goes to a bathroom at 1:00 am, and usually stays less than 10 minutes, but if the same person stays at the bathroom for longer time, which should be regarded as an abnormal behavior. Therefore, we need to add it to the definition that ”an episode whose sequence of events are similar and whose time it happened is close to the previous one, but whose duration for each event is not close enough to the previous one.”

4. ABNORMAL BEHAVIOR DETECTION

In this section, we describe the overview of our system that detects abnormal human behavior, the function for measuring similarity, and the method to determine the threshold to divide human behaviors into two groups.

4.1 Overview

Data can be collected by SunSpot sensors as a type of events. Events can include information such as ”a person is passing by hallway” or ”a person is leaving out of a bed.” The actual implementation is described in [15]. Once the system gathers events from SunSpots whenever they are activated, it can have a series of events, which forms an episode. This kind of events will be stored on the system, so that they can be used as a training set for the future. After that, the two groups of sample episodes are compared to every episode in the training set using a sliding window method. The two groups are: i) a normal group that is supposed to have most of normal episodes in it, and ii) an abnormal group that is supposed to have most of abnormal episodes in it.

In order to determine the similarity between two comparing episodes (the current episode and the one in the training set), we create a similarity measuring function, which considers the similarity of the sequence, the number of common events, the time in a day, and a duration of time gap between two consecutive activated sensors. The measuring function returns one decimal value, which will be compared to a threshold value so that we can divide the episodes into two groups, normal and abnormal behavior.

Two groups of sample episodes may have common values, which make overlapping groups. Here, we need to determine a threshold value that can divide these overlapping classes into two smaller groups. The way how to determine the threshold value is described in section 4.2.2.

4.2 Similarity Search

The assumption for data gathering from wireless sensor networks is that we can have noisy and unreliable data from the sensors. That means when we get an episode, which is a series of events, some of events in the series might contain faulty data. If we assume that the faulty values cannot be fixed or removed completely by any methods, we just compare episodes that include faulty data. For example, figure 1 shows what happens in faulty data. Suppose that the original episode is a sequence of events, ’a’, ’b’, ’c’, ’d’, and ’e’. And the events from measured sensor data are ’a’, ’f’, ’c’, ’d’, and ’b’. Then ’f’ is an unexpected event due to unreliable sensor reading or transmission, ’b’ is one of the original events but reported delayed, and ’e’ is supposed to appear in the sequence but is missing. In this case, exact pattern matching or sequential pattern mining technique fails to find original episode since they try to find exact sequence. But since we consider that the sequence may contain faulty data, we compare two episodes so that their similarity should be based on the fact ’how much they resemble.’ Two episodes in figure 1 have ’a’, ’c’, and ’d’ as their longest common subsequence and we reflect this for our main measuring function. Event ’b’ is not included in the longest common subsequence, but this could be considered when we determine similarity. Therefore, we create a minor function to handle an event that is not included in the longest common subsequence but that is common in two sequences.

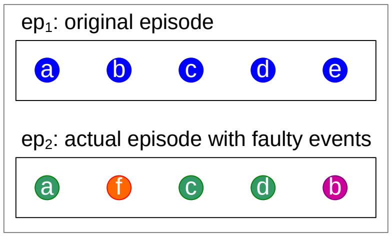


Figure 1: An episode with faulty events

4.2.1 Similarity Function

In this section, we describe how we build the similarity function that is used to compare episodes as a series of events. We define a main similarity function S which incorporates a number of individual similarity sub-functions s_i each of which is given a different weight.

$$S(E_1, E_2) = \sum_{i}^{n} w_i s_i \tag{3}$$

where, E_1 and E_2 are arbitrary episodes, whose lengths are the same, w_i is the weight, and s_i is an individual similarity measuring function. Every s_i is normalized so that it can have a value between 0 and 1.

The individual similarity sub-functions that we use are explained bellow. First, we define s_1 as the individual similarity measuring function comparing longest common subsequences from two episodes. We adopt the existing algorithm of finding longest common subsequences (LCS) [6].

$$s_1 = \frac{l_{LCS}}{k} \tag{4}$$

where, l_{LCS} stands for the length of the longest common subsequences, and k means the total length of an episode, which is the number of events in an episode.

Second, we have s_2 function to consider the situation that was mentioned in section 4.2. That means s_2 considers events which are not in longest common subsequences but just in both sequences. If we define two sequences Q_A, Q_B, and longest common subsequences as Q_{LCS},

$$Q_A = (e_1^a, e_2^a, ..., e_n^a)$$

$$Q_B = (e_1^b, e_2^b, ..., e_n^b)$$

$$Q_{LCS} = (e_1^{LCS}, e_2^{LCS}, ..., e_{n_1}^{LCS}), n_1 \leq n$$

then, we can define multisets that allow to have duplicated elements, A, B, and LCS, whose elements are correspondent to ones in Q_A, Q_B, and Q_{LCS}, respectively.

$$A = \{e_1^a, e_2^a, ..., e_n^a\}$$

$$B = \{e_1^b, e_2^b, ..., e_n^b\}$$

$$LCS = \{e_1^{LCS}, e_2^{LCS}, ..., e_{n_1}^{LCS}\}$$

Now we can define s_2 as follows.

$$s_2 = \frac{|\{e_i | e_i \notin LCS, e_i \in A, e_i \in B\}|}{n} \tag{5}$$

Note that we have multisets instead of general sets that do not allow to have duplicated elements. This is because we need to consider multiple times of same sensor activation in one episode.

Third, in order to consider time factor, we define s_3. This measures how close two episodes in terms of time. It compares timestamps of two events when they are activated.

$$s_3 = 1 - \frac{s_t}{t_C} \tag{6}$$

$$s_t = \frac{\sum_j^{l_{LCS}} |t_j - t'_j|}{n_1}$$

where, t_C stands for time constant, which is 43200 seconds (12 hours), t_j is a timestamp for one of the events in an episode, t'_j is a timestamp for one of the events in the other episode to be compared, and n_1 is the length of LCS. Therefore, the time similarity can be determined only for the events that are in the LCS. But, this is dependent on the sequence of common events, which is determined by s_1. In order to have an independent function to s_1, we prepare variation of s_3, which has k (the length of an episode) instead of n_1 (the length of longest common subsequences).

Fourth, we consider a duration, which is time difference between two consecutive events. As mentioned in section 3.3, among two similar episodes that have same sequences of events and same timestamps, one of them can be determined as abnormal behavior if duration for some of events are much different from the other. To consider this, we define s_4 as follows:

$$s_4 = \frac{\sum_j^{l_{LCS}} \frac{Min(d_j, d'_j)}{Max(d_j, d'_j)}}{n_1} \tag{7}$$

where, d_j is a duration between two consecutive events in one episode and d'_j is a duration between two consecutive events in the other episode, and n_1 is the length of LCS. Once again, we prepare the variation of s_4, which is not dependent on the s_1 by replacing n_1 with k. The function s_4 tries to figure out how similar the duration for matching events in two comparing episodes.

4.2.2 *Comparing new episodes with database.*

The goal of this section is to explain how we can compare newly arriving episodes with the ones stored in our database in order to determine weather a new episode should be considered as normal or abnormal.

As we defined in equations (1) and (2), an event is a 3-tuple that has sensor ID, time, and duration, and an episode is series of events. Based on these, we can define *history sequence*, which contains events by increasing order of timestamp in the individual event. Generally, the number of events in history sequence is much bigger than one in an episode.

$$H = (e_1, e_2, ..., e_m), m \gg n \tag{8}$$

Now, a current episode can be compared to all the episodes in history sequence by using a sliding window method and similarity function. If we define a current episode as $E_c = (e_1^c, e_2^c, ..., e_n^c)$, E_c will be compared to all the subsequences of H, such as $H_1 = (e_1^h, e_2^h, ..., e_n^h)$, $H_2 = (e_2^h, e_3^h, ..., e_{n+1}^h)$, and up to $H_{m-n+1} = (e_{m-n+1}^h, e_{m-n+2}^h, ..., e_m^h)$. Using the similarity function S, we can get the score of all the comparing pairs, (E_c, H_1), (E_c, H_2), and up to (E_c, H_{m-n+1}). After that, we can define H_{max}, which has the highest score among others.

$$i_{max} = \underset{1 \leq i \leq m-n+1}{\text{argmax}} [S(E_c, H_i)] \tag{9}$$

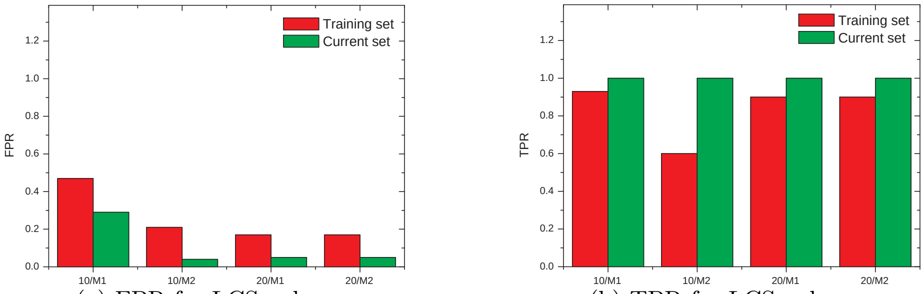
$$H_{max} = H_{i_{max}} \tag{10}$$

Hence, we define the score of the most similar episode to the current episode E_c as follows.

$$S_{max}^c = S(E_c, H_{max}) \tag{11}$$

Whenever a new event happens, a new episode is created. If we have an initial episode that has been used as a current episode, $E_c = (e_1, e_2, ..., e_n)$, and now we have a new event e_{n+1}, then the current episode is replaced by $E'_c = (e_2, e_3, ..., e_{n+1})$. This new episode is now compared to all the subsequences in the history sequence. And we can get $S_{max}^{\prime c}$ as the same way as we did to get S_{max}^c. This is repeated whenever a new event happens. Hence, we can get the set of S_{max}^i for all the episodes newly created.

4.3 Classification using Threshold



(a) FPR for LCS only case (b) TPR for LCS only case

Figure 2: FPR and TPR for training set and current set with a calculated threshold value. 10(20)/M1(M2) stands for length 10(20) episodes and threshold method M1(M2).

It is always possible for sensors to measure wrong value, to respond to an incorrect input (false positive), or not to respond to a correct input (false negative). If these sensor data are transmitted wirelessly, it is more likely to have wrong or unexpected result. Due to these faulty data that can be included in an episode, we may have a normal (or abnormal) episode, which is classified into an abnormal (or normal) one incorrectly. Even though we adopt data cleaning techniques to remove these faulty data in our previous work [15], it is not guaranteed to get rid of all the unnecessary data perfectly. That means it is necessary to consider an episode that contains faulty data. Hence, a training set that is supposed to have all the normal episodes is likely to have several abnormal episodes in it depending on a threshold value that separates episodes into two groups. Also, if we have a set of episodes that is created intentionally to have abnormal data, it is also likely to have normal episodes in it. These result in a fact that we have overlapping classes to be separated.

Since we already applied a similarity function for the classification of episodes, the similarity scores are recorded on the one-dimensional line. From now on, we adopt a simple idea to have a threshold value to classify episodes efficiently. Since the values are one-dimensional, we do not need to use multi-dimensional classifying techniques, such as support vector machine (SVM). The main purpose of threshold value is to classify two one-dimensional overlapping classes so that we have as smallest number of episodes as possible that are classified into the opposite class.

Suppose we have two classes, C_1 and C_2. C_1 has two subsets, C_1^N, which has elements that are classified into C_N, and C_1^A, which has elements that are classified into C_A, where C_N is normal class and C_A is abnormal class. C_2 also has two subsets, C_2^N and C_2^A. Sets of real numbers, C_1 and C_2 are defined as,

$$C_1 = \{a_1, a_2, ..., a_n\}$$

$$C_2 = \{b_1, b_2, ..., b_m\}$$

and C_1^A, C_1^N, C_2^A, and C_2^N are:

$$C_1^A = \{x | x \in C_1, x \leq v_i\}$$

$$C_1^N = \{x | x \in C_1, x > v_i\}$$

$$C_2^A = \{x | x \in C_2, x \leq v_i\}$$

$$C_2^N = \{x | x \in C_2, x > v_i\}$$

where x is a real number and v_i is a threshold value, which can be represented as $0.005 \times i, 1 \leq i \leq 200$. We suggest two methods to determine threshold value. First, we try to get a threshold value that minimizes the ratio of wrong classification. We suggest a threshold that considers true positive rate (TPR) and false positive rate (FPR) and choose a threshold that minimizes sum of 1 - TPR and FPR.

Table 1: Comparison between normal/abnormal set and disease classification

Test | Actual	C_1 (present)	C_2 (absent)
Normal (positive)	C_1^N (TP)	C_2^N (FP)
Abnormal (negative)	C_1^A (FN)	C_2^A (TN)

$$i_1 = \underset{1 \leq i \leq 200}{\text{argmin}} [\frac{|C_1^A|}{|C_1|} + \frac{|C_2^N|}{|C_2|}] \tag{12}$$

where, $\frac{|C_1^A|}{|C_1|}$ is same as 1 - TPR and $\frac{|C_2^N|}{|C_2|}$ is same as FPR. Now, a threshold value for the first method, v_{th1} is when i is i_1.

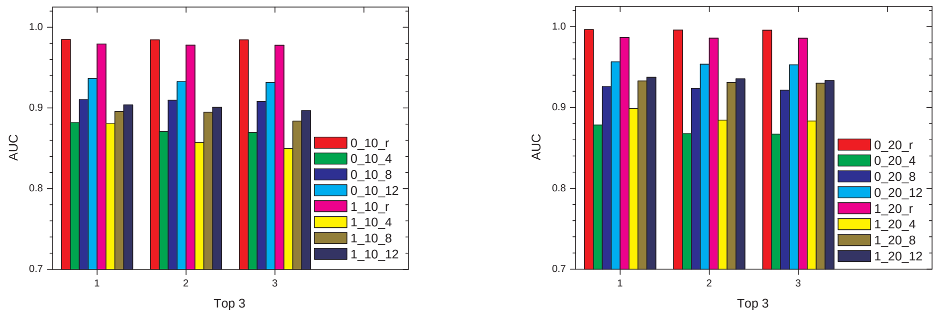
$$v_{th1} = v_{i1}$$

Second, we consider receiver operating characteristics (ROC) that is represented as FPR and TPR. Here, we try to find a threshold value whose point in the ROC is the closest one to a point, where FPR=0 and TPR=1.

$$i_2 = \underset{1 \leq i \leq 200}{\text{argmin}} [\sqrt{(0 - \frac{|C_2^N|}{|C_2|})^2 + (1 - \frac{|C_1^N|}{|C_1|})^2}] \tag{13}$$

Similarly, a threshold for the second method, v_{th2} is when i is i_2.

$$v_{th2} = v_{i2}$$



(a) length 10 episode (b) length 20 episode

Figure 3: Top 3 Area Under Curve

5. EXPERIMENTAL RESULTS

In this section, we show our experimental results. We used a data set (subject1) which is provided by [16]. We name this data set as T1 data set from now on. The first experiment is about a calculated threshold value by equation (12) and (13) that compares the results when only sequences are considered and when temporal aspects are also considered. The second experiment is to find a best combination of parameters that produces maximum area under curve (AUC) at ROC.

In our experiment, we have several parameters that determine similarity functions and all the experiment conditions. As described in section 4.2.1, we have two modes to calculate s_3 (equation (6)) and s_4 (equation (7)) in similarity function. We use symbols s_I and s_D to represent independent functions to s_1 and dependent functions to s_1, respectively. We test two different lengths of episodes, 10 and 20. Weights on similarity function, w_1, w_2, w_3, and w_4, are also considered to have various combinations such that sum of them is equal to 1 and their individual precision is tenth.

For the T1 data set, we used 70% of it for training and the rest 30% for testing purposes. The T1 data set contains only cases of normal behavior. We create synthetic cases of abnormal behavior using different methods. The first one is created by random generation. The second is synthesized from normal data set by shifting activation time by 4 hours, which follows Gaussian distribution $N(0, 1)$. The third and fourth synthesized abnormal sets are 8 hours shifted one and 12 hours shifted one, respectively.

5.1 Fixed Threshold

In this section, we basically compare the results from similarity functions that consider only sequence (s_1) and that consider temporal aspects ($s_1 \sim s_4$). We first calculate best threshold values using both method 1 (M1, equation (12)) and method 2 (M2, equation (13)). 50% of T1 data set is used for as a gold standard of normal cases. Another 20% of it is used to determine the optimal threshold value by comparing episodes that appear in this part, mixed with synthetic abnormal episodes, with episodes of the previous part. Once a threshold value has been calculated, we apply it to classify episodes of two testing sets, one with the remaining 30% of T1 data set and the remaining part of the randomly generated data set, and the other with same part of data that is used with the randomly generated data set and a part of data set, which is 4hr shifted synthetic data set. The result shows that we have improved values of both FPR and TPR for the current data sets regardless of the different combinations of a length of episodes (10/20) and threshold methods (M1/M2) (see figure 2).

Now, for considering temporal aspects, we have 84 different weight combinations of similarity function. Similarly, we calculate threshold values from M1 and M2, and apply them to current sets (normal set with randomly generated data and normal set with 4hr shifted data). But, unlike the previous experiment, we choose the best weight combination from individual case that produces the best argument in equations (12) and (13). The result is summarized in the tables 2 and 3.

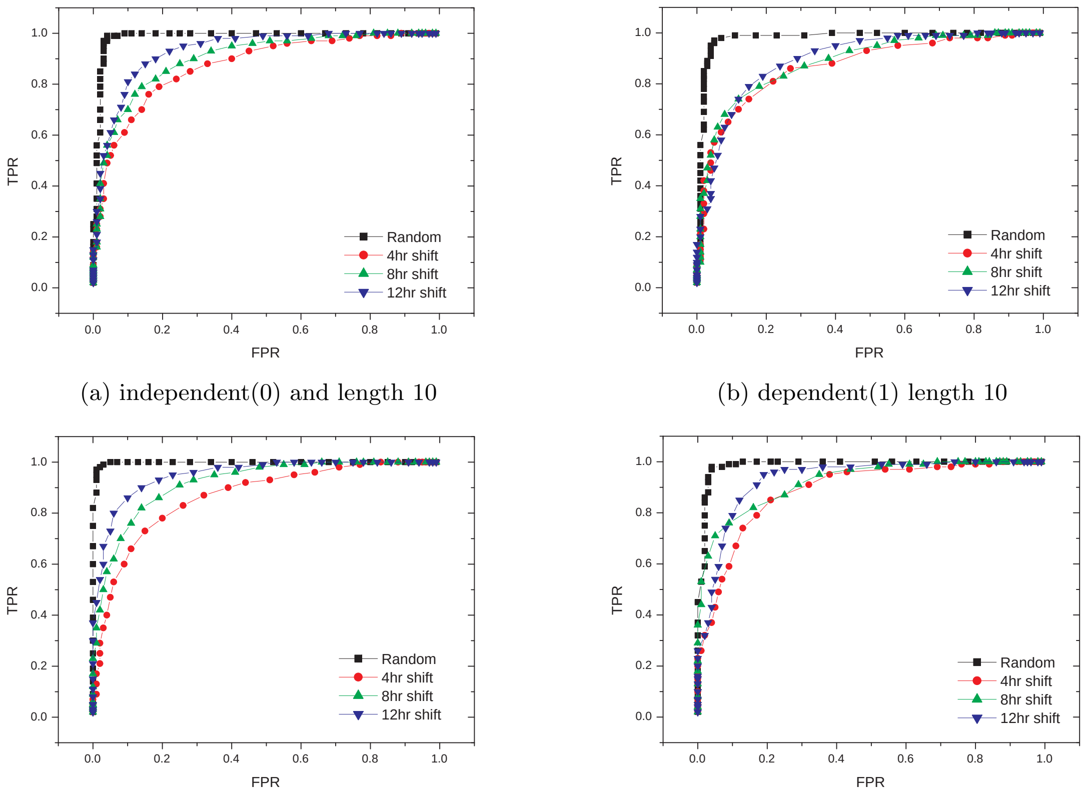
Sum of FPR and 1-TPR in M1 and distance in M2 of the random set have improved compared to ones for the training set in all the cases (see tables 2 and 3, where I/D stands for independent/dependent on the sequence of common events, EL for episode length, and cid for ID of 84 combinations of weights.) This is because larger amount of data in current set can reflect correctly the characteristics in the training set. But for the 4hr shifted data set, threshold values that are calculated for the best partition in training set do not properly separate normal and abnormal set in current data set, since all the values (sum and distance) get worse. This is because episodes that are slightly time-shifted can have high similarity score, which causes a proper separation to be difficult.

5.2 Study of AUC

Now, we give all the different threshold values[1] and try to measure area under curve (AUC) from the ROC graph to better know what combination of parameters can have the best results. We tested a combination of episode lengths of 10 and 20, and dependent/independent temporal functions (s_3 and s_4) to s_1. '0' indicates that the temporal functions are not dependent on the number of matching events in episodes and '1' indicates that the temporal functions are dependent on the number of matching events in episodes. Again, we have 84 different combinations of weights for similarity function.

In figure 3, we show the top 3 cases in terms of AUC. The average weights (w_1, w_2, w_3, and w_4) for the top 3 are 0.4, 0.12, 0.17, and 0.31 respectively for length 10 case, and 0.33, 0.16, 0.18, and 0.33 respectively for length 20 case.

[1] Finite number of threshold values, from 0.005 to 0.995 increased by 0.005



(a) independent(0) and length 10

(b) dependent(1) length 10

(c) independent(0) length 20

(d) dependent(1) length 20

Figure 4: ROC graphs for 4 cases

In both cases, the weights for the sequence matching and temporal aspect (duration) play important role to produce better results. If we average them, we get the combination of weights as 0.36, 0.14, 0.18, and 0.32. Independent temporal functions have slightly better results than dependent temporal functions in both length 10 and 20 cases. For the time shifted three different sets (4hr, 8hr, and 12hr), the maximum time shifted set (12hr) has the best result and 4hr shifted set has the worst result. This is because the bigger time shifted value can be separated more clearly into normal and abnormal sets due to the function s_3 than the smaller ones.

In figure 4, we show the best combination of weights in ROC graphs for the different 4 cases, which are combined with dependent/independent temporal functions (0/1) and length of episodes (10/20). The common fact for the 4 cases is that the result of ROC for the random set is the best, and 12hr shifted set is the next, followed by 8hr shifted set and 4hr shifted set. Since the random set has episodes that are not much similar to the existing ones, the result of ROC outperforms the sets that have time shifted episodes. When we have independent temporal functions regardless of length of episodes, the individual ROC graph is clearly separated and the order for the 4 sets are distinct.

6. CONCLUSIONS AND FUTURE WORK

In this work, we proposed a method to classify normal and abnormal behavior in an assisted living environments by applying a similarity function that considers various parameters. We also tested our proposed similarity function on the existing data set and synthetic data set. We suggested two methods to determine threshold values. These methods worked well for the random set, but they did not work for the 4hr shifted due to unclear separation point for the normal and abnormal episodes. The more time shifted (no more than 12 hours), the better result we have in terms of AUC and ROC. Since our similarity function considers temporal aspects as well as the sequence of events, abnormal behavior that has similar sequence of events but has different time stamps or duration of events can be classified into normal and abnormal classes.

7. ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work is supported in part by the National Science Foundation under award numbers CT-ISG 0716261 and MRI 0923494. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

8. REFERENCES

[1] J. M. Agosta, C. Diuk-Wasser, J. Chandrashekar, and C. Livadas. An adaptive anomaly detector for worm detection. In *SYSML'07*.

Table 2: The best results from 84 cases for threshold method 1 (M1).

Data set	I/D	EL	cid	Weights				Training Set				Current Set		
				w_1	w_2	w_3	w_4	v_{th}	FPR	1-TPR	Sum	FPR	1-TPR	Sum
Random	I	10	65	0.4	0.1	0.1	0.4	0.35	0.19	0.01	0.2	0.11	0	0.11
	D	10	78	0.5	0.2	0.1	0.2	0.36	0.16	0.11	0.27	0.05	0.04	0.09
	I	20	40	0.2	0.3	0.1	0.4	0.32	0.11	0.01	0.12	0.03	0	0.03
	D	20	80	0.5	0.3	0.1	0.1	0.29	0.17	0.03	0.2	0.08	0	0.08
4hr shifted	I	10	1	0.1	0.1	0.1	0.7	0.42	0.21	0.04	0.25	0.65	0.02	0.67
	D	10	75	0.5	0.1	0.1	0.3	0.44	0.18	0.19	0.37	0.49	0.07	0.56
	I	20	2	0.1	0.1	0.2	0.6	0.42	0.13	0.01	0.14	0.6	0.01	0.61
	D	20	75	0.5	0.1	0.1	0.3	0.41	0.11	0.27	0.38	0.32	0.09	0.41

Table 3: The best results from 84 cases for threshold method 2 (M2).

Data set	I/D	EL	cid	Weights				Training Set				Current Set
				w_1	w_2	w_3	w_4	v_{th}	FPR	TPR	distance	distance
Random	I	10	65	0.4	0.1	0.1	0.4	0.37	0.18	0.94	0.19	0.04
	D	10	78	0.5	0.2	0.1	0.2	0.36	0.16	0.89	0.19	0.06
	I	20	65	0.4	0.1	0.1	0.4	0.34	0.14	0.96	0.15	0.02
	D	20	74	0.4	0.4	0.1	0.1	0.3	0.16	0.91	0.18	0.04
4hr shifted	I	10	2	0.1	0.1	0.2	0.6	0.47	0.18	0.93	0.19	0.59
	D	10	65	0.4	0.1	0.1	0.4	0.52	0.23	0.74	0.35	0.47
	I	20	1	0.1	0.1	0.1	0.7	0.39	0.09	0.92	0.12	0.52
	D	20	75	0.5	0.1	0.1	0.3	0.41	0.11	0.73	0.29	0.33

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *VLDB'94*.

[3] M. Bezzi and R. Groenevelt. Towards understanding and modeling individual behavior and group dynamics. 2008.

[4] C.-S. Chao, Y.-X. Chen, and A.-C. Liu. Abnormal event detection for network flooding attacks. *Journal of Information Science and Engineering*, 20(6):1079–1091, 2004.

[5] P.-C. Chung and C.-D. Liu. A daily behavior enabled hidden markov model for human behavior understanding. *Pattern Recognition*, 41(5):1572–1580, May 2008.

[6] T. H. Cormen, C. E. Leiserson, and R. Rivest. *Introduction to Algorithms*. 1990.

[7] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *CVPR '05*.

[8] A. M. Faizal, M. M. Zaki, S. Shahrin, Y. Robiah, and S. S. Rahayu. Threshold verification technique for network intrusion detection system, June 2009.

[9] X. Geng, G. Li, Y. Ye, Y. Tu, and H. Dai. Abnormal behavior detection for early warning of terrorist attack. In *AI 2006: Advances in Artificial Intelligence*, 2006.

[10] P. D. Haghighi, M. M. Gaber, S. Krishnaswamy, and A. Z. S. W. Loke. Context-aware adaptive data stream mining. *Special Issue on Knowledge Discovery from Data Streams*, 2008.

[11] J. He, H. Li, and J. Tan. Real-time daily activity classification with wireless sensor networks using hidden markov model. In *EMBS '07*.

[12] S. Lühr, S. Venkatesh, G. West, and H. H. Bui. Duration abnormality detection in sequences of human activity. Technical Report TR-2004/02, Curtin University of Technology, 2004.

[13] S. Lühr, G. West, and S. Venkatesh. Recognition of emergent human behaviour in a smart home: A data mining approach. *Pervasive Mob. Comput.*, 3(2):95–116, 2007.

[14] D. Lymberopoulos, A. Bamis, and A. Savvides. Extracting spatiotemporal human activity patterns in assisted living using a home sensor network. In *PETRA '08*.

[15] K. Park, E. Becker, J. K. Vinjumur, Z. Le, and F. Makedon. Human behavioral detection and data cleaning in assisted living environment using wireless sensor networks. In *PETRA 09*, 2009.

[16] E. M. Tapia, S. S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. In *Pervasive Computing*, pages 158–175, 2004.

[17] N. Vaswani, A. R. Chowdhury, and R. Chellappa. ”shape activity”: A continuous state hmm for moving/deforming shapes with application to abnormal activity detection. *IEEE Trans. on Image Processing*, 14(10):1603–1616, 2005.