Noname manuscript No.
(will be inserted by the editor)

Temporal Features and Relations Discovery of Activities from Sensor Data

Ehsan Nazerfard

the date of receipt and acceptance should be inserted later

Abstract One of the most important problems that arises during the knowledge discovery from data (KDD) and data mining process in many new emerging technologies is mining data with temporal dependencies. One such application is activity recognition and prediction. Activity recognition is used in many real world settings, such as assisted living systems. Although activity recognition has been vastly studied by many researchers, the temporal features that constitute an activity, which can provide useful insights for activity models, have not been exploited to their full potentials by mining algorithms. In this paper, we utilize temporal features for activity recognition and prediction in assisted living settings. We discover temporal relations such as the order of activities, as well as their corresponding start time and duration features. Analysis of real data collected from smart homes was used to validate the proposed method.

Keywords Temporal pattern mining · Frequent itemset mining · Association Rules · Activity Recognition · Activity Prediction · Clustering · Internet of Things · Smart Homes

1 Introduction

Recent advancements in machine learning and data mining as well as pervasive sensing technologies have opened doors to a wide variety of pervasive and context-aware applications. One research area that has benefited from the aforementioned advancements is smart environments. A smart environment is any physical environment (e.g. home, office, conference room) that senses the state of its resident and the physical surroundings and acts in order to ensure the well-being of the resident and the environment. Data is obtained from the sensors and is analyzed using machine learning and data mining techniques. The main goal of such technologies is to achieve greater comfort, productivity, and energy efficiency.

Over the past decade, researchers have recognized the importance of smart environment technologies to provide in-home assisted living (Lotfi et al. 2012) and companies have started taking advantage of such technologies in the marketplace (BrainAid 2013). In many smart home projects, the ultimate goal is to automate residents' interactions with the environment, in particular interactions that are repetitive or cumbersome to perform for older adults or patients with cognitive impairments. An example of assisted living technologies is a remote health monitoring system that monitors and tracks activities of daily living (ADLs) of older adults with memory impairment. ADLs consist of self-care activities such as eating, cooking, taking medication, bathing and sleeping. The ability to perform ADLs independently and completely on a regular basis provides measurements for the functional well-being of inhabitants. The need for developing such in-home assisted living is highlighted by the increasing aging population, the cost of health care facilities, and individuals' preference to stay at their homes, rather than health care facilities.

In addition to the physical infrastructure, there have also been a number of machine learning methods for activity-related tasks including, but not limited to: activity recognition, activity discovery, activity prediction and activity reminder systems. Since many of these methods rely upon activity recognition, this subfield has received the utmost attention in the literature. Activity recognition methods range from simple approaches such as decision trees (Maurer et al. 2006) and naïve Bayes classifiers (Brdiczka et al. 2005) to

E. Nazerfard
Ambient Intelligence Research Laboratory (AIR Lab)
Department of Computer Engineering
Amirkabir University of Technology, Tehran, Iran
Tel.: +98 21 64545113
E-mail: nazerfard@aut.ac.ir

more complex methods such as hidden Markov models (Singla et al. 2009), dynamic Bayesian networks (Hall et al. 2009), and conditional random fields (Vail et al. 2007).

As intelligent systems become more prevalent, they must also be able to predict the occurrence of various events, such as the activities that residents perform, in order to have necessary foresight to make decisions in various situations. An especially common problem is sequential prediction, where a sequence of events is used to predict future events. The applications of such a prediction model are endless. In a smart home setting, for instance, predicting residents' activities provides a basis to automate their interactions with the environment to provide context-aware services. As an example, we will briefly discuss how an activity prediction component together with an activity recognition module can be utilized to generate automated context-aware prompts for smart home residents.

Reminder systems have been long in existence and range from simple alarm clocks to complex systems that are based on rules, planning, or machine learning. Rule-based reminder systems allow a user to specify rules based on time, context, and preferences (Lim et al. 2008). More adaptive reminder systems integrate reinforcement learning (Rudary et al. 2004), which requires a pre-specified complete schedule of activities but can make adjustments without direct user feedback. Other approaches use dynamic Bayesian networks (Pollack et al. 2003), Markov decision processes (Pineau et al. 2003), and Markov-based planning (Boger et al. 2005) to deliver timely prompts for these pre-scheduled activities. Active learning has also been employed (Weber and Pollack 2007) to interactively manage calendar synchronization.

When older adults with cognitive impairment fail to initiate or complete everyday ADLs, typically caregivers are responsible to monitor ADLs and deliver a prompt. The prompt is defined as any form of verbal or non-verbal intervention delivered based on time or context to assist an individual to complete an activity successfully. The discussed interventions are time consuming and burdensome and often have negative impact on the caregiver's own health. Smart home technologies that can detect when assistance is needed and automatically deliver prompts can potentially reduce caregivers' burden and allow aging adults to retain their functional independence longer.

While reminder systems have been explored deeply in the literature, few systems take an individual's behavioural patterns into account to provide context aware prompts. However, studies suggest that activity aware prompts offer significant advantages over traditional time based prompts (Kaushik et al. 2008). By taking advantage of such a prediction module, a reminder system can customize its behaviour to fit the lifestyles of the residents with no input on their part.

As an example application of activity prediction, the technology can feasibly utilize data collected from a smart home to learn context-aware rules for prompting the resident of the home to initiate important daily activities. Such an application assumes that sensor data is collected in a home while the resident performs her routine daily activities. We also assume that the training data is available from when the resident was performing the activities correctly or was prompted by a caregiver to initiate daily activities. Finally, we assume that we are given a list of critical activities for which the resident needs to be prompted. The following scenario highlights the role of an activity prediction component together with the activity recognition module to provide an automated context-aware prompt, assuming that "Taking Medication" is a critical activity (Nazerfard and Cook 2015):

"In the morning, the activity recognition module recognizes that the breakfast activity has taken place. Then the activity prediction component, which has already been trained with the correctly labeled activities, makes the following prediction with a high enough confidence: The "Taking Medication" activity should follow within 30 *minutes". Then the activity prompting system would have a relative time offset for when "Taking Medication" usually happens. When the typical timespan passes and the medication is not taken, a prompt is delivered."*

The above scenario shows how temporal features can be useful in an assisted living setting. The discovered temporal information can be used to construct a schedule of activities for an upcoming period. Such a schedule is constructed based on the predicted start time, as well as the relative order of the activities.

In this paper, we propose a framework for discovering and representing temporal aspects of activity patterns, including temporal ordering of activities and their usual start time and duration. The discovered temporal information can be beneficial in many applications, such as for home automation, constructing the schedule of activities for a context-aware activity reminder systems, and abnormal behavior detection in smart homes. The rest of the paper is organized as follows. We first review some related work in the field of smart environment together with activity prediction approaches and (temporal) association rules. Next we overview the frequent patten mining terminologies. We then present the details of our proposed temporal feature and relation discovery model. Next, we provide the validation results of the proposed model. Finally, we present some concluding remarks and future directions.

2 Related Work

In this section, we first review a number of smart environments deployed during the past decade. Next, we discuss recent proposed activity prediction approaches in the field of smart environments. Finally, we briefly review the concept of association rules.

2.1 Smart Environments

Over the past decade, a number of smart environment testbeds have been deployed, including but not limited to: MavHome project (Cook et al. 2003), Georgia Tech Aware Home (Abowd and Mynatt 2004), Gator Tech Smart House (Helal et al. 2005), iDorm (Doctor et al. 2005), PlaceLab (Intille et al. 2006), and CASAS project (Crandall 2011). In addition to creating physical testbeds, researchers have designed approaches to track locations and activities of inhabitants, discover abnormal behaviour, deliver timely prompts, and predict inhabitants' future activities. The differences among existing approaches can be categorized as follows:

1. Differences in sensor modalities used to monitor activities. These include but are not limited to: supervision cameras (Mocanu et al. 2011), RFID tags (Gu et al. 2011), accelerometers (Yin et al. 2008), and wearable sensors (O'Donovan et al. 2009).
2. Differences in methods designed to learn activity patterns. These include but are not limited to: Bayesian networks (Kasteren and Krose 2007), fuzzy logic (Medjahed et al. 2009), artificial neural networks (Mahmoud et al. 2013), and support vector machines (Hamm et al. 2013).
3. Differences in experimental conditions. These include but are not limited to: smart environment inhabitants performing scripted activities (Maurer et al. 2006) and smart environment inhabitants performing regular unscripted daily living activities (Cook et al. 2013).

Based on the above-mentioned advancements, researchers have realized the importance of employing smart home technologies for health monitoring and assistance (Lotfi et al. 2012; Catarinucci et al. 2015; Hossain and Muhammad 2016; Chen et al. 2017) and companies have started looking into the potential of such technologies in the market (BrainAid 2013).

2.2 Activity Prediction Approaches

In spite of the significant work that has been done to recognize and track activities in the smart home research (Heung-II et al. 2010; Cook et al. 2013), less attention has been paid to predict the occurrence of various events, such as the activities that residents perform. As smart environments become more widespread, an important functionality that they require to posses is the ability to predict the activities that residents perform, in order to have the visibility to make decisions in various situations. In machine learning literature, "Prediction" often refers to sequential prediction, where the goal is to predict the next event based on a known limited history of past events.

One of the seminal works in this area, the Active LeZi (ALZ) algorithm (Gopalratnam and Cook 2007) approaches the problem from an information theoretic perspective using compression methods. ALZ is an online sequential prediction algorithm that can reason about the future in stochastic domains with no domain-specific information. The authors experimentally analyzed their sequential prediction algorithm using real sequential data obtained from the MavHome smart home environment (Cook et al. 2003). Its precursor, LeZi-update (Dufkova et al. 2009), provides a pattern matching method for location management in cellular communication networks that can exploit the position information of an inhabitant for message routing. The LeZi-update framework uses a symbolic space to represent the sensing zone of the smart environment as alphabetic symbols. As a result, it captures the entire inhabitant's movement history as a string of symbols.

Furthermore, the researchers in (Mocanu and Florea 2012) propose a multi-agent architecture for a supervising system which includes an activity prediction layer. Their activity prediction component takes advantage of the active LeZi algorithm (Gopalratnam and Cook 2004) in order to detect emergencies in smart environments. Also, the authors in (Tapia et al. 2010) employ the activity prediction component to intervene and interact with the user as a means of prompting the user and preventing accidents. Moreover, the researchers in (Mahmoud et al. 2013) discuss the application of soft computing techniques in prediction of an older adult's behaviour in a smart environment. In order to build the prediction model, they examine different types of artificial neural networks. Their results suggest that recurrent neural networks such as NARX achieve a great ability to finding the temporal relationships of the input patterns.

While many activity prediction and recognition approaches have been proposed, they are typically designed for constrained situations with pre-segmented data and a single user environment. The researchers in (Krishnan and Cook 2014) have extended these approaches to consider generalization of activity models over multiple users with real-time labeling. In order to avoid off-line data segmentation, the activity recognition component extracts features from a sliding window that moves over the data in real time as it is collected. The window size dynamically adjusts to sensor readings based on likely current activities and their associated likely durations.

Just recently, Minor and Cook (2017) describe an algorithm for automated activity prediction in smart home environments. The authors refer to their proposed algorithm as

AF, short for activity forecasting. AF predicts the time that will elapse until a target activity occurs. This method generates an activity forecast using a regression tree classifier and can be applied in forecasting situations where a numeric time prediction is valuable.

In summary, Table 1 highlights the previous prediction approaches discussed in this section. The main characteristics and contributions of the proposed prediction method are as follows:

- Discovering temporal relations such as the order of activities as well as the their start time and duration features.
- Handling the outlying observations through the probabilistic clustering techniques.
- Employing frequent pattern mining to discover the temporal relations of activities.

2.3 Association Rule Mining

The frequent pattern mining techniques focus on finding "interesting patterns" from a large set of data items. One of the well-known frequent pattern mining approaches is the association rules (Agrawal et al. 1993), which is usually used to find all the co-occurrence relationships from a set of items called associations. A simple example that is often used to explain the concept of association rule is discovering items that are purchased together within a transactional dataset.

Mining association rules in a market basket database is a well researched area. Even though the main problem of finding association rules is well defined and a number of algorithms exist in the literature to solve it, some particularities are not handled by these common algorithms. One of these particularities are the items that are being sold together during some specific time intervals, e.g. items that are better sold together in morning times. Therefore, a temporal association rule is defined as an association rule that holds during specific time intervals. The authors in (Li et al. 2001) study temporal association rules during time intervals specified by user-given calendar schemas. The discovered association rules along with their temporal patterns, in terms of calendar schemas, are easier to understand. This extension suggests that we might discover different rules for different timeframes. As a result, a rule might be valid during certain timeframe, but not during some other timeframes.

Activity pattern dataset in smart homes also include a timestamp. The timestamp implies when a particular activity was performed, or more specifically when a specific sensor was triggered. Similar to association rule mining, considering the concept of temporal features in activity patterns can be quite useful. For instance, in a home automation setting, we can determine when a certain activity is expected to occur, and which activities are most likely to occur next (Nazerfard et al. 2010). Despite the potential use of temporal features in activity patterns, this key aspect is usually neglected and has not been exploited to its full potential.

3 Frequent Pattern Mining

The proposed model in this paper is based on frequent pattern mining techniques. The main difference between the frequent pattern mining methods and the other mining approaches is that the former techniques are focused on finding "interesting patterns" from a large set of data items.

Frequent patterns are patterns (e.g., itemsets) that appear frequently in a dataset. For instance, a set of items, such as *butter* and *bread*, that appear frequently together in a transaction dataset is a frequent itemset. As another example, a subsequence such as buying first a *laptop* and then a *backpack*, if it occurs frequently in a shopping transactional database, is a frequent associated pattern.

3.1 Preliminaries

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of items. An *itemset* C is a subset of I, i.e. $C \subseteq I$. An itemset that contains k items is a *k-itemset*. We denote the size of C, i.e. the number of items in C, by $|C|$. A transaction is defined as $T = (tid, C)$, where *tid* is a transaction identifier. A transaction database is a set of transactions, which is denoted by D.

The *support_count* of an itemset X is the number of transactions that contain the itemset and is formally defined as follows:

$$Support\_count(X) = |\{tid \mid X \subseteq C,\ (tid, C) \in D\}| \quad (1)$$

Therefore, the *support* of an itemset X is the proportion of transactions that contain the itemset and is defined as follows:

$$Support(X) = \frac{support\_count(X)}{|D|}, \quad (2)$$

where $|D|$ denotes the total number of transactions. If the *support* of an itemset X satisfies a given threshold, then X is a frequent itemset.

Moreover, an association rule is an implication of the form $A \implies B$, where $A \subset I$, $B \subset I$, $A \neq \emptyset$, $B \neq \emptyset$, and $A \cap B = \emptyset$. The rule $A \implies B$ holds in the transaction set D with *support* s, where s is the percentage of transactions in D that contain $A \cup B$, i.e. $support(A \implies B) = support(A \cup B)$. The rule $A \implies B$ has *confidence* c in the transaction set D, where c is the percentage of transactions in D containing A that also contain B:

Table 1: The previous activity prediction studies.

	LeZi-update	ALZ	NARX	AF
Temporal Features Discovery	No	No	Yes	No
Time Forecasting	No	No	No	Yes
Outlier Detecting	No	No	No	No
Underlying Methodology	Pattern Matching	Compression Methods	Recurrent Neural Networks	Regression Trees

$$\begin{aligned} Confidence(A \implies B) &= \frac{support(A \cup B)}{support(A)} \\ &= \frac{support\_count(A \cup B)}{support\_count(A)} \end{aligned} \tag{3}$$

A rule is called *strong* if it satisfies both a minimum support threshold (*minsup*) and a minimum confidence threshold (*minconf*). Eq. 3 suggests that once the support counts of A and $A \cup B$ are determined, it is straightforward to generate the corresponding association rule $A \implies B$ and check whether it is strong or not. As a result, the problem of mining association rules can be reduced to mining frequent itemset. In summary, the problem of discovering strong association rules can be viewed as a two-step process:

– Frequent itemset mining.
– Generating strong association rules from the found frequent itemsets.

3.2 Frequent Itemset Mining

In this section, two well-known frequent itemset mining algorithms are reviewed: Apriori and FP-growth.

3.2.1 Apriori Algorithm: Finding Frequent Itemsets using Candidate Generation

The Apriori algorithm, by Agrawal and Srikant (1994), begins by discovering frequent items by counting them, making a pass over D. Then it combines these frequent items to generate candidate 2-itemsets, and determines their supports by making another pass over D, removing infrequent candidates. The algorithm iteratively continues to extend k-itemset candidates by one item and determines their supports by making another pass over D to check if they are frequent or not. One should note that if an itemset is frequent, all of its non-empty subsets must be frequent. Also known as the *Apriori property*, this property indicates that if an itemset is not frequent, none of its subsets are frequent. By taking advantage of the the Apriori property, the algorithm prunes those candidates for which a subset is infrequent.

The Apriori algorithm process candidates in a prefix-tree structure. The common *k-prefix* of two itemsets are the first k items in those sets that are common. The *prefix tree* is a tree structure where each path represents an itemset, which is the path from the root to a node, and sibling nodes share the same prefix. The Apriori-based algorithms process candidates in a prefix-tree in a level-wise manner, i.e. k-itemsets must be processed before $(k+1)$-itemsets.

3.2.2 FP-growth Algorithm: Finding Frequent Itemsets without Candidate Generation

The Apriori algorithm can suffer from two major problems. First, it may need to generate a large number of candidate sets. For instance, if there are 2000 frequent 1-itemsets, the algorithm will require to generate around 10^6 2-itemset candidates. Also, the Apriori algorithm may need to repeatedly scan the whole database and check a huge set of transactions by determining the support of the candidate itemsets.

The FP-growth (short for frequent pattern growth) algorithm, by Han et al. (2000), is an attempt to mine the complete set of frequent itemsets, without such a costly candidate generation. The FP-growth algorithm compresses the database representing frequent items into a FP-tree (short for frequent pattern tree), which is an extended prefix-tree structure for storing compressed and association information about frequent patterns. The algorithm then divides the compressed database into a set of conditional databases, each associated with one frequent or *pattern fragment*, and mines each database separately. For each pattern fragment, only its associated datasets need to be considered. As a result, this method may dramatically reduce the search space.

3.3 Generating Association Rules

As already discussed in the *Preliminaries* section, the problem of discovering association rules can be viewed as a two-step process: finding frequent itemsets and then generating the strong association rules. The first step can be done using either of the Apriori or FP-growth approaches discussed in the previous section.

Once the frequent itemsets from a transaction database D have been found, it is straightforward to generate strong association rules from them, using the Apriori algorithm, where strong association rules are those that satisfy both *minsup* and *minconf*. This step can be done using Eq. 4 (also appears in section 3.1):

$$Confidence(A \implies B) = P(B|A) = \frac{support\_count(A \cup B)}{support\_count(A)} \quad (4)$$

According to Eq. 4, association rules can be generated as follows:

- For each frequent itemset l, generate all nonempty subsets of l.
- For every nonempty subset s of l, generate the following rule:

  $s \implies (l - s)$,

  if $\frac{support\_count(l)}{support\_count(s)} \geq minconf$, where $minconf$ is the minimum confidence threshold.

Note that since the rules are generated from frequent itemsets, each one already satisfies the minimum support condition. As an example, given a transaction database D placed in a supermarket, we may find an association rule of the following form:

$$PeanutButter \implies Bread \quad (support: 3\%, confidence: 75\%), \quad (5)$$

which indicates that 3% of all transactions contain the items $peanutbutter$ and $bread$, and 75% of the transactions that have the item $peanutbutter$, contain the item $bread$ as well.

One interesting extension to association rules is to include a temporal dimension. For instance, $peanutbutter$ and $bread$ may be purchased together mainly between $6: 30AM$ and $11AM$. Thus, we may find the above association rule has a support as high as 40% among the transactions that occur between $6: 30AM$ and $11AM$ and has a support as low as 4% in other transactions. We refer to the above-mentioned extension as temporal association rules.

4 Proposed Model

The proposed TEREDA model, abbreviation for TEmporal features and RElations Discovery of Activities, discovers temporal features and relations of activity patterns from sensor data. The model is able to discover features and relations, such as the order of the activities, their usual start times and durations through the use of rule mining and clustering techniques.

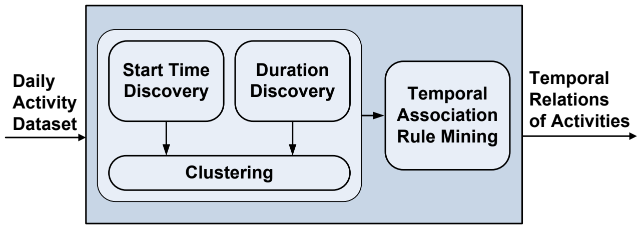

Fig. 1: The TEREDA architecture

The architecture of TEREDA is illustrated in Fig. 1, which consists of two main components: the temporal feature discovery component and the temporal relation discovery. Each component will be described in more depth in the following sections. The input consists of a set of sensor events collected from various sensors deployed in the space. Each sensor event includes an identifier (ID), a timestamp and an optional activity label. In order to better understand how TEREDA works, we consider an example, the *Taking Medication* activity, throughout following discussions.

4.1 Temporal Activity Features Discovery

Initially, start time and duration of activities are obtained for each activity. After extracting start times for all instances of a specific activity, start times are clustered in order to obtain a canonical representation. For this purpose we take advantage of the Expectation Maximization (EM) clustering algorithm to construct a normal mixture model for each activity start time.

Let t_i denote start time of the i-th instance of activity A, depicted as A_i. The probability of t_i belonging to a certain cluster, the k-th cluster, can be expressed as a normal probability density function parameterized by $\Theta_k = (\overline{x}, s)$, as represented in Equation 6, where $\overline{x}$ and s are sample mean and sample variance, respectively.

$$Prob(t_i|\Theta_k) = \frac{1}{\sqrt{2\pi s^2}} e^{-\frac{(t_i - \overline{x})^2}{2s^2}} \quad (6)$$

The parameters of the mixture normal model are determined automatically from the available data. Fig. 2 illustrates the results of finding canonical start times for the “Taking Medication” activity in the form of a mixture of four normal distributions. According to normal distribution characteristics, the distance of “two standard deviations” from the mean account for approximately 95% of the values (see Fig. 3). Therefore, if only observations falling within two standard deviations are considered, observations that are deviating from the mean will be automatically discarded. Such observations, distant from the rest of the data are regarded as “outliers”.

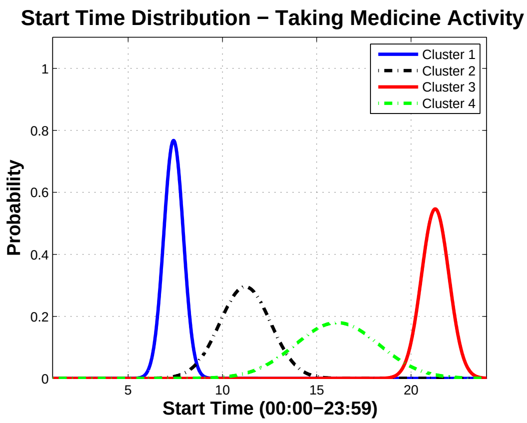

Fig. 2: A mixture model for the start time of the Taking Medication activity

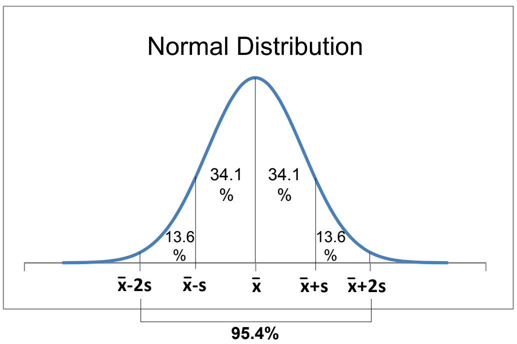

Fig. 3: The normal distribution characteristics

Duration of an activity is also considered in addition to start time, where average duration of all instances within the cluster are calculated for each resulting cluster from the start time discovery step.

4.2 Temporal Activity Relations Discovery

Discovering *temporal relations* of activities is the main component of TEREDA, wherein features obtained from the previous step, i.e. the canonical start times and durations are used to produce a set of temporal relations between activities. The temporal relations will determine the order of activities with respect to their start times, i.e. for a specific time what are the most probable activities following a specific activity. Such results can be useful in a variety of activity prediction scenarios. To discover the temporal relations of activities, we employ the FP-growth algorithm.

In order to provide a more precise understanding of the temporal relation discovery component we consider the following notations. let B_j denote the successor activity of A_i, where A_i and B_j denote the i-th and j-th instances of activities A and B, respectively. As previously mentioned, each

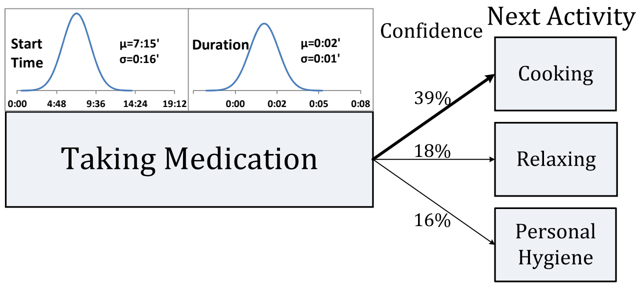

Fig. 4: Temporal relations of the Taking Medication activity (The results are shown for the first cluster).

activity instance belongs to a specific cluster Θ_k defined by the start time of the activity instance. Moreover, the A_i^k notation is used to show that activity A_i belongs to a specific cluster Θ_k. The temporal relation "B follows A" formulated as $A \implies B$ is ultimately obtained and the set containing instances of all activities is represented as D.

Denoting the estimated mean and standard deviation of cluster k by $\overline{x}_k$ and s_k, we refer to the number of instances of all activities and activity A falling within $[\overline{x}_k - 2s_k, \overline{x}_k + 2s_k]$ interval as $|D^k|$ and $|A^k|$, respectively. Then we can define the *support* of the "follows" relation as in Equation 7 and its confidence as in Equation 8.

$$Support(A^k \implies B) = \frac{\sum_{i,j}(A_i^k \implies B_j)}{|D^k|} \tag{7}$$

$$Confidence(A^k \implies B) = \frac{\sum_{i,j}(A_i^k \implies B_j)}{|A^k|} \tag{8}$$

The outcome of this step is a set of temporal relation rules corresponding to each cluster. Fig. 4 illustrates the discovered temporal relation rules, with *support* and *confidence* values greater than 0.1, for the first cluster of the Taking Medication activity. According to Fig. 4, if the *Taking Medication* activity occurs in time interval $[6:43, 7:47]$, it would usually lasts less than 2 minutes and the next activities would typically be "Cooking" with a confidence of 0.39, "Relaxing" with a confidence of 0.18 and "Personal Hygiene" with a confidence of 0.16.

5 Experimental Results

This section presents experimental results of the proposed TEREDA model. Before getting into the details of our results, we explain the settings of our experiment.

5.1 Experimental Setup

Two one bedroom single resident smart home apartments, referred to as Apt1 and Apt2, hosting older adults perform-

Algorithm 1: TEREDA algorithm

Data: A dataset of activity patterns
Result: A set of temporal association rules between activities, as well as their corresponding start time and duration features

```
for an activity A do
    t = ∅
    /* t denotes the start time of the activity A */
    for all A_j do
        /* A_j denotes instance j of the activity A */
        t = t ∪ t_j
        /* t_j denotes the start time of A_j */
    end
    Cluster all t_j's using the EM algorithm
    /* A^k denotes instances of A falling into cluster k */
    for each cluster k do
        - Model the cluster using a normal distribution parameterized by Θ_k = (x̄_k, s_k).
        - Keep the observations falling within [x̄_k − 2s_k, x̄_k + 2s_k] interval and discard the rest.
        - Employ the FP-growth algorithm to find frequent sets of activities together with the Apriori algorithm to
            generate strong association rules.
    end
end
```

ing normal unscripted daily activities were selected as study environment. Sensors were installed on ceilings, walls, doors and cabinets in order to track resident movements. The sensor layouts for our testbeds are illustrated in Fig. 5, where red circles and blue triangles represent infrared motion/area sensors and magnetic door/cabinet sensors, respectively. A sensor network was applied to capture all events generated by the sensors and middleware was used to store events in an SQL database.

A sensor network captures all of the events generated by the sensors, and our middleware stores them in an SQL database. As a means for providing real activity data for our experiments, data were collected while residents were living in smart homes performing normal daily routines. Table 2 provides the characteristics of our smart home testbeds[1].

Table 2: Characteristics of the smart homes for the present study

	Apt1	**Apt2**
# of motion sensors	20	18
# of door/cabinet sensors	12	12
# of residents	1	1
# of sensor events collected	368,821	248,923
Timespan	6 months	4 months

For our experiments, we consider 11 ADLs performed by residents of the smart homes. The list of the activities taken into consideration for the present study are as follows: Bathing, Bed-Toilet Transition, Cooking, Eating, Enter Home, Housekeeping, Leave Home, Personal Hygiene, Relaxing, Sleeping, and Taking Medication.

[1] Results provided in the validation section correspond to the Apt1 dataset, available online at http://eecs.wsu.edu/∼nazerfard/AIR/datasets/data1.zip

The datasets used consist of a set of discrete individual sensor events collected from various sensors deployed in the space. Each corresponding sensor within a smart home generates a message if resident movement is detected in its field of view. Events in the dataset were manually annotated with corresponding activity labels by trained researchers who employed visualization tools and interviews with residents to generate accurate ground truth labels. Table 3 provides a sample data related to the "Taking Medication" activity, where $D27$ and $M03$ represent a cabinet sensor and a motion sensor, respectively.

Table 3: A sample for sensor events used in our study.

Timestamp	**Sensor ID**	**Label**
2009-07-18, 07:20:43	D27	Taking Medication begin
...	...	...
2009-07-18, 07:20:55	M03	...
...	...	...
2009-07-18, 07:21:12	D27	Taking Medication end

Table 4 provides the parameter values used for the EM clustering and FP-growth association rules for our experiments.

Table 4: The EM clustering and FP-growth parameters

EM Clustering	max-iterations = 100
	min-standard deviation = $1.0e-6$
FP-growth	minsup = 0.1
Apriori	minconf = 0.1

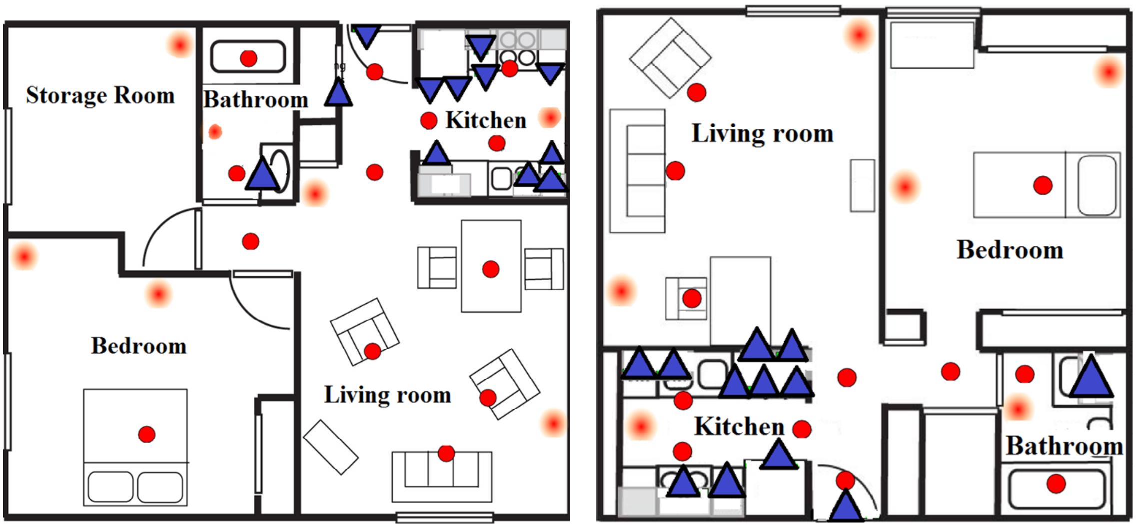

Fig. 5: Layouts for Apt1 (left) and Apt2 (right), where the red circles represent motion/area sensors, and the blue triangles represent door/cabinet sensors.

Table 5: The number of discovered start time clusters for each activity

Activity	# of clusters	Activity	# of clusters
Bathing	2	Housekeeping	1
Bed-Toilet Transition	2	Personal Hygiene	4
Cooking	5	Relaxing	7
Eating	5	Sleeping	2
Enter Home	4	Taking Medication	4

5.2 Evaluation of TEREDA

In this section, we provide the results of running TEREDA on the smart home dataset and compare its accuracy against a number of other prediction techniques. As already mentioned in section 1, employing the EM clustering algorithm enables TEREDA to discover different numbers of clusters for different activities. Taking into account the EM algorithm parameters provided in Table 4, Table 5 presents the number of discovered clusters, corresponding to the start times of each activity, after running TEREDA on our dataset.

For instance, results in Table 5 for the "Enter Home" activity suggest that TEREDA discovers 4 start time clusters for this activity. The discovered temporal relations for the Enter Home activity are illustrated in Fig. 6 through Fig. 9. As mentioned in section 4.1, in order to handle outliers, TEREDA only retains start time values within the [$\overline{x}$-2s, $\overline{x}$+2s] interval for each activity. It is worth noting that "Enter Home" is the one activity in our dataset for which considering the "duration" feature is not applicable. Consequently, there are no normal distributions corresponding to the durations of the "Enter Home" activity in Fig. 6 to Fig. 9.

According to Fig. 6, if the "Enter Home" activity occurs in the [$7:52$, $10:08$] timeframe, it is typically followed by the "Relaxing" and "Eating" activities with confidence values of 0.41 and 0.25, respectively. Furthermore, Fig. 7 indicates that when "Enter Home" occurs between $13:27$ and $13:35$, the next followup activities are "Cooking" with a confidence of 0.36, "Relaxing" with a confidence of 0.17, and "Eating" with a confidence of 0.15. It is worth mentioning that only maximum of the three most probable succeeding activities with confidence values greater than 0.1 are represented. Fig. 8 indicates that when the "Enter Home" activity takes place between $16:38$ and $17:10$, the most probable followup activity is "Cooking" with a confidence of 0.82. Finally, Fig. 9 suggests that if the "Enter Home" occurs during the [$18:41$, $21:57$] interval, it is most likely followed by three activities: Taking Medication, Eating, and Relaxing. The confidence for these followup activities are 0.28, 0.27, 0.20, respectively.

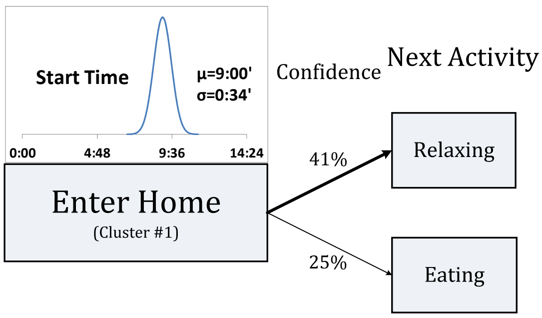

Fig. 6: Results for the 1st cluster

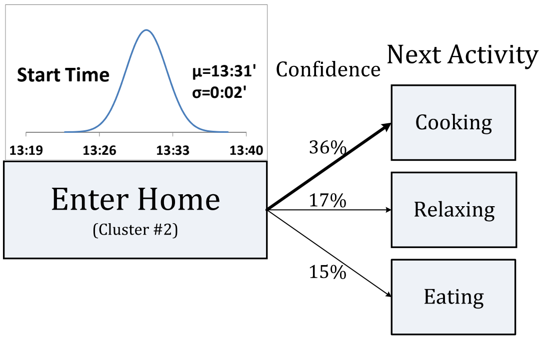

Fig. 7: Results for the 2nd cluster

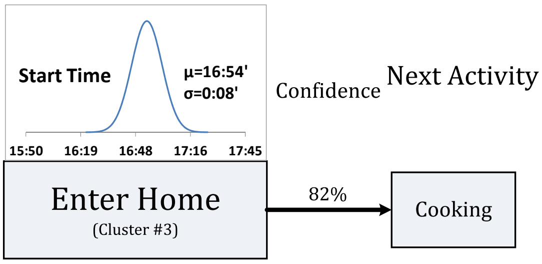

Fig. 8: Results for the 3rd cluster

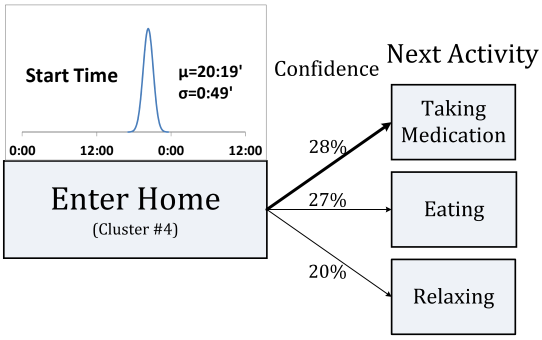

Fig. 9: Results for the 4th cluster

Table 6 provides results of running TEREDA on our smart home data for the first five activities. The results indicate that the Bathing activity takes place in two time clusters. For the first cluster, the activity normally happens between $8:31$ and $15:15$, where it lasts between 3 to 9 minutes and is followed by Personal Hygiene, Eating, or Leave Home activities with confidence values of 0.45, 0.12, and 0.12, respectively. As previously mentioned, we represent maximum of three followup activities whose confidence values are greater than 0.1. The results corresponding to the second cluster of the Bathing activity suggests that the Bathing activity is usually performed during the $20:24$ to $22:06$ time interval, where it lasts approximately 19 to 21 minutes and is followed by three activities: Personal Hygiene, Sleeping and Relaxing. The corresponding confidence values for these three activities are 0.58, 0.13 and 0.13, respectively.

According to results of TEREDA regarding the Bed-Toilet activity, it seems that this activity occurs mostly in two main time clusters. The first cluster usually happens between $2:36$ and $5:06$ early morning, where it takes 4 to 6 minutes and is followed by the sleeping activity almost all the times. The results corresponding to the second cluster indicate that the Bed-Toilet activity takes place in the evening from $21:46$ to $23:22$ and is followed by Sleeping (78%), Bathing (11%) and Taking Medication (11%).

Moreover, results provided in Table 6 suggest the cooking activity is typically performed in 5 clusters within a day. As previously discussed in TEREDA's description, discovering different numbers of clusters for different activities is due to employing the EM clustering algorithm. Similar to the Cooking activity, results from Table 6 suggest that the Eating activity also occurs in 5 main clusters of a day. The observations regarding the Enter Home activity have already been discussed in the main body of the paper, section 5.2.

Table 7 demonstrates results of running TEREDA on our smart home data for the second five activities. The results suggest that TEREDA discovers only one cluster of a day for the occurrence of the Housekeeping activity, when it takes less than 22 minutes and is typically followed by the Leaving Home, Personal Hygiene, and Relaxing activities with the specified confidence values. Furthermore, the discovered results regarding the Personal Hygiene activity indicate that this activity happens mainly in 4 time clusters per day and takes no more than 10 minutes.

Table 7 also demonstrates findings regarding the last three activities (i.e. Relaxing, Sleeping, and Taking Medication). These findings indicate that the Relaxing activity occurs in 7 clusters of a day. As a side note, the Relaxing activity is performed on a couch in the smart home and is typically accompanied with other activities, including Watching TV, Snacking, Eating, etc. In contrast to the Relaxing activity, the Sleeping activity takes place in Bed, generally during nighttime. Regarding the Taking Medication activity, the results suggest that the resident takes her Medication mainly in 4 time clusters. Interestingly, the results convey that the most probable activity following the Taking Medication activity is Cooking, regardless of the time of day at which the Taking Medication activity occurs.

Finally, we compare TEREDA's prediction accuracy against a number of other algorithms for the task of activity label prediction. The other prediction approaches used to evaluate TEREDA are naïve Bayes (NB), Decision Tree (C4.5), Multi-Layer Perceptron (MLP) and Support Vector Machines (SVMs). For the mentioned algorithms, three features were considered to predict the next activity label:

– *Activity label*: the label of the current activity.

Table 6: TEREDA's discovered rules, where *minsup* and *minconf* are both set to 0.1.
(The results are shown for the first 5 ADLs)

	Activity	**Cluster #**	**Start Time (hh:mm)** $[\overline{x}_s - 2s_s \ to \ \overline{x}_s + 2s_s]$	**Duration (h:mm)** $[\overline{x}_d - 2s_d \ to \ \overline{x}_d + 2s_d]$	**Next Activity**	**Conf**
1	Bathing	1	[08:31 – 15:15]	[0:03 – 0:09]	Personal Hygiene	.45
					Eating	.12
					Leave Home	.12
		2	[20:24 – 22:06]	[0:19 – 0:21]	Personal Hygiene	.58
					Sleeping	.13
					Relaxing	.13
2	Bed-Toilet Transition	1	[02:36 – 05:06]	[0:04 – 0:06]	Sleeping	.99
		2	[21:46 – 23:22]	[0:03 – 0:05]	Sleeping	.78
					Bathing	.11
					Taking Medication	.11
3	Cooking	1	[06:28 – 07:18]	[0:02 – 0:06]	Relaxing	.47
					Personal Hygiene	.20
					Taking Medication	.14
		2	[09:58 – 11:58]	[0:06 – 0:12]	Taking Medication	.27
					Relaxing	.27
					Eating	.14
		3	[11:45 – 12:09]	[1:18 – 1:26]	Taking Medication	.42
					Relaxing	.27
		4	[13:44 – 18:02]	[0:21 – 0:43]	Relaxing	.33
					Taking Medication	.24
					Eating	.12
		5	[20:28 – 21:52]	[0:12 – 0:22]	Taking Medication	.31
					Personal Hygiene	.20
					Eating	.18
4	Eating	1	[6:48 – 7:36]	[0:02 – 0:04]	Personal Hygiene	.50
					Cooking	.25
					Leave Home	.13
		2	[8:56 – 10:32]	[0:01 – 0:07]	Relaxing	.21
					Eating	.18
					Leave Home	.18
		3	[12:35 – 15:09]	[0:26 – 0:36]	Taking Medication	.29
					Personal Hygiene	.21
					Eating	.14
		4	[17:34 – 20:44]	[0:19 – 0:25]	Relaxing	.24
					Personal Hygiene	.20
					Taking Medication	.20
		5	[20:16 – 21:18]	[0:05 – 0:09]	Taking Medication	.28
					Relaxing	.26
					Leave Home	.15
5	Enter Home	1	[7:52 – 10:08]	N/A	Relaxing	.41
					Eating	.25
		2	[13:27 – 13:35]	N/A	Cooking	.36
					Relaxing	.17
					Eating	.15
		3	[16:38 – 17:10]	N/A	Cooking	.82
		4	[18:41 – 21:57]	N/A	Taking Medication	.28
					Eating	.27
					Relaxing	.20

– *Activity time of day*: a discretized value of the time when the current activity occurs. Time values are binned into the following ranges: $0-3$, $4-7$, $8-11$, $12-15$, $16-19$, and $20-23$.

Table 7: The discovered temporal relation rules, where *minsup* and *minconf* are both set to 0.1.
(The results are presented for the second 5 ADLs)

	Activity	Cluster #	Start Time (hh:mm) $[\overline{x}_s - 2s_s \ to \ \overline{x}_s + 2s_s]$	Duration (h:mm) $[\overline{x}_d - 2s_d \ to \ \overline{x}_d + 2s_d]$	Next Activity	Conf
6	Housekeeping	1	[13:05 – 20:07]	[0:02 – 0:22]	Leave Home	.38
					Personal Hygiene	.31
					Relaxing	.15
7	Personal Hygiene	1	[6:21 – 7:19]	[0:02 – 0:06]	Personal Hygiene	.35
					Cooking	.27
					Leave Home	.18
		2	[7:23 – 13:01]	[0:01 – 0:05]	Leave Home	.22
					Personal Hygiene	.22
					Cooking	.20
		3	[14:23 – 19:59]	[0:01 – 0:03]	Leave Home	.24
					Relaxing	.22
					Cooking	.14
		4	[21:47 – 22:29]	[0:02 – 0:10]	Relaxing	.79
8	Relaxing	1	[6:41 – 7:21]	[1:09 – 1:41]	Personal Hygiene	.77
		2	[8:40 – 10:54]	[0:23 – 0:47]	Personal Hygiene	.34
					Leave Home	.18
					Relaxing	.17
		3	[12:07 – 12:25]	[0:15 – 0:33]	Relaxing	.33
					Leave Home	.17
					Cooking	.17
		4	[13:11 – 14:45]	[1:40 – 2:50]	Personal Hygiene	.37
					Leave Home	.27
					Eating	.16
		5	[15:26 – 16:08]	[0:10 – 0:52]	Personal Hygiene	.50
					Leave Home	.19
					Relaxing	.11
		6	[17:03 – 17:13]	[0:15 – 0:23]	Personal Hygiene	.43
					Relaxing	.29
					Housekeeping	.14
		7	[20:15 – 21:37]	[0:17 – 0:49]	Leave Home	.26
					Personal Hygiene	.22
					Eating	.17
9	Sleeping	1	[1:56 – 6:06]	[5:39 – 8:25]	Personal Hygiene	.52
					Bed-Toilet Transition	.44
		2	[21:46 – 22:50]	[0:05 – 1:43]	Bed-Toilet Transition	.82
10	Taking Medication	1	[6:44 – 7:46]	[0:01 – 0:03]	Cooking	.39
					Relaxing	.18
					Personal Hygiene	.16
		2	[9:52 – 12:34]	[0:11 – 0:13]	Cooking	.46
					Eating	.15
					Taking Medication	.15
		3	[13:52 – 18:18]	[0:05 – 0:07]	Cooking	.51
					Eating	.13
					Relaxing	.11
		4	[20:33 – 22:01]	[0:02 – 0:04]	Cooking	.38
					Personal Hygiene	.20
					Relaxing	.16

– *Activity day of week*: an integer value ranging from 1 to 7 representing the day of the week in which the current activity happens.

The NB algorithm is a probabilistic classifier based on applying Bayes' theorem that assumes all of the above mentioned features are independent (Nazerfard and Cook 2015). The specific parameters for the other compared algorithms are provided in Table 8.

Table 8: Parameters used for the other compared algorithms

	Parameter	Value
C4.5	Pruning confidence threshold	0.25
	Minimum number of instances per leaf	2
MLP	Learning rate	0.3
	Number of hidden units	2
SVMs	Complexity constant	1
	Kernel type	Polynomial

All prediction approaches were tested using 10-fold cross validation. Each prediction method was trained on nine out of ten groups and tested on the remaining one. The results from all ten permutations were used to obtain significance values and averaged together for acquiring an overall accuracy.[2]

As already discussed, Table 6 and Table 7 demonstrate the details of running TEREDA on our smart home data. Table 9 and Fig. 10 compare the activity label prediction outputs between TEREDA and other discussed algorithms. Results suggest that TEREDA achieves an accuracy of 73.69% with a standard deviation of 4.03 for Apt1 and an accuracy of 64.8% with a standard deviation of 4.64 for Apt2. As compared to TEREDA, NB shows a 11.2% decline for Apt1 and a 12.18% decline in accuracy for Apt2. The prediction results for the C4.5 algorithm suggest a 6.77% accuracy drop for Apt1 and a 9.57% drop for Apt2. Prediction results of the MLP algorithm also indicate a 9.76% drop in accuracy for Apt1 and a 10% accuracy drop for Apt2. Prediction accuracies of SVMs with a quadratic polynomial kernel on our smart apartment testbeds also imply a 11.85% decline for Apt1 and a 11.37% decline for Apt2, compared to TEREDA.

Table 9: Activity label prediction accuracy for TEREDA, NB, C4.5, MLP and SVMs.

	TEREDA	NB	C4.5	MLP	SVMs
Apt1	73.69%	62.49%	66.92%	63.93%	61.84%
Apt2	63.81%	52.62%	55.23%	54.8%	53.43%

Finally, Table 10 provides the confusion matrix of TEREDA predictions for Apt1. The results indicate that our dataset is highly imbalanced. For example, the "Personal Hygiene" activity evidently is over-represented in our dataset, as compared to other activities. Hence, the imbalanced nature of

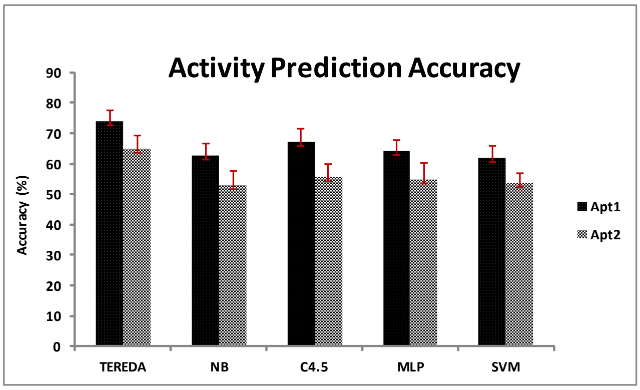

Fig. 10: The overall activity label prediction comparison among discussed approaches.

our dataset is one of the major reasons for particular confusions made by TEREDA. However, there are other possible underlying reasons behind major confusions observed in Table 10, for instance, results suggest that some confusions occur between the "Taking Medication" and "Eating" activities. This confusion is mainly a result of similar location of occurrence. Also, the confusion between the "Cooking" and the "Personal Hygiene" activities is most likely due to the two activities overlapping too often. Most of the remaining confusions occur between two activities that typically happen consecutively, including "Personal Hygiene" and "Leave Home", "Eating" and "Taking Medication", and "Eating" and "Leave Home".

6 Discussion on Scalable TEREDA

A number of work have set up to address the problem of frequent itemset mining in parallel and distributed settings, including work by Moens et al. (2013), Anastasiu et al. (2014), Zhang et al. (2015), and Huynh et al. (2017).

In this regard, studies on parallel programming have mainly followed upon two categories of shared memory and distributed architectures. Shared memory systems are parallel mechanisms in which processes share a single memory address space. Even though implementing parallelism on shared memory systems seems easier, the achieved scalability is not satisfactory (Moens et al. 2013). In the message passing interface (MPI) (Li et al. 2014) processes interact only through direct message passing. Messages are often sent over a network connection. Despite certain advantages in iterative computation, the disadvantages of MPI include its high communication load due to data exchanges between different workstations and lack of fault tolerance mechanism.

MapReduce by Dean and Ghemawat (2008) is a parallel programming framework that provides a relatively simple

[2] Dataset corresponding to Apt2 is also available at http://eecs.wsu.edu/∼nazerfard/AIR/datasets/data2.zip

Table 10: Confusion matrix for the TEREDA predictions

Predicted as ⟶	A	B	C	D	E	F	G	H	I	J	Individual Accuracy
A=Sleeping	**160**	0	5	4	0	0	1	0	2	0	93.02%
B=Bed-Toilet Transition	0	**104**	11	0	0	0	0	0	0	0	90.43%
C=Personal Hygiene	10	22	**513**	1	22	8	23	0	5	9	83.69%
D=Taking Medication	3	0	31	**8**	23	0	5	0	1	3	10.81%
E=Cooking	4	0	26	5	**150**	9	17	0	5	5	67.87%
F=Eating	0	0	7	0	2	**174**	3	0	0	0	93.55%
G=Leave Home	3	0	24	1	3	5	**123**	0	5	2	74.10%
H=Enter Home	1	0	0	0	0	0	0	**164**	0	0	99.39%
I=Housekeeping	0	0	10	1	3	0	2	0	**16**	1	48.48%
J=Relaxing	1	0	8	0	1	0	4	0	0	**24**	63.16%

programming interface. Computation in MapReduce consists of two main phases: *map* and *reduce*. The problem input is specified as a set of *key-value* pairs. In the *map* phase, each mapper processes a distinct split of data and generates *key-value* pairs. During *reduce* phase, the *key-value* pairs produced in the *map* phase are grouped by *key* and fed to *reducers* as pairs of *key-value* lists. The *reducers* further process these intermediate parts of information to produce the final output. The MapReduce framework proves to be an efficient platform for parallel and distributed data mining of large scale datasets. However, it is not appropriate for iterative computation, since repeated read/write operations to its distributed file system would lead to high I/O load and time cost.

To overcome the above-mentioned problems, we propose the *Apache Spark* platform by Zaharia et al. (2010), a memory-based distributed framework, as a solution architecture for the parallel and scalable version of TEREDA for future directions.

7 Conclusions and Future Work

In this paper, we introduced TEREDA to discover the temporal relations of the activities of daily living. The proposed approach is based on association rule mining and the EM clustering techniques. TEREDA also discovers usual start time and duration of activities as a mixture normal model.

One of the technologies that has received increasing attention over the past few years is the Internet of Things (IoT). IoT is considered the technology of seamlessly integrating classical networks and networked objects (Miorandi et al. 2012). One of the most important questions that arises in this new emerging technology is how to convert the massive data captured by IoT into knowledge to provide more convenient environments for people (Tsai et al. 2014). In future, we plan to develop distributed and scalable TEREDA, using Spark, and employ it to provide possible solutions to discover hidden information in the IoT big data.

Acknowledgement

The author would like to thank D. J. Cook and P. Rashidi for their thorough comments and suggestions on this work.

References

Abowd G and Mynatt E. (2004). Designing for the human experience in smart environments. In *Cook D, Das S, editors. Smart Environments: Technology, Protocols, and Applications*, pages 153–174, Wiley.

Agrawal R and Srikant R. (1994). Fast algorithms for mining association rules in large databases. In *VLDB*, pages 487–499.

Agrawal R, Imielinski T, and Swami A. (1993). Mining associations between sets of items in large databases. In *ACM SIGMOD International Conference on Management of Data*, pages 207–216.

Anastasiu D, Iverson J, Smith S, and Karypis G. (2014). Big data frequent pattern mining. In *Frequent Patten Mining*, pages 225–259.

Boger J, Poupart P, Hoey J, Boutilier C, Fernie G, and Mihailidis A. (2005). A decision-theoretic approach to task assistance for persons with dementia. In *In Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1293–1299.

BrainAid. (2013). PEAT: Android application for people with cognitive challenges [online]. http://brainaid.com, visited september 25.

Brdiczka O, Maisonnasse J, and Reignier P. (2005). Automatic detection of interaction groups. In *Proceedings of the international conference on Multimodal interfaces*, pages 32–36.

Catarinucci L, Donno D, Mainetti L, Palano L, Patrono L, Stefanizzi M, and Tarricone L. (2015). An IoT-aware architecture for smart healthcare systems. *IEEE Internet of Things*, 2(6):515–526.

Chen M, Ma Y, Li Y, Wu D, Zhang Y, and Youn C.-H. (2017). Wearable 2.0: Enabling human-cloud integration in next generation healthcare systems. *IEEE Communications Magazine*, 55(1):54–61.

Cook D, Youngblood M, Heierman-III E, Gopalratnam K, Rao S, Litvin A, and Khawaja F. (2003). MavHome: an agent-based smart home. In *IEEE International Conference on Pervasive Computing and Communications*, pages 521–524.

Cook D, Crandall A, Thomas B, and Krishnan N. (2013). CASAS: A smart home in a box. *IEEE Computer*, 46(6):26–33.

Crandall A. (2011). *Behaviometrics for Multiple Residents in a Smart Environment, Ph.D. Dissertation*. Washington State University.

Dean J and Ghemawat S. (2008). Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.

Doctor F, Hagras H, and Callaghan V. (2005). A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 35(1):55–65.

Dufkova K, Kencl L, and Bjelica M. (2009). Predicting user-cell association in cellular networks from tracked data. In *AAAI Spring Symposium on Intelligent Environments*.

Gopalratnam K and Cook D. (2004). Active lezi: An incremental parsing algorithm for sequential prediction. *International Journal of Artificial Intelligence Tools*, 14(1-2):917–930.

Gopalratnam K and Cook D. (2007). Online sequential prediction via incremental parsing: The active lezi algorithm. *IEEE Intelligent Systems*, 22(1):52–58.

Gu T, Wang L, Wu Z, and Tao X. (2011). A pattern mining approach to sensor-based human activity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 23(9):1359–1372.

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, and Witten I. (2009). The weka data mining software: An update. *SIGKDD Exploration*, 11(1).

Hamm J, Stone B, Belkin M, and Dennis S. (2013). Automatic annotation of daily activity from smartphone-based multisensory streams. In *Mobile Computing, Applications, and Services*, pages 328–342.

Han J, Pei J, and Yin Y. (2000). Mining frequent patterns without candidate generation. In *International Conference on Management of Data (SIGMOD)*, pages 1–12.

Helal S, Mann W, El-Zabadani H, King J, Kaddoura Y, and Jansen E. (2005). The Gator Tech Smart House: A programmable pervasive space. *IEEE Computer Society*, 38(3):50–60.

Heung-II S, Bong-Kee S, and Seong-Whan L. (2010). Hand gesture recognition based on dynamic bayesian network framework. *Pattern recognition*, 43:3059–3072.

Hossain M and Muhammad G. (2016). Cloud-assisted industrial internet of things (IIoT) enabled framework for health monitoring. *Computer Networks*, 101:192–202.

Huynh B, Vo B, and Snasel V. (2017). An efficient method for mining frequent sequential patterns using multi-core processors. *Journal of Applied Intelligence*, 46(3):703–716.

Intille S, Larson K, Tapia E, Beaudin J, Kaushik P, Nawyn J, and Rockinson R. (2006). Using a live-in laboratory for ubiquitous computing research. In *Proceedings of PERVASIVE*, pages 349–365.

Kasteren T and Krose B. (2007). Bayesian activity recognition in residence for elders. In *International Conference on Intelligent Environments*, pages 209–212.

Kaushik P, Intille S, and Larson K. (2008). User-adaptive reminders for home-based medical tasks. a case study. *Methods of Information in Medicine*, 47(2):203–207.

Krishnan N and Cook D. (2014). Activity recognition on streaming sensor data. *Pervasive and Mobile Computing*, 10:138–154.

Li S, Hoefler T, Hu C, and Snir M. (2014). Improvedmpi collectives for mpi processes in shared address spaces. *Cluster Computing*, 14(4):1139–1155.

Li Y, Ning P, Wang X, and Jajodia S. (2001). Discovering calendar-based temporal association rules. *Data and Knowledge Engineering*, 44(2):193–218.

Lim M, Choi J, Kim D, and Park S. (2008). A smart medication prompting system and context reasoning in home environments. In *Proceedings of the Fourth IEEE International Conference on Networked Computing and Advance Information Management*, pages 115–118.

Lotfi A, Langensiepen C, Mahmoud S, and Akhlaghinia M. (2012). Smart homes for the elderly dementia sufferers: Identifcation and prediction of abnormal behaviour. *Journal of Ambient Intelligence and Humanized Computing (JAIHC)*, 3(3):205–218.

Mahmoud S, Lotfi A, and Langensiepen C. (2013). Behavioural pattern identification and prediction in intelligent environments. *Journal of Applied Soft Computing*, 13(4):1813–1822.

Maurer U, Smailagic A, Siewiorek D, and Deisher M. (2006). Activity recognition and monitoring using multiple sensors on different body positions. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, pages 113–116.

Medjahed H, Istrate D, Boudy J, and Dorizzi B. (2009). Human activities of daily living recognition using fuzzy logic for elderly home monitoring. In *IEEE International Conference on Fuzzy Systems*, pages 2001–2006.

Minor B and Cook D. (2017). Forecasting occurrences of activities. *Pervasive and Mobile Computing*, 38(1):77–91.

Miorandi D, Sicari S, Pellegrini F. D, and Chlamtac I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516.

Mocanu I and Florea A. (2012). A multi-agent supervising system for smart environments. In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, WIMS, pages 1–55.

Mocanu S, Mocanu I, Anton S, and Munteanu C. (2011). Amihomecare: A complex ambient intelligent system for home medical assistance. In *Proceedings of the International Conference on Applied Computer and Applied Computational Science*, pages 181–186.

Moens S, Aksehirli E, and Goethals B. (2013). Frequent itemset mining for big data. In *IEEE International Conference on Big Data*, pages 111–118.

Nazerfard E and Cook D. (2015). CRAFFT: An activity prediction model based on bayesian networks. *Journal of Ambient Intelligence and Humanized Computing (AIHC)*, 6(2):193–205.

Nazerfard E, Rashidi P, and Cook D. (2010). Discovering temporal features and relations of activity patterns. In *Proceedings of the ICDM Workshop on Data Mining for Service (DMS)*, pages 1069–1075.

O'Donovan T, O'Donoghue J, Sreenan C, Sammon D, O'Reilly P, and O'Connor K. (2009). A context aware wireless body area network (BAN). In *Pervasive Computing Technologies for Healthcare*, pages 1–8.

Pineau J, Montemerlo M, Pollack M, Roy N, and Thrun S. (2003). Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems*, 42(3-4):271–281.

Pollack M, Brown L, Colbry D, McCarthy C, Orosz C, Peintner B, Ramakrishnan S, and Tsamardinos I. (2003). Autominder: An intelligent cognitive orthotic system for people with memory impairment. *Robotics and Auotnomous Systems*, 44(3-4):273–282.

Rudary M, Singh S, and Pollack M. (2004). Adaptive cognitive orthotics: Combining reinforcement learning and constraint-based temporal reasoning. In *Proceeding of International Conference on Machine Learning*, pages 91–98.

Singla G, Cook D, and Schmitter-Edgecombe M. (2009). Tracking activities in complex settings using smart environment technologies. *International Journal of BioScience, Psychiatry and Technology*, 1(1):25–35.

Tapia D, Abraham A, Corchado J, and Alonso R. (2010). Agents and ambient intelligence: case studies. *Journal of Ambient Intelligence and Humanized Computing*, 1(2):85–93.

Tsai C.-W, Lai C.-F, Chiang M.-C, and Yang L. (2014). Data mining for internet of things: A survey. *IEEE Communications Services and Tutorials*, 16(1):77–97.

Vail D, Veloso M, and Lafferty J. (2007). Conditional random fields for activity recognition. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1–8.

Weber J and Pollack M. (2007). Entropy-driven online active learning for interactive calendar management. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 141–149.

Yin J, Yang Q, and Pan J. (2008). Sensor-based abnormal human-activity detection. *IEEE Transactions on Knowledge and Data Engineering*, 20(8):1082–1090.

Zaharia M, Chowdhury M, Franklin M, Shenker S, and Stoica I. (2010). Spark: cluster computing with working sets. In *Proceedings of the USENIX conference on Hot topics in cloud computing*, pages 10–10.

Zhang F, Liu M, Gui F, Shen W, Shami A, and Ma Y. (2015). Scalable algorithms for association mining. *Journal of Cluster Computing*, 18(4):1493–1501.