A Multidimensional Time-Series Similarity Measure With Applications to Eldercare Monitoring

Zahra Hajihashemi, *Member, IEEE*, and Mihail Popescu, *Senior Member, IEEE*

Abstract—In the last decade, data mining techniques have been applied to sensor data in a wide range of application domains, such as healthcare monitoring systems, manufacturing processes, intrusion detection, database management, and others. Many data mining techniques are based on computing the similarity between two sensor data patterns. A variety of representations and similarity measures for multiattribute time series have been proposed in the literature. In this paper, we describe a novel method for computing the similarity of two multiattribute time series based on a temporal version of Smith–Waterman (SW), a well-known bioinformatics algorithm. We then apply our method to sensor data from an eldercare application for early illness detection. Our method mitigates difficulties related to data uncertainty and aggregation that often arise when processing sensor data. The experiments take place at an aging-in-place facility, TigerPlace, located in Columbia, MO, USA. To validate our method, we used data from nonwearable sensor networks placed in TigerPlace apartments, combined with information from an electronic health record. We provide a set of experiments that investigate temporal version of SW properties, together with experiments on TigerPlace datasets. On a pilot sensor dataset from nine residents, with a total of 1902 days and around 2.1 million sensor hits of collected data, we obtained an average abnormal events prediction F-measure of 0.75.

Index Terms—Genetic algorithm, multiattribute time series, Smith–Waterman (SW) algorithm, time-series data mining.

I. INTRODUCTION

THE American Academy of Nursing, in 1996, asked researchers to propose new solutions to modify and enhance the standards of eldercare services [1]. In the developed countries, the rapid aging of the population garnered attention from healthcare providers. Published statistics show an increase in the elderly (aged 65 and older) from 13% in 2010 to 19% in 2030, while the ratio between working-age (age between 15 and 64 years) and elderly population will decline from 4.3 to 2.3 [2]. In the meantime, older adults are eager to live independently, regardless of complex chronic conditions such as frailty, dementia, and risk of falling. However, independent living may in turn lead to delayed health assessments due to lack of monitoring, which is associated with poor long-term health outcomes [3]. Late health assessment is an aggravating risk factor that

usually occurs because of the fear of institutionalization and the failure of a physician's assessments [4]. A possible solution to unreported health problems is to utilize an automatic health monitoring system that is able to detect and report signs of early illness.

In the last decades, a growing number of projects have deployed ubiquitous sensor networks to monitor health of older adults. MIT PlaceLab, Georgia Tech's aware House, Honeywell's Independent Lifestyle Assistant, and The University of Missouri's TigerPlace are such examples [5]–[8]. A variety of activity recognition (AR) methodologies that asses individuals ability to complete Activity of Daily Livings such as eating, grooming, cooking, drinking, and taking medicine have been reported in the literature [9]–[12]. Primarily differences in the reported AR methodologies are due to underlying sensing technology, machine learning models, and the realism of experimental setup [13], [14]. Regardless of these differences, the majority of reported AR techniques were performed on scripted or presegmented sequences of activities. One of the strengths of our paper is that it uses data collected in a real living environment for a relatively long period of time (1902 days). Moreover, since we do not have a given time interval for experiments, our dataset is continuously expanding.

A key feature of our health-monitoring systems is the ability to continuously and unobtrusively collect information about daily activities of older adults. The system processes the acquired sensor data, infers activities such as "bathroom visit" or "out of the apartment," and tries to detect changes in behavior of the monitored residents. Early signs of an impending illness or an exacerbation of an existing chronic condition may produce detectable behavior changes. The link between behavior changes and health patterns is based on the assumption that, for a given person, similar medical conditions result in similar (abnormal) behaviors, hence in similar sensor patterns. If a sensor pattern is not similar to previous ones observed in a similar context (time of the day, location), then we assume that it is produced by some (unknown) health condition. While these assumptions might not in general be true for a younger, more mobile population, they are in general valid in elderly [15]–[17]. To capture a person's behavior, researchers employed various sensors, such as motion, Kinect, radar, sound, etc. [8] that produced multiattribute time-series datasets (MATS). The mathematical basis of assessing similarity of behavior patterns based on sensor data is computing the similarity of multiattribute sensor time series.

There are many approaches to computing MATS similarity depending on the application and the attribute type (discrete or continuous). Euclidian-based distance functions [18] are reported for two continuous MATS sequences of equal lengths,

whereas algorithms based on dynamic time warping (DTW) or longest common subsequence (LCSS) are applied on length mismatched sequences [19]–[23]. Symbolic aggregate approximation, SAX, is a type of MATS representation that allows dimensionality reduction and lower bounds distance measures, based on DTW, on the original time series [24]. For discrete sequences (symbolic data), such as protein sequences in bioinformatics, similarity measures such as Smith–Waterman (SW) [25] have been proposed. Measuring the similarity of mixed sequences (some dimensions are continuous, such as heart rate values, and others are discrete, such as motion detector hits) in applications such as eldercare monitoring systems is a challenging task. Although similarity measures such as LCSS [26] or SW are originally designed for discrete data, they can be applied to continuous data by either modifying the original algorithm [19] or converting the sequence from discrete to continuous format [16], [23], [27], [28].

There are two factors that distinguish our work from other related works: the dataset and the machine learning model. *Dataset*: our dataset consists of real world living environments where there is no prescripted scenario. Elderly residents live in their personal environment and conduct their daily routines with no instructions from researchers. Our dataset reflects the complexity of unconstrained real-world and it allows for testing the robustness of our AR approach. *Machine Learning Model:* we describe a similarity measure for discrete time series and its applications to eldercare. Our approach does not require series data conversion to continuous format. Instead, we employ the natural arrangement of all different sensor hits together with their time stamps into a one-dimensional sequence. We briefly introduced the idea of a modified SW framework for time stamped discrete sequences, Temporal Smith–Waterman (TSW), in [29] and performed some limited testing for three TigerPlace residents. In this paper, we extend our previous work by providing a scalable TSW version, showing other possible applications and testing TSW on larger datasets.

Our paper is structured as follows. In Section II, we describe our system architecture and dataset. In Section III, we explain our method and its application to AR in eldercare. In Section IV, we provide extensive experiments to investigate our approach. Finally, in Sections V and VI, we discuss the implications of our approach and conclusion, respectively.

II. SYSTEM ARCHITECTURE

We implemented our integrated monitoring system in TigerPlace, an aging-in-place facility from Columbia, MO, USA, with the University of Missouri IRB approval. We decided to use only nonwearable sensors for the monitoring purposes after conduction multiple focus groups in 2004 with elderly residents of TigerPlace [8], [30], [31]. The monitoring began in the fall of 2005. Presently, we have 50 apartments online. Fig. 1 shows the architecture of our data acquisition system.

Various motion sensors have been deployed in different parts of the TigerPlace apartments: bedroom, kitchen, living room, and bathroom. For activity monitoring, we used wireless passive infrared (PIR) and pneumatic bed sensors networked us-
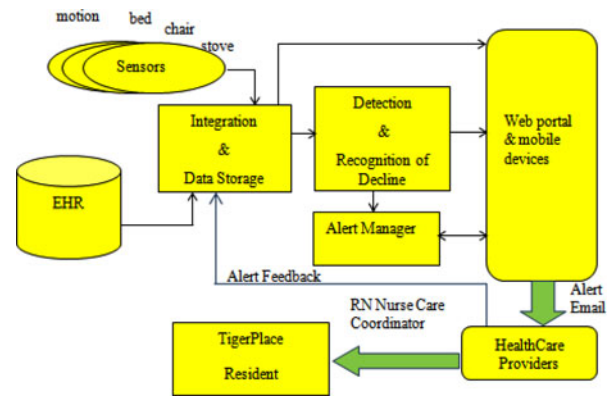

Fig. 1.   TigerPlace sensor network architecture.

TABLE I
SENSOR IDENTIFIERS USED IN THIS PAPER

ID	Type: Sensors	8. Pulse1 (low)*	16. Stove TempHigh**
1. BedMovement1*	9. Pulse2 (ok)*	17. Stove TempLow**
2. BedMovement2*	10. Pulse3 (high)*	18. Drawer**
3. BedMovement3*	11. BedroomMotion**	19. Cabinet**
4. BedMovement4*	12. Living Motion**	20. Cup Cabinet**
5. Breathing1 (low)*	13. Bath Room**	21. Refrigerator**
6. Breathing2 (ok)*	14. Off Chair**	22. Plate Cabinet**
7. Breathing3 (high)*	15. On Chair**	23. Silverware Drawer**

**indicates X10 sensors and *indicates bed sensors.

ing the X10 protocol. Each sensor sends an X10 signal that is logged with a time stamp in our sensor database. When triggered, the X10 motion sensor sends an ON signal and waits for at least seven seconds before motion will trigger the sensor to send another ON signal. After 60 seconds of no motion after an ON signal, the sensor sends an OFF signal. Table I presents the sensors and their identifiers used in this paper. The bed sensor is able to measure bed movements (ID 1–4), breathing (ID 5–7), and pulse (ID 8–10). The bed sensor differs from the X10 motion sensors in which they do not check for motion using PIR sensing but instead use a pneumatic pressure strip to monitor various levels of three kinds of signals [32]. The pneumatic strip captures vibrations from movement, breathing, and the resident's ballistocardiogram, which is the mechanical effect of the heartbeat. The discretization of the bed sensor is performed in hardware by the bed sensor system as described in [11]. We are currently in the process of deploying a new bed sensor [9] that provides continuous signals for all three previously mentioned quantities without the artificial quantization.

Table II shows a small sequence (of length 8) of sensor firing data recorded in the log file of the system for the resident #3 (the person has probably got out of bed and went to his living room). Table III describes the pilot sensor data used in this study with a total of 1902 sensor days and 2.1 million sensor hits, obtained from nine residents of TigerPlace. For each resident, we also retrieved visit notes about various physical and/or mental health complaints added by the nurses on site to the resident's EHR. In our dataset, there are fewer notes than sensor data (automatically logged for each resident), as for some days, there were

TABLE II
SENSOR SEQUENCE SNIPPET FOR RESIDENT#3

ResidentID	SensorID	Year	Month	Day	Time
3	3	2005	10	5	12:34:38
3	2	2005	10	5	12:36:52
3	2	2005	10	5	12:37:04
3	2	2005	10	5	12:37:11
3	1	2005	10	5	12:37:26
3	1	2005	10	5	12:37:28
3	12	2005	10	5	12:38:32
3	12	2005	10	5	12:41:18

TABLE III
PILOT DATASET

Resident Code	Number of days	Abnormal days
1	440	83
2	463	44
3	499	335
4	225	130
5	81	60
6	72	55
7	45	15
8	39	10
9	38	15

no nursing comments. However, some residents (e.g., resident #3) may have multiple comments per day. In addition, each day was manually coded as normal or abnormal for each individual resident. Multiple clinical experts retrospectively reviewed the TigerPlace EHR data (nursing comments, vital signs, etc.) to flag days (denoted as "abnormal") with resident illness episodes (column 3 in Table III), but no interrater agreement was computed [33]. We use the abnormal days to test our early illness recognition (EIR) algorithm.

III. METHOD

In this section, we describe the TSW similarity measure and its application.

A. TSW Similarity

We define a time series S as a set of n couples $(s_i t_i)$, i.e., $S = \{(s_1 t_1), \ldots, (s_n t_n)\}$. Each couple $(s_i t_i)$, $1 \leq i \leq n$, has two components: a sensor signal s_i that belongs to a symbol set Σ and a time stamp t_i that represents the time when s_i was recorded in the database. The alphabet Σ is a set of identifiers we use to represent multidimensional sensor time series. In our case, Σ comprises of the symbols shown in Table I.

While some sensors such as motion detectors can be naturally described by symbolic data, others, such as bed sensor, may need to be quantized. That was the case of our bed motion sensor for which the bed motion was empirically divided [11] into four categories: less than 3 s, 3–7 s, 7–14 s, more than 14 s (sensor ID 1–4 in Table I).

Given two discrete time series $S_1 = \{(s_{11} t_{11}), \ldots, (s_{1n} t_{1n})\}$ of length n and $S_2 = \{(s_{21} t_{21}), \ldots, (s_{2m} t_{2m})\}$ of length m, with $s_{1i}, s_{2j} \in \Sigma$, $1 \leq i \leq n$, $1 \leq j \leq m$, we can

compute their TSW similarity, $\text{TSW}(S_1, S_2)$, using the following algorithm [29]:

$$H_{i0} = H_{0j}, i \in [1, n] \text{ and } j \in [1, m] \tag{1}$$

$$H_{ij} =$$
$$\max\{0, H_{i-1,j-1} + Sim(s_{1i}, s_{2j}), \max_{k \geq 1}\{H_{i-k,j} - W_{\Delta t}),$$
$$\max_{k \geq 1}\{H_{i,j-k} - W_{\Delta t}\}\} \tag{2}$$

$$W_{\Delta t} = g + c|t_{1i} - t_{2j}| \tag{3}$$

$$\text{TSW}(S_1, S_2) = \max\{H_{ij}\} / \min\{n, m\} \tag{4}$$

where H is a work matrix used in the dynamic programming to trace back the best alignment between S_1 and S_2 and Sim is a symbol similarity matrix that reflects the compatibility between symbols. For example, $Sim(\text{BedMovement1}, \text{BedMovement2}) = 0.9$, since they are both bed motion sensor firings (see Table I), whereas $Sim(\text{BedMovement1}, \text{Cabinet}) = 0$ since they belong to different type of sensors. The g constant is a penalty for opening a gap, while c is a penalty for extending a gap. The gap open penalty is the score taken away when a gap is created. The best value depends on the dataset. We set $g = 0$ to limit our search space. This would not have any effect on the performance of TSW on TigerPlace dataset.

Similar to the traditional SW used in bioinformatics [25], TSW considers the time difference between two sensor firings as a gap and, consequently, computes a gap penalty $W_{\Delta t}$ using the time stamp associated to each symbol (3). The time is given in seconds and the constant c used in (3) controls the time scale associated with the symbol firing. While the TSW algorithm can be used for any symbolic time series, the time scale used in (3) is application dependent. For example, in computing the similarity between patients represented by sequences of International Classification of Disease version 9 (ICD-9) diagnoses [34], the time scale would probably be of the order of months where in our application, the time scale is of the order of minutes. The exact computation of c can be performed if a training set of known sequence similarities is available, as we will show in Section IV-A.

B. Sequence Search Using TSW

Our main motivation for developing a sensor sequence similarity measure is to be able to compare human behaviors and activities. One simple approach [29] is to divide the entire sensor sequence acquired from a TigerPlace apartment into fixed intervals such as days (sets of 24-h long sensor sequences) and use TSW to compute the similarity between them. However, since a given activity can be performed at different times of the day, week, or year, we should be able to search the entire database for a given pattern. To address this question, in this paper we employ a window-based TSW (WTSW) method. WTSW algorithm uses a sliding window approach to find the similar subsequences to a user defined (query) subsequence based on TSW similarity measure. In our case, examples of user defined subsequences are bathroom visits or food preparation activities. Suppose we denote the sequence of past sensor firings by D and the user
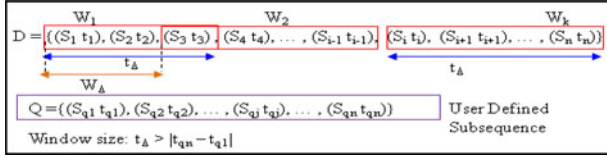
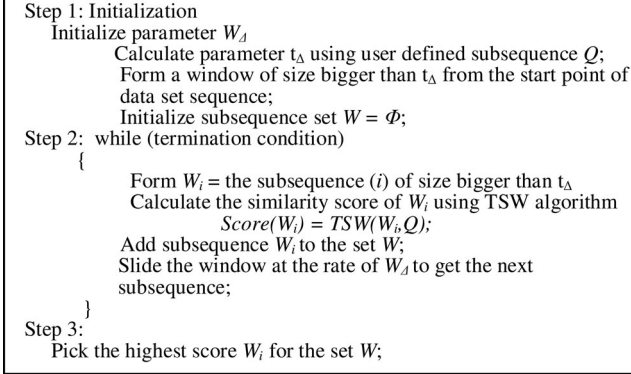Fig. 2.   Finding the most similar subsequence using WTSW method.

Fig. 3.   Pseudocode of WTSW method.

defined subsequence (query) by Q. To find the most similar subsequence to Q in D, we slide a window of size t_Δ over D. Fig. 2 shows this process.

The size of a window, t_Δ, must be greater than the difference of the times when S_{q1} and S_{qn} were observed in Q, that is $t_\Delta > (t_{qn} - t_{q1})$. Consecutive windows overlap by an interval of time $(t_\Delta - W_\Delta)$. To exclude trivial matches (subsequences part of Q), we do not consider subsequences with time stamps that overlap with Q. After extracting all of nontrivial subsequences, we assign a similarity score to all of subsequences using TSW similarity measure. We then pick the subsequence with the highest score as the most similar subsequence to the given user-defined subsequence Q. The pseudocode of the WTSW algorithm is represented in Fig. 3. The WTSW faces two challenges. The first challenge is speed. The TSW algorithm is fairly time intensive. The second challenge is initializing parameter W_Δ. If W_Δ is small, the results are more accurate, but the computing time is high. The running time can be decreased by reducing the overlap; however, some relevant subsequences might not be retrieved.

The TSW similarity of two subsequences of lengths m and n has $O(mn)$ complexity. If length of D is r, then we might need between r/n (no overlap) and $r - n$ ($n - 1$ overlap) TSW evaluations. If $r >> n$, then the upper bound of complexity is rn^2, which may be high for large r. For example, "bathroom visit" behavior has typically 200 symbols ($n = 200$) and the five years past behavior sequences may be 2 million symbol long ($r = 2\,000\,000$). To address these two challenges, we employ the genetic algorithm approach described in the following section.

C. Speeding of WTSW Using a Genetic Algorithm Approach

Similarity search is a subtask of most of time-series data mining algorithms. The time complexity of search algorithm

has always been the bottleneck for large time series. Depending on the properties of the distance (similarity) measure used in the search, different techniques have been proposed in the literature to speed up searching process [20]–[23], [35]. In this paper, we use genetic algorithm as a first try to speed up WTSW and leave other possible speeding up methods for future work. We acknowledge that using genetic algorithm to speed up the searching process is not the best solution. We applied genetic algorithm as a solution that mimics the sliding window feature of WTSW method and produces satisfactory results in a reasonable time. Lower boundary searches and early abandoning [23] may speed up TSW. For computing the similarity of two subsequences, early abandoning keeps track of the minimal distance found so far when computing the distance of two new elements of the subsequences. If it approximates that the current minimum will be exceeded, it terminates. Although the early abandoning speeds up computing the similarity of two subsequences, it still needs to perform all pairwise comparisons. Lower bounding techniques, on the other hand, perform well on uniform scaling distances. For larger values of n, however, the algorithm still remains intractable (n is the length of dataset time series—D in Fig. 2—that for TigerPlace dataset represents two years of sensor sequences).

Note that the focus of this paper is to study the effectiveness of TSW measure on collected data from TigerPlace and its practical utility in resolving real-world problem of eldercare such as detection of daily routines and abnormal events detection. In what follows, we define main GA parameters.

Gene: Given a subsequence S of length m, where $S = \{(s_1 t_1), \ldots, (s_m t_m)\}$, a gene is a pair (s_i, t_i), $1 \leq i \leq m$, that identifies the sensor s_i that fired at time t_i. That is, a gene in GATSW is a window representing a sensor firing and its time stamp.

Chromosome: A chromosome is a sequence of sensor firings within a specified time interval, namely a set of genes. Given a time series T of length n, a subsequence S_i of T is an ordered continues sampling of length m, $S_i = \{(s_i t_i), \ldots, (s_{i+m-1} t_{i+m-1})\}$, where $m \leq n$, $1 \leq i \leq n - m + 1$.

Population: A population of size p is a set of p different chromosomes.

In any genetic algorithm, we use a fitness function to determine which chromosomes survive from one generation (iteration) to another.

Fitness function: In GATSW, the fitness of a chromosome S_i is calculated by assessing the similarity of the chromosome to the given user defined subsequence Q. The *fitness* of chromosome S_i is $Fitness(S_i) = \text{TSW}(S_i, Q)$.

Mutation: Given a chromosome S_j, a mutation over S_j changes the start position based upon its fitness

$$S_j^a\,(position) \;=\; [S_j^{a-1}\,(position) + (mutation_{rate})$$

$$\times(|Fitness(S_j^{a-1}) - threshold|).\quad (5)$$

In (5), $S_j^a\,(position)$ is the position of chromosome S_j in the iteration a, and $mutation_{rate}$ and *threshold* are parameters of mutation process. $Fitness(S_j^{a-1})$ is the fitness of chromosome

Step 1: Initialization

    I. Initialize the size of population and generation, and mutation rate;

    II. Generate initial population;

Step 2: While (termination condition){

    I. Evaluate the fitness of individuals using TSW;

    II. Select parent chromosomes;

    III. Mutate parent chromosome;

    VI. Select new population;

    }

Step 3: Pick the fittest chromosome as the final solution;

Fig. 4. GATSW algorithm.

Algorithm: Abnormal events detection using TSW similarity distribution.
Input: *NS*: a set of normal sensor sequences of size *n*; S_x: an unknown sensor sequence.
Output: Label of S_x as normal or abnormal sequence;
Steps:
1. Using all sequences in set $NS=\{s_i\}_{i=1,n}$ build the similarity matrix $S=\{s_{ij}\}_{i,j=1,n}$ by calculating their pair-wise similarities using TSW method.
2. Assume a distribution function that best suites data points in matrix S.
3. For the sequence S_x, computing its similarities to all normal sequences $S_1,...S_n$, $\{s_{ix}\}_{i=1,n}$, and find the maximum similarity of $\{s_{ix}\}$, $s_{x,max}$.
4. Calculate the confidence of the similarity, $C(S_x)$, using the similarity distribution function in step 2.
5. If $C(S_x)<\delta$ then classify the unknown sensor sequence as an abnormal event, otherwise as a normal event.

Fig. 5. Abnormal events detection using TSW similarity distribution.

S_j in the previous iteration $(a-1)$. The amount of mutation depends on the fitness of the parent chromosome. The fittest parent chromosome changes its position slightly, whereas the parent chromosome with the lower fitness changes its position drastically. Fig. 4 represents the pseudocode of GATSW.

The parameter *threshold* demonstrates the best similarity score that is equal to the fitness of a perfect match. We use this parameter in mutation to control the rate of selection of children chromosomes for the next generation. The parameter $mutation_{rate}$ controls the influence of its fitness on the chromosome's position in the next generation. The more dissimilar is a chromosome from the user defined subsequence, the higher the mutation.

D. Mathematical Properties of TSW

TSW is a nonnegative function that measures the similarity of two temporal sensor sequences s_1 and s_2 over a finite alphabet Σ. If we denote by DTSW the distance-induced TSW, DTSW = 1 – TSW, then DTSW is a metric since 1) $\mathrm{DTSW}(s_1, s_2) = 0$ if and only if $s_1 = s_2$; 2) DTSW is symmetrical, that is $\mathrm{DTSW}(s_1, s_2) = \mathrm{DTSW}(s_2, s_1)$; and 3) DTSW satisfies the triangular inequality:

$$\mathrm{DTSW}(s_1, s_3) \leq \mathrm{DTSW}(s_1, s_2) + \mathrm{DTSW}(s_2, s_3).$$

We refer readers to [36] for formal proofs of these properties. These properties of TSW will be useful in our clustering experiments presented later in this paper.

E. Health Pattern Prediction: TSW Applications

An important goal of our TigerPlace research is to predict the change in resident's health status based on sensor data produced by in-home monitoring system. The capability of sensor networks for predicting health patterns using logistic regressions, multiple instance learning, and temporal clustering has been previously investigated on aggregated sensor data [37], [38]. In [14] and [39], the authors proposed new methods to use sensor network data for tracking changes in daily routines that can be used in cognitive health assessment. Multiple features such as aggregation of sensor counts and temporal features have been extracted from sensor sequences to quantify performed activities. An SVM learner as well as a search algorithm has been applied to detect activities and track changes. Linear regression and Gaussian process has been used in [40] to map the extracted features from aggregated sensor count to the functional ability of residents. Using the same data aggregation approach, in [41], the authors proposed a statistical method to estimate the average amount of time a resident spent in each room within his apartment. Our proposed approach is different in a way that we do not aggregate sensor counts. In this paper, we explore the capacity of the TSW method to detect health patterns.

1) Detection of Abnormal Activities: In elderly population, decline of functional ability happens by slight changes in normal activities of elderly. We apply TSW measure on sensor sequences to track changes in sensor patterns and detect abnormal events. Abnormal events are defined by unusual sensor activity patterns that require a nurse's assessment of the resident. TSW measure is used to classify sensor sequences as "normal" or "abnormal" by employing the distribution of similar sensor sequences. We claim that if a given sensor sequence is not similar enough to previously seen normal sensor sequences, it is an indication of a possible abnormal event [42]. We use the TSW measure, since we would like to find similar behaviors across different days that happen at roughly the same time [42].

We propose our abnormal activity detection method as follows: Given the set of *n* normal sequences $NS = \{s_i\}_{i=1,n}$, the pairwise similarity matrix $S = \{s_{ij}\}_{i,j=1,n}$ is calculated where $s_{ij} = \mathrm{TSW}(s_i, s_j)$. A distribution function, *f(S)*, that suites best the $\{s_{ij}\}$ similarities of these "normal" days is chosen. To classify an unknown sequence S_x, first, its similarities with all normal sequences $s_1, \ldots, s_n$ is computed resulting similarities $\{s_{ix}\}_{i=1,n}$. Then, the maximum similarity value $s_{x,\max} = \max(\{s_{ix}\}_{i=1,n})$ is obtained. The confidence that S_x is abnormal, $C(S_x)$, is 0 if $s_{x,\max} > \mathrm{mean}(f(S))$ and equal to *1 – P*, where *P* is the likelihood that $s_{x,\max}$ comes from *f(S)*. Fig. 5 shows the algorithm of abnormal event detection using TSW similarity distribution.

2) Detection of Daily Bathroom Visits: TSW measure is applied to develop a novel approach to identify deviations in patterns of day-to-day activities of older adults to generate alerts to the healthcare providers for timely interventions [43]. Daily routines, such as bathroom visits, can be monitored by sensor

```
Algorithm: Hierarchical Clustering using TSW
Input: N sensor sequences, and an N*N similarity matrix;
Output: a dendrogram representing the hierarchical structure;
Steps:
    1. Consider each sensor sequence as its own cluster.
    2. Find the most similar pair of sensor sequences using TSW and
       merge them into a single cluster.
    3. Compute distances between clusters as follows D(Cᵢ,Cⱼ)=max[1-
       TSW(Sᵢₖ,Sⱼₖ·)]ₖ₌₁,ₙ,ₖ'₌₁,ₘ where cluster i has n and cluster j has m
       sensor sequences.
    4. Repeat steps 2 and 3 until all sensor sequences are clustered into a
       single cluster of size N.
```

Fig. 6.   Hierarchical clustering using TSW algorithm.

TABLE IV
PARAMETER SETTINGS OF WTSW AND GATSW METHODS

WTSW Method	W_Δ (min)	t_Δ (hour)	Time (sec)
	20	2	14
GATSW Method	Population Size	Generation Size	Time (sec)
	50–60	50–60	7

TABLE V
PERFORMANCE COMPARISONS ON WTSW AND GATSW METHODS

WTSW Method	Precision	Recall	F-measure
	0.66	0.6	0.66
GATSW Method	0.77	0.71	0.75

systems. We focus on bathroom visits, as an important daily routine. A hierarchical clustering approach using TSW is applied to cluster bathroom visits and extract frequent bathroom patterns. Fig. 6 demonstrates the clustering process using TSW. We refer readers to [43] for more details on this method.

F. Evaluation Metrics and Experiment Setup

Our goal is to detect events (days) that a resident needs to be assessed (referred to as an abnormal day). Abnormal day is a subjective matter that differs for each elderly resident. That means an event that is considered abnormal for a resident might be a normal event for another resident. For example, some residents have more sound sleeps during the night, for these residents multiple bathroom visits during mid night is considered as an abnormal event, whereas for another group of residents, several midnight bathroom visits are considered normal. The labeling of normal/abnormal events depends on the functional ability of residents, their health status, taken medications, etc. We report the performance of TSW in EIR using several approaches in Section IV. We use following evaluation metrics in our experiments

$$Precision = \frac{tp}{tp + fp} \quad (6)$$

$$Recall = \frac{tp}{tp + fn} \quad (7)$$

$$F - measure = 2 * \frac{Precision * Recall}{Precision + Recall}. \quad (8)$$

Here, *tp* is the number of true positives, *tn* is the number of true negatives, *fp* is the number of false positives, and *fn* is the number of false negatives.

IV. RESULTS

In this section, we demonstrate the performances of proposed methods on TigerPlace dataset.

A. Abnormal Events Detection

Our goal is to build an EIR system for TigerPlace residents using sensor pattern similarity. For a given resident with available historical sensor data, the EIR system searches for similar patterns to the current recorded activity. The high similarity of the current sensor pattern to one in the past that was related to an abnormal health event indicates the possibility that the previous illness reoccurred. The health pattern prediction experiments are performed using a *k*-nearest neighbor (kNN) and a leave-one-out cross-validation approach. For each unknown sensor sequence S_i, we compute the distances (using WTSW and GATSW for comparison) to all other past sequences. Then, we select k ($k \sim \sqrt{n}$, n is the number of samples in the training dataset for each resident) most similar sequences together with their "normal" or "abnormal" labels. Finally, the classifier predicts the label of S_i based on the label of its k nearest neighbors using the following heuristic: if any of the labels of the k retrieved sequences is "abnormal," then we label S_i as abnormal; if all k labels are normal, then S_i is labeled as normal. This heuristic was motivated by the fact that in medical applications the cost of a missed detection far outweighs the one of a false alarm.

Here, for each resident, we analyze the performance of WTSW and GATSW in health pattern recognition in terms of retrieving "normal" and "abnormal" days. Table IV shows the parameter setting, and Table V compares the average performance of WTSW and GATSW on all residents of TigerPlace dataset. In this experiment, different parameter settings have been tested and the setting that provides the best performance has been reported in Table IV. Column "Time" presents the maximum execution time over all residents' data for each method in seconds.

Generally, by increasing t_Δ or W_Δ the *F-measure* decreases. The reason is that in a larger time interval, it is less likely that residents do the exact same thing at the exact same time of every day. Moreover, larger W_Δ results in fewer window candidates and decreases the performance of prediction. In GATSW experiments with 60 chromosomes and after 60 generation, it achieves 0.75 of F-measure in almost half a time that WTSW takes. Even though our dataset has about two millions of sensor hits and stretched over three years, this experiment shows that for larger datasets (billions of hits), GATSW can still be time demanding. This can happen if the search will go, for example, across multiple residents with similar health conditions.

We acknowledge that imbalance dataset might decrease the performance of classifier in some cases. We avoided the normal strategies of over/under sampling for the following reasons. In under sampling, the majority class would decrease *F-measure* in some cases where we have only few abnormal events; hence, the

TABLE VI
COMPARISON ON ABNORMAL AR

Method	TSW with Gamma distribution	TSW with Gaussian distribution	TSW with k-NN	Aggregated data
AUC	0.79	0.61	0.67	0.53
F-measure	0.80	0.65	0.73	0.57

classifier would be trained with few overall samples. Oversampling of the minority class is difficult, since we do not understand the problem well enough to oversample (not sure how to modify/generate abnormal events). To address the problem, we used instead a one class classifier (see Section IV-B1) that uses only the majority class to identify abnormal events.

B. TSW Applications

The most important advantage of a smart home in eldercare is the ability to track changes in resident's activities and identify potential health changes. In this section, we present the results of two different approaches to detect abnormal events using TSW method.

1) Detection of Abnormal Activities as a Sign of Illness: Most health pattern prediction techniques detect abnormal events by representing sensor data as vectors in some high dimensional space R^p (p would be 23 in our case, see Table I). Then, methods such as one class classifiers [38] or fuzzy rules [44] are used to detect abnormal events. However, since in our case we do not directly represent the sequences (activities) but compute their pairwise similarity, we need to develop another method for detecting abnormal events. The idea is similar to a one class classifier, but instead of finding a model of the normal events in R^p, we find one in the space of pairwise similarities.

Given n (training) normal sensor sequences $\{ns_i\}_{i=1,n}$, we applied the algorithm described in Fig. 5 to calculate the pairwise similarity matrix. We provide the results of our experiment using two similarity distribution functions including a *Gaussian* distribution with parameters μ and σ, and a *Gamma* distribution with parameters a and b, i.e. $\Gamma(a,b)$. Equation (9) computes the likelihood that $s_{x,\max}$ comes from the *Gamma distribution*, $\Gamma(a,b)$

$$P(s_{x,\max}, a, b) = \int_0^{s_{x,\max}} \left(\frac{1}{\Gamma(a)b} \left(\frac{t}{b} \right)^{a-1} e^{-\frac{t}{b}} \right) dt. \quad (9)$$

We performed classification experiments using a leave-one-out approach. We compare our results to the ones obtained using a k-NN approach with $k \approx \sqrt{n}$, where n is the number of normal days in the training dataset for each resident, and using the same TSW algorithm. Table VI shows the average F-measure and the area under ROC curve (AUC) on the TigerPlace dataset. Table VI also compares a data aggregation approach such that for each activity the aggregated sensor hits in each hour (1–24) is calculated from the motion sensor log files. Each day is represented by a 4×24 matrix (each row for an activity and each column for aggregated motion hits in an hour). Activities are
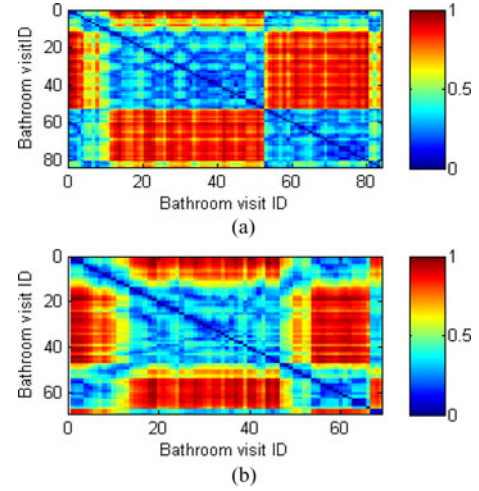

Fig. 7. Tracking changes in bathroom visits of resident #1 (a) Day1-Day 10, 84 visits (b) Day11-Day 20, 68 visits.

included bedroom, bathroom, kitchen, and living room motion sensors. A window of two weeks is considered to calculate the parameters of a single dimensional Gaussian distribution for each activity. Readers are referred to [44] for more details. TSW with distribution out performs other methods. The reason is that TSW similarity measure more accurately distinguishes between similar activities, whereas single dimensional Gaussian loses the accuracy of detection of abnormal events by aggregating sensor hits.

While our method has a 20% false alarm rate, we don't envision using this method alone. We believe that the best approach is to use multiple methods together with a fusion methodology to reduce the false alarms. The detection methods have to be as different as possible such that they do not make the same mistakes. From this point of view, we believe that the proposed methodology is different enough from the traditional one-class classifiers such that it will be beneficial in a fusion approach.

2) Detection of Changes in Daily Bathroom Visits: Frequent AR is an important step toward monitoring the health of smart home residents. If frequent activities are suddenly not performed anymore, it might be another early sign of illness. To assess TSW's ability to detect frequent activities, we manually segmented bathroom visits of two residents over two consecutive periods of ten days (20 days in total). Then, for each period, we computed pairwise TSW similarity and use the visual assessment of cluster tendency (VAT, [45]) on DTSW = 1 − TSW to find repeating bathroom patterns and their changes using (1 − TSW) as a distance measure. Fig. 7 shows the plots comparisons for resident #1.

The color map visually represents the groups of bathroom visits for each individual resident. Fig. 7(a) shows resident #1 has at least two large and distinct clusters (diagonal blocks) in the first period (Day 1–10), whereas in plot (b), these clusters are vanishing and changing in the second period (Day 11–20), because this resident got flue in the second period and the medication affected his bathroom visits. This resident uses walker which force him to be more mindful of his needs. Fig. 8 shows the same experiment for resident #2.
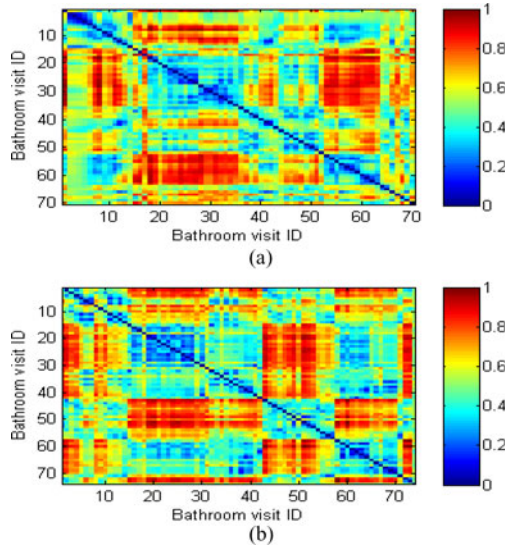
Fig. 8.　Tracking changes in bathroom visits of resident #2 (a) Day1-Day 10, 75 visits (a) Day11-Day 20, 70 visits.

Fig. 9.　Effect of the parameter c on Synthetic dataset (a) Parameter c versus correlation (b) C=15 seconds.

Resident #2 is the youngest resident who is healthier. His bathroom visits have more variations, thus more clusters. However, a change in clusters can be observed in two periods in plots (a) and (b) that is because of mid night bathroom visits. In the first period, plot (a), he has sleep problems and more mid night bathroom visits; whereas, in the second plot, plot (b), he is taking sleep medication which results in more sound sleep and fewer mid night bathroom visits. More details about this analysis can be found in [43].

V. DISCUSSION

Our approaches may help caregivers and clinicians to identify older adults who are at the greatest risk for decline and adverse events. We aim to provide computational algorithm for automated in-home monitoring system to characterize health trajectories of older adults and alert healthcare providers to implement timely interventions. Here, we provide a detailed discussion on the properties of TSW followed by presenting the results of the performance of the proposed approach on a benchmark dataset [46].

A. TSW Properties

To investigate TSW properties, we conducted two experiments: one in which sequences have the same symbols but variable time stamps and another in which the sequences are different but equidistant in time.

1) Influence of the Time Between Symbols on TSW: To study the influence of the event time distribution, we used two different synthetic sequences of length 20, S_1 and S_2, to generate ten patterns by randomly changing the time stamp of the symbols (and keeping the symbol order unchanged). Each pattern is 5 min long. The resulting 20 sequence set is denoted as $S_{\Delta t}$. The time stamp of each symbol was generated using a uniform distribution. Since we have 20 symbols distributed within 300 s, we expect an average time interval of about
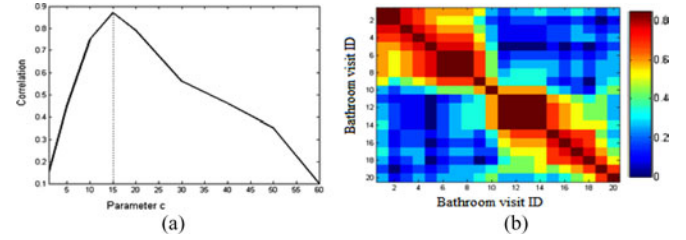
15 seconds between events. We then computed the pairwise TSW similarity, M_{TSW}, between all $S_{\Delta t}[i, j]$ for all $i, j \in [1, 20]$.

As the first half of M_{TSW} was built based on S_1 and the second half based on S_2, we expect to see a clear separation between the two groups. To quantify the separation, we correlated M_{TSW} to the ideal similarity matrix IM_{TSW} that has two all 1 blocks of size 10 on the main diagonal and 0 in the rest. Fig. 9(a) shows the value of the Pearson correlation, $corr(M_{\text{TSW}}, IM_{\text{TSW}})$, for various values of parameter c in TSW.

The best correlation (hence the best cluster separation) is achieved when c is close to the average time interval between the symbols, that is $c = 15$ s.

Fig. 9(b) represents the M_{TSW} obtained for $c = 15$ s where we can observe a clear separation between the first ten and last ten sequences. This experiment provides a method for computing the constant c in (3) for other sequence types.

2) Influence of the Rate of Symbol Change on TSW: In this experiment, we used a synthetic dataset of 20 different sequences with equidistant time but different symbols to get a sense of the TSW sensitivity to symbol change (mutation). In a similar fashion as above, we generated ten sets of 20 sequences (length 20) starting from two different sequences, S_1 and S_2, by changing 1 to 10 symbols at a time (mutation rates between $1/20 = 5\%$ to $10/20 = 50\%$).

We also used the Pearson correlation between the pairwise similarity matrix, M_{TSW}, and the ideal one IM_{TSW} to assess the cluster separation. Fig. 10(a) shows the value of the correlation for various mutation values. Fig. 10(b) depicts the similarity matrices M_{TSW} for mutation values 2 ($2/20 = 10\%$ mutation rate) and 5 ($5/20 = 25\%$ mutation rate). We see that even at 25% symbol difference [Fig. 10(b) right side plot], the similarity between two sequences is significant (about 0.6). We mention that in biological sequence, the similarity between sequences with the same function may be as low as 40%.

B. TSW Evaluation on a Benchmark Dataset

For further TSW evaluation, we applied our proposed methodology to the localization data for posture recognition dataset [46]. The dataset contains the motion sensor hits of five people performing 11 activities such as walking, sitting, falling, lying down, standing up from sitting on the ground, lying, sitting down, standing up from sitting, standing up from lying, on all fours, and sitting on the ground. Each person performs these activities in five different episodes while wearing four motion
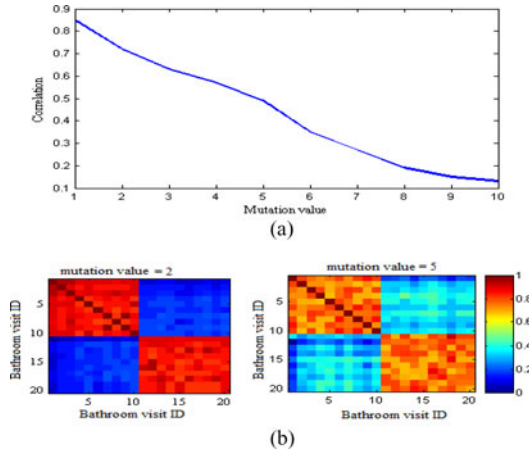
Fig. 10. Effect of mutation value on similarities in synthetic dataset (a) Mutation value versus correlation (b) Cluster seperation with different mutation values.

TABLE VII
CLASSIFICATION PERFORMANCE ON BENCHMARK DATASET

Method	GATSW with k-NN	C4.5	SVM
F-measure	0.82	0.69	0.71

sensors on the left ankle, right ankle, chest, and belt. In average, each episode lasts 3 min, and between two consecutive episodes there is a roughly 3-min pause. The dataset contains a total of 164 860 sensor hits, and the stamp time is given in milliseconds. The average time difference between two consecutive sensor hits is 32 ms. We run a set of experiments to find the best value for the parameter c in (3). We observe that the sensitivity of the posture recognition has a sharp maximum around $c = 0.0003$ which means that the relevant time is 2750 milliseconds $\sim$2.7 s.

For posture recognition, we use a k-NN classifier ($k = 7$) using GATSW as a similarity function with a leave-one-out cross-validation approach. To calculate the F-measure, we performed one versus others classification approach. Then, we calculate the average of F-measure of all experiments. Table VII shows the results of the posture recognition using three methods GATSW with k-NN, C4.5, and SVM. These results were obtained using linear kernel with sequential minimal optimization in SVM and confidence factor equals to 0.25 and the minimum number of instances per leaf equals to 2 in C4.5.

The best F-measure that GATSW achieve is 0.82 using a window of $t_\Delta = 3$ s. Our proposed method improves the posture recognition on this dataset compare to other methods.

VI. CONCLUSION

In this paper, we described a similarity measure, TSW, for MATS data. To scale up TSW for real data, we introduced a window-based algorithm, WTSW, which uses TSW to search for the best match in long MATS. The proposed method provides a natural segmentation of the sequences that provides finer grained illness recognition solution. Since WTSW could be potentially slow for eldercare applications, we proposed a genetic

version of it, GATSW. Finally, we demonstrated how TSW can be used in a various frameworks for detecting health patterns. We tested our algorithms on multiple datasets: two synthetic ones, one obtained in TigerPlace and another one obtained from [46]. On the TigerPlace dataset, we obtained the abnormal days predictions F-measure of 0.75 on average over all residents used in this study. For future directions, we suggest researchers to use TSW with multiple methods together with a fusion methodology to reduce the false alarms rate. On the sensor dataset obtained from [46], we obtained 0.82 F-measure on posture recognition.

REFERENCES

[1] Editor, "TigerPlace: An innovative 'aging in place' community," *Am. J. Nursing*, vol. 113, no. 1, pp. 68–69, 2013.
[2] K. V. Grayson and A. V. Victoria, "The next four decades: the older population in the United States: 2010 to 2050," U.S. Dept. Commerce, Economics and Statistics Admin., U.S. Census Bureau, Washington DC, USA, 2010, pp. 1–14.
[3] J. S. Goodwin *et al.*, "Risk of continued institutionalization after hospitalization in older adults," *J. Gerontol. A. Biol. Sci. Med. Sci.*, vol. 66, pp. 1321–1327, 2011.
[4] T. L. Hayes *et al.*, "An unobtrusive in-home monitoring system for detection of key motor changes preceding cognitive decline," in *Proc. 26th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2004, pp. 2480–2483.
[5] S Intille *et al.*, "Using a live-in laboratory for ubiquitous computing research," in *Proc. 4th Int. Conf. Pervasive Comput.*, 2006, pp. 349–365.
[6] C. D. Kidd *et al.*, "The aware home: A living laboratory for ubiquitous computing research," in *Proc. 2nd Int. Workshop Cooperative Buildings Integrating Inf., Org., Archit.*, 1999, pp. 191–198.
[7] K. Z. Haigh, L. M. Kiff, and G. Ho, "Independent lifestyle assistant: Lessons learned," *Assistive Technol.*, vol. 18, pp. 87–106, 2006.
[8] M Skubic *et al.*, "A smart home application to eldercare: Current status and lessons learned," *J. Technol. Health Care*, vol. 17, pp. 183–201, 2009.
[9] D. Heise and M. Skubic, "Monitoring pulse and respiration with a non-invasive hydraulic bed sensor," presented at the 32nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., Buenos Aires, Argentina, 2010.
[10] R. Beckwith, "Designing for ubiquity: The perception of privacy," *IEEE Pervasive Comput.*, vol. 2, no. 2, pp. 40–46, Apr–Jun. 2003.
[11] D. Mack *et al.*, "A passive and portable system for monitoring heart rate and detecting sleep apnea and arousals: Preliminary validation," in *Proc. 1st Transdisciplinary Conf. Distrib. Diagnosis Home Healthcare*, 2006, pp. 51–54.
[12] M. J. Rantz *et al.*, "A technology and nursing collaboration to help older adults age in place," *Nursing Outlook*, vol. 53, pp. 40–45, 2005.
[13] C. K. Narayanan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive Mobile Comput.*, vol. 10, pp. 138–154, 2014.
[14] D. J. Cook *et al.*, "Activity discovery and activity recognition: A new partnership," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 820–828, Jun. 2013.
[15] M. Popescu and E. Florea, "Linking clinical events in elderly to in-home monitoring sensor data: A brief review and a pilot study on predicting pulse pressure," *J. Comput. Sci. Eng.*, vol. 2, pp. 180–199, 2008.
[16] M. Popescu and A. Mahnot, "Early illness recognition in older adults using in-home monitoring sensors and multiple instance learning," *Methods Inf. Med.*, vol. 51, no. 4, pp. 359–367, 2012.
[17] Y. Lee, D. Ho, and M. Popescu, "Microphone array system for automatic fall detection," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 5, pp. 1291–1301, May 2012.
[18] T. Kahveci, A. Singh, and A. Gurel, "Similarity searching for multi-attribute sequences," in *Proc. 14th Int. Conf. Sci. Statist. Database Manag.*, 2002, pp. 175–184.
[19] G. A. H. Ten *et al.*, "Multi-dimensional dynamic time warping for gesture recognition," in *Proc. 13th Conf. Adv. School Comput. Imaging*, 2007.
[20] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh, "Indexing multi-dimensional time-series with support for multi distance measures," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 216–225.
[21] P. Sanguansat, "Multiple multidimensional sequence alignment using generalized dynamic time warping," *WSEAS Trans. Math.*, vol. 11, no. 8, pp. 684–694, 2012.

[22] H. Ding *et al.*, "Querying and mining of time series data: Experimental comparison of representations and distance measures," *Proc. VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[23] T. Rakthanmanon *et al.*, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 262–270.

[24] J. Lin *et al.*, "Experiencing SAX: A novel symbolic representation of time series," *Data Mining Knowl. Discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[25] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *J. Molecular Biol.*, vol. 147, pp. 195–197, 1981.

[26] L. Bergroth, H. Hakonen, and T. Raita, "A survey of longest common subsequence algorithms," in *Proc. 7th Int. Symp. String Process. Inf. Retrieval*, 2000, pp. 39–48.

[27] Z. Hajihashemi and M. Popescu, "Predicting health patterns using sensor sequence similarity and NLP," in *Proc. IEEE Int. Conf. Bioinformat. Biomed. Workshops*, 2012, pp. 948–950.

[28] F. Duchene, C. Garbay, and V. Riale, "Learning recurrent behaviors from heterogeneous multivariate time-series," *Art. Intell. Med.*, vol. 39, pp. 25–47, 2007.

[29] Z. Hajihashemi and M. Popescu, "An early illness recognition framework using a temporal Smith Waterman algorithm and NLP," in *Proc. AMIA Fall Symp.*, 2013, pp. 548–557.

[30] G. Demiris *et al.*, "Older adults' attitudes towards and perceptions of 'smarthome' technologies: A pilot study," *Med. Inf. Internet Med.*, vol. 29, no. 2, pp. 87–94, Jun. 2004.

[31] G. Demiris, M. J. Rantz, M. A. Aud, K. D. Marek, H. W. Tyrer, M. Skubic, and A. A. Hussam, "Seniors' attitudes towards home-based assistive technologies," presented at the 29th Annu. MNRS Res. Conf., Cincinnati, OH, USA, Apr. 1–4, 2005.

[32] D. C. Mack, J. T. Patrie, P. M. Suratt, R. A. Felder, and M. Alwan, "Development and preliminary validation of heart rate and breathing rate detection using a passive, ballistocardiography-based sleep monitoring system," *IEEE Trans. Inform. Technol. Biomed.*, vol. 13, no. 1, pp. 111–120, Jan. 2009.

[33] M. Rantz *et al.*, "Evaluation of health alerts from an early illness warning system in independent living," *Comput., Inf., Nursing*, vol. 31, no. 6, pp. 274–280, 2013.

[34] M. Popescu, "An ontological fuzzy Smith-Waterman with application to patient retrieval in electronic medical records," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2010, pp. 1–6.

[35] L. Wei *et al.*, "Atomic wedgie: Efficient query filtering for streaming time series," in *Proc. 5th IEEE Int. Conf. Data Mining*, Houston, TX, USA, Nov. 27–30, 2005, pp. 490–497.

[36] S. Waterman, "Some biological sequence metrics," *Adv. Math.*, vol. 20, no. 3, pp. 367–387, 1976.

[37] M. Popescu and E. Florea, "Linking clinical events in elderly to in-home monitoring sensor data: A brief review and a pilot study on predicting pulse pressure," *J. Comput. Sci. Eng.*, vol. 2, no. 1, pp. 180–199, Mar. 2008.

[38] M. Popescu and A. Mahnot, "Acoustic fall detection using one-class classifiers," in *Proc. 31th Int. Conf. IEEE Eng. Med. Biol. Soc.*, Minneapolis, MN, USA, Sep. 2–6, 2009, pp. 3505–3508.

[39] P. Dawadi *et al.*, "Smart home-based longitudinal functional assessment," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 1217–1224.

[40] S. Robben, "How is grandma doing? Predicting functional health status from binary ambient sensor data," in *Proc. AAAI Fall Symp. Series*, 2012, pp. 26–31.

[41] G. Virone *et al.*, "Behavioral patterns of older adults in assisted living," *IEEE Trans. Inf. Technol. Biomed.*, vol. 2, no. 3, pp. 387–398, May 2008.

[42] Z. Hajihashemi and M. Popescu, "Detection of abnormal sensor patterns in eldercare," in *Proc. E-Health Bioeng. Conf.*, 2013, pp. 1–4.

[43] Z. Hajihashemi and M. Popescu, "Detecting daily routines of older adults using sensor time series clustering," presented at the 36th Int. Conf. IEEE Eng. Med. Biol. Soc., Chicago, IL, USA, Aug. 26–30, 2014.

[44] M. Skubic, R. D. Guevara, and M. Rantz, "Testing classifiers for embedded health assessment," in *Proc. Int. Conf. Smart Homes Health Telematics*, Artimino, Italy, Jun. 12–15, 2012, pp. 198–205.

[45] L Wang *et al.*, "iVAT and aVAT: Enhanced visual analysis for cluster tendency assessment," *Adv. Knowl. Discovery Data Mining*, vol. 6118, pp. 16–27, 2010.

[46] B. Kaluza *et al.*, "An agent-based approach to care in independent living," *Ambient Intell., Lecture Notes Comput. Sci.*, vol. 6439, pp. 177–186, 2010.