Mining Sensor Streams for Discovering Human Activity Patterns Over Time

Parisa Rashidi
EECS Department
Washington State University
Pullman, US
Email: prashidi@wsu.edu

Diane J. Cook
EECS Department
Washington State University
Pullman, US
Email: cook@eecs.wsu.edu

Abstract—In recent years, new emerging application domains have introduced new constraints and methods in data mining field. One of such application domains is activity discovery from sensor data. Activity discovery and recognition plays an important role in a wide range of applications from assisted living to security and surveillance. Most of the current approaches for activity discovery assume a static model of the activities and ignore the problem of mining and discovering activities from a data stream over time. Inspired by the unique requirements of activity discovery application domain, in this paper we propose a new stream mining method for finding sequential patterns over time from streaming non-transaction data using multiple time granularities. Our algorithm is able to find sequential patterns, even if the patterns exhibit discontinuities (interruptions) or variations in the sequence order. Our algorithm also addresses the problem of dealing with rare events across space and over time. We validate the results of our algorithms using data collected from two different smart apartments.

Index Terms—Activity Data Mining; Smart Environments; Sensor Data; Stream Sequence Mining;

I. INTRODUCTION

In recent years, different emerging application domains have introduced new constraints and methods for data mining. One such application domain is activity discovery and recognition in smart environments. Due to the increasing aging population and the rising demand for in-place monitoring, smart environments have attracted many researchers from different disciplines. A smart environment refers to an environment which is equipped with different types of sensors, such as infrared motion sensors, contact switch sensors, RFID tags, power-line controllers, etc. The low level sensor data obtained from different sensors is mined and analyzed to detect residents' activity patterns. Recognizing residents' activities and their daily routines can greatly help in providing automation, security, and more importantly in remote health monitoring of elderly, or people with disabilities. For example, by monitoring the daily routines of a person with dementia, the system can track how completely and consistently the daily routines are performed. It also can determine when the resident needs assistance or raise an alarm if needed.

A variety of supervised methods have already been proposed for human activity recognition, e.g. neural networks [1], naive Bayes [2], conditional random fields [3], decision trees [4], Markov models [5], and dynamic Bayes networks [6]. In a real world situation, using supervised methods is not very practical, as it requires labeled data for training. Manual labeling of human activity data is time consuming, laborious and error-prone. Besides, one usually needs to deploy invasive devices in the environment during data collection phase to obtain reliable annotations. Another option is to ask the residents to report their activities. Asking the residents to report their activities puts the burden on the residents, and in case of elderly with memory problems such as dementia it would be out of question.

In an effort to address the annotation problem, recently a few unsupervised methods have been proposed for mining human activity data. Those methods include frequent sensor mining [7], mining discontinuous activity patterns [8], mining mixed frequent-periodic activity patterns [9], detecting activity structure using low dimensional eigenspaces [10], and discovering discontinuous varied order patterns [11]. None of these mining approaches take into account the streaming nature of data, nor the possibility that the patterns might change over time. In a real world situation, in a smart environment we have to deal with a potentially infinite and unbounded flow of data. Also the discovered activity patterns can change over time. Mining the stream of data over time not only allows us to find new emerging patterns in the data, but it also allows us to detect changes in the patterns. Detecting changes in the patterns can be beneficial for many applications. For example a caregiver can look at the pattern trends over time and spot any suspicious changes immediately.

In the last decade, many stream mining methods have been proposed as a result of different emerging application domains, such as network traffic analysis, Web click stream mining, and power consumption measurements. Most of the proposed methods try to find frequent itemsets over data streams [12]–[15]. Also some methods have been proposed for finding frequent sequences over data streams [16]–[18]. To the best of our knowledge, no stream mining method has been proposed so far for mining human activity patterns from sensor data in the context of smart environments.

In this paper, we extend the tilted-time window approach proposed by Giannella et al. [13], in order to discover activity pattern sequences over time. The tilted-time window approach finds the frequent itemsets using a set of tilted-time windows,

such that the frequency of the item is kept at a finer level for recent time frames and at a coarser level for older time frames. Such a tilted window approach can be quite useful for human activity pattern discovery. For example a caregiver is usually interested in the recent changes of the patient at a finer level, and in the older patterns (e.g. from three months ago) at a coarser level.

Due to the special requirements of our application domain, we cannot directly use the method proposed by Giannella et al. [13]. First of all, the time-tilted approach [13], as well as most of the other similar stream mining methods [16]–[18] were designed to find sequences or itemsets in transaction-based streams. The data obtained in smart environment is a continuous stream of unbounded sensor events with no boundary between episodes or activities. Second, as discussed in [11], due to the complex and erratic nature of human activity, we need to consider an activity pattern as a sequence of events. In such a sequence, the patterns might be interrupted by irrelevant events (called a discontinuous pattern). Also the order of events in the sequence might change from occurrence to occurrence (called a varied order pattern). Finding variations of a general pattern and determining their relative importance can be beneficial in many applications. For example in a study, Hayes, et al. [19] found that variation in the overall activity performance at home was correlated with mild cognitive impairment. This highlights the fact that it is important to recognize and monitor all the activities and their variations which are performed regularly by an individual in a daily environments. Third, we also need to address the problem of varying frequencies for activities performed in different regions of the space. A person might spend the majority of his/her time in the living room during the day, and only go to the bedroom for sleeping. We still need to discover the sleep pattern though its sensor support count might be substantially less than the support count of the sensors in the living room.

In this paper, we extend the DVSM method proposed by Rashidi and Cook [11] into a streaming version based on using a tilted-time window [13]. Our proposed method allows us to find discontinuous varied-order patterns in streaming non transaction sensor data over time. As another extension to this work [11], we also address the problem of varying frequencies to better handle real life data and to find a higher percentage of the interesting patterns over time. This represents the first reported stream mining method for discovering human activity patterns in sensor data over time. Besides activity mining, our discontinuous and varied order stream mining method can be useful in other application domains, where different variations of a pattern can reveal useful information, such as Web click mining.

The remainder of the paper is organized as follows. First we explain the related stream mining works in more detail in section II. Next we describe the titled-time window in more detail in section III. Our proposed solution is explained in section IV. We then show the results of our experiments on data obtained from two different smart apartments in section V. Finally we end the paper with our conclusions and discussion of future work.

II. STREAM MINING RELATED WORKS

Sequential pattern mining has been studied for more than a decade [20] and many methods have been proposed for finding sequential patterns in data [20]–[23]. Compared to the classic problem of mining sequential patterns from a static database, mining sequential patterns over data streams is far more challenging. In a data stream, new elements are generated continuously and no blocking operation can be performed on the data. Despite being more challenging, with the rapid emergence of new application domains over the past few years the stream mining problem has also been studied in a wide range of different application domains. A few such application domains include network traffic analysis, fraud detection, Web click stream mining, power consumption measurements and trend learning [24].

For finding patterns in a data stream, approximation and using a relaxed support threshold is a key concept [15], [25]. The first approach was introduced by Manku et al. [15] based on using a landmark model and calculating the count of the patterns from the start of the stream. Later Li et al [14] proposed methods for incorporating the discovered patterns into a prefix tree. They also designed methods for regression-based stream mining algorithms [26]. More recent approaches have introduced methods for managing the history of items over time [13], [26]. The main idea is that one usually is more interested in recent changes in more detail, while older changes are preferred in coarser granularity in long term.

There also have been several methods for finding sequential patterns over data streams, including the SPEED algorithm [18], methods for finding approximate sequential patterns in Web usage [17], a data cubing algorithm [27] and mining multidimensional sequential patterns over data streams [28]. All of these approaches consider data to be in a transactional format. However, input data stream in a smart environment is a continuous flow of unbounded data. Figure 1 depicts the difference between transaction data and sensor data. As can be seen in Figure 1a, for transaction data, each single transaction is associated with a set of items and is identified by a transaction ID, making it clearly separated from the next transaction. The sensor data has no boundaries separating different activities or episodes from each, and it is just a continuous stream of sensor events over time.

Approaches proposed by sensor stream mining community [29], [30] try to turn a sensor stream into a transactional dataset using techniques such as Apriori technique [20] to group frequent events together. Another method is to simply use fixed or varied clock ticks [30]. In our scenario, using such simple techniques does not allow us to deal with complex activity patterns that can be discontinuous, varied order, and of arbitrary length. Instead, we extend the DVSM method [11] to group together the co-occurring events into varied-order discontinuous activity patterns.

It should be noted that researchers from the ubiquitous computing and smart environment community have also pro-

Transaction ID	Items
1	$\{A, B, D, F, G, H\}$
2	$\{D, F, G, H, X\}$
3	$\{A, B, C, X, Y\}$

(a)
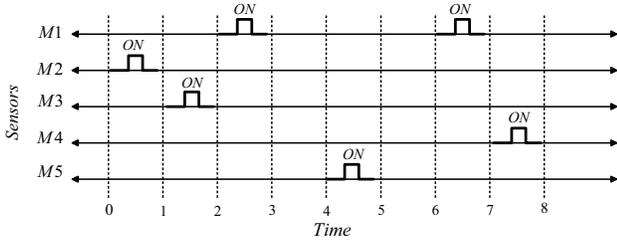
$$M2 - M3 - M1 - M5 - M1 - M4 - ...$$

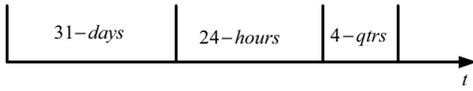(b)

Fig. 1: Transaction data vs. sensor data.

Fig. 2: Natural tilted-time window.

posed methods for finding patterns from sensor data [7]–[11]. However none of these approaches address the stream mining problem. To the best of our knowledge this is the first work addressing activity pattern discovery from sensor data over time.

III. TILTED-TIME WINDOW MODEL

In this section, we explain the tilted window model [13] in more detail. Figure 2 shows an example of a natural tilted-time window where the frequency of the most recent item is kept with an initial precision granularity of an hour, in another level of granularity in the last 24 hours and then again at another level in the last 31 days. As new data items arrive over time, the history of items will be shifted back in the tilted-time window to reflect the changes. Other variations such as logarithmic tilted-time window have also been proposed to provide a more efficient storage [13].

The tilted-time window model uses a relaxed threshold to find patterns according to the following definition.

Definition 1. *Let the minimum support be denoted by σ, and the maximum support error be denoted by ϵ. An itemset I is said to be frequent if its support is no less than σ. If support of I is less than σ, but no less than $\sigma - \epsilon$, it is sub-frequent; otherwise it is considered to be infrequent.*

Using the approximation approach for frequencies allows for the sub-frequent patterns to become frequent later, while discarding infrequent patterns. To reduce the number of frequency records in the tilted-time windows, the old frequency records of an itemset I are pruned. Let $\bar{f}_j(I)$ denote the computed frequency of I in time unit j, and let N_j denote

TABLE I: Example sensor data.

Timestamp (t)	Sensor ID (s)
7/17/2009 09:52:25	$M4$
7/17/2009 09:56:55	$M30$
7/17/2009 14:12:20	$M15$

the number of transactions received within time unit j. Also let τ refer to the most recent time point. For some m where $1 \le m \le \tau$, the frequency records $\bar{f}_1(I), ..., \bar{f}_m(I)$ are pruned if Equation 1 and Equation 2 hold [31].

$$\exists n \le \tau, \ \forall i, 1 \le i \le n, \ \bar{f}_i(I) < \sigma N_i \qquad (1)$$

$$\forall l, 1 \le l \le m \le n, \ \sum_{j=1}^{l} \bar{f}_j(I) < (\sigma - \epsilon) \sum_{j=1}^{l} N_i \qquad (2)$$

Equation 1 finds a point n in the stream such that before that point, the computed frequency of the itemset I is always less than the minimum frequency required. Equation 2 finds a point m, where $1 \le m \le n$, such that before that point, the sum of the computed support of I is always less than the relaxed minimum support threshold. In this case the frequency records of I from 1 to m are considered as unpromising and are pruned. This type of pruning is referred to as "tail pruning". In our model, we will extend the above defections and pruning techniques for discontinuous, varied order patterns.

IV. PROPOSED MODEL

In the following subsections, first we give an overview of definitions and notations, then we will describe our model in more detail.

A. Definitions

The input data in our model is an unbounded stream of sensor events, each in the form of $e = \langle s, t \rangle$, where s refers to a sensor ID and t refers to the timestamp when sensor s has been activated. Table I shows an example of several such sensor events. We define an activity instance as a sequence of n sensor events $\langle e_1, e_2, .., e_n \rangle$. Note that in out notations an activity instance is considered as a sequence of sensor events, not a set of unordered events.

We assume that the input data is broken into batches $B_{a_1}^{b_1}...B_{a_n}^{b_n}$ where each $B_{a_i}^{b_i}$ is associated with a time period $[a_i..b_i]$, and the most recent batch is denoted by $B_{a_\tau}^{b_\tau}$ or for short as B_τ. Each batch $B_{a_i}^{b_i}$ contains a number of sensor events, denoted by $|B_{a_i}^{b_i}|$.

As we mentioned before, we use a time tilted-time window for maintaining the history of patterns over time. Instead of maintaining the frequency records, we maintain the "compression" records, which will be explained in more detail in section IV-B. Also, in our model as the frequency of an item is not the single deciding factor, and other factors such as the length of the pattern and its continuity also play a role, we will use the term "interesting" pattern instead of a "frequent" pattern.

Month	· · · · ·	Month	Week	Week	Week	Week

$\longrightarrow$ t

Fig. 3: Our tilted-time window.

The tilted-time window used in our model is depicted in Figure 3. This tilted-time window keeps the history records of a pattern during the past 4 weeks at the finer level of week granularity. History records older than 4 weeks are only kept at the month granularity. Regarding our application domain and considering its end users, a natural time-tilted window provides a more natural representation vs. a logarithmic tilted-time window. For example it would be it easier for a nurse or caregiver to interpret the pattern trend using a natural representation. Second, as we don't expect the activity patterns to change substantially over a very short period of time, we omit the day and hour information for the sake of a more efficient representation. For example, in case of dementia patients it takes weeks and months to see some changes to develop in their daily activity patterns. Using such a schema we only need to maintain 15 compression records instead of $365 * 24 * 4$ records in a normal natural tilted-time window keeping day and hour information. Note that we chose such a representation in this study for the reasons mentioned above. However if necessary, one can adopt other tilted-window models such as the logarithmic windows, as the choice of titled-time window has no effect on the model except for efficiency.

To update the tilted-time window, whenever a new batch of data arrives, we will replace the compression values at the finest level of time granularity and shift back to the next level of finest time granularity. During shifting, we check if the intermediate window is full. If so, the window is shifted back even more; otherwise the shifting stops.

B. Mining Activity Patterns

Our goal is to develop a method that can automatically discover resident activity patterns over time from streaming sensor data. Even if the patterns are somehow discontinuous or have different event orders across their instances. Both situations happen quite frequently while dealing with human activity data. For example, consider the "meal preparation activity". Most people will not perform this activity in exactly the same way each time, rather some of the steps or the order of the steps might be changed (variations). In addition the activity might be interrupted by irrelevant events such as answering the phone (discontinuous). The DVSM method proposed in [11] finds such patterns in a static dataset. For example, the pattern $\langle a, b \rangle$ can be discovered from instances $\{b, x, c, a\}$, $\{a, b, q\}$, and $\{a, u, b\}$, despite the fact that the events are discontinuous and have varied orders.

We discover the sequential patterns from the current data batch B_τ by using an extended version of DVSM that is able to find patterns in streaming data and is also able to deal with varying frequencies across different regions of a physical space. After finding patterns in current data batch B_τ, we will update the tilted-time windows, and will prune any pattern that seems to be unpromising.

To find patterns in data, first a reduced batch B_τ^r is created from the current data batch B_τ. The reduced batch contains only frequent and subfrequent sensor events, which will be used for constructing longer patterns. A minimum support is required to identify such frequent and subfrequent events. DVSM uses as global minimum support, and it only identifies the frequent events. Here, we introduce the maximum sensor support error ϵ_s to allow for the subfrequent patterns to be also discovered. We will also automatically derive multiple minimum supports values corresponding to different regions of the space.

In mining real life activity patterns, the frequency of sensor events can vary across different regions of the home or other space. If the differences in sensor event frequencies across different regions of the space are not taken into account, the patterns that occur in less frequently used areas of the space might be ignored. For example, if the resident spends most of his/her time in the living-room during the day and only goes to the bedroom for sleeping, then the sensors will be triggered more frequently in the living-room than in the bedroom. Therefore when looking for frequent patterns, the sensor events in the bedroom might be ignored and consequently the sleep pattern might not be discovered. The same problem happens with different types of sensors, as usually the motion sensors are triggered much more frequently than other type of sensors such as cabinet sensors. This problem is known as "rare item" problem in market basket analysis and is usually addressed by providing multiple minimum support values [32]. Our proposed solution for solving the problem of rare items in data stream can be applied to other application domains, such as Web click mining. For example, a web page might have a lower chance of being visited by visitors, but we still might be interested in finding click patterns in such pages.

We will automatically derive multiple minimum sensor support values across space and over time. To do this, we identify different regions of the space using location tags l, corresponding to the functional areas such as bedroom, bathroom, etc. Also different types of sensor might exhibit varying frequencies. In our experiments, we categorized the sensors into motion sensors and interaction-based sensors. The motion sensors are those sensors tracking the motion of a person around a home, e.g. infrared motion sensors. Interaction-based sensors, as we will call them "key sensors", are the non-motion tracking sensors, such as cabinet sensors or RFID tags on items. Based on observing and analyzing sensor frequencies in multiple smart homes, we found that a motion sensor might have a higher chance of being triggered than a key sensor in some regions. Hence we will derive separate minimum sensor supports for different sensor categories.

For the current data batch B_τ, we compute the minimum regional support for different categories of sensors as in Equation 3. Here l refers to a specific location. c refers to the sensor's category, and $\mathcal{S}_c$ refers to the set of sensors in a
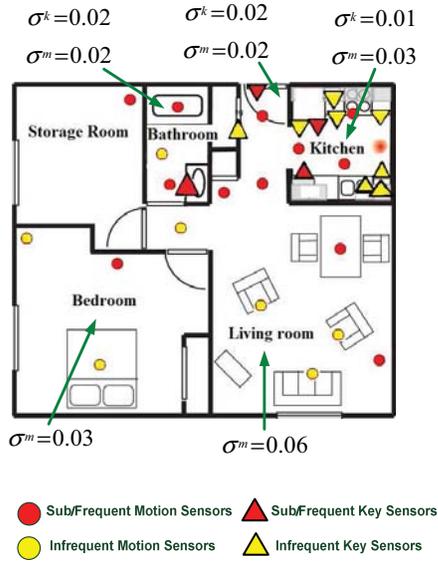
Fig. 4: The frequent/subfrequent sensors are selected based on the minimum regional support, instead of a global support.

category c. Also $f_T(s)$ refers to the frequency of a sensor s over a time period T.

$$\sigma_T^c(l) = 1/\sum_{s\in\mathcal{S}_c^l}^{|\mathcal{S}_c^l|} f_T(s) \quad s.t. \quad \mathcal{S}_c^l = \{s\,|\,s \in l \wedge s \in \mathcal{S}_c\} \quad (3)$$

As an illustrative example, Figure 4 shows the computed minimum regional sensor support values for a smart home used in our experiments.

Using the minimum regional sensor frequencies, frequent and subfrequent sensors are defined as following.

Definition 2. *Let s be a sensor of category c located in location l. The frequency of s over a time period T, denoted by $f_T(s)$, is the number of times in time period T in which s occurs. The support of s in location l and over time period T is $f_T(S)$ divided by the total number of sensor events of the same category occurring in L during T. Let ϵ_s be the maximum sensor support error. Sensor s is said to be frequent if its support is no less than $\sigma_T^c(l)$. It is sub-frequent if its support is less than $\sigma_T^c(l)$, but no less than $\sigma_T^c(l) - \epsilon_s$; otherwise it is infrequent.*

Only the sensor events from the frequent and subfrequent sensors will be added to the reduced batch B_τ^r, which is then used for constructing longer sequences. We use a pattern growth method as in [9] which grows a pattern by its prefix and suffix. To account for the variations in the patterns, the concept of a general pattern is introduced. A general pattern is a set of all of the variations that have a similar structure in terms of comprising sensor events, but have different event orders [11]. During pattern growth, if an already discovered variation matches a newly discovered of pattern, its frequency and continuity information will be updated. If the newly discovered

pattern matches the general pattern, but does not exactly match any of the variations, it is added as a new variation. Otherwise it will be considered as a new general pattern.

At the end of each pattern growth iteration, infrequent or highly discontinuous patterns and variations will be discarded as uninteresting patterns. Instead of solely using a pattern's frequency as a measure of interest, we use a compression objective based on the minimum description length (MDL) [33]. Using a compression objective allows us to take into account the ability of the pattern to compress a dataset with respect to pattern's length and continuity. The compression value of a general pattern a over a time period T is defined as in Equation 4. The compression value of a variation a_i of a general pattern over a time period T is defined as in Equation 5. Here L refers to the description length as defined by Minimum Description length principle (MDL), and Γ refers to continuity as defined in [11]. Continuity basically shows how contiguous the component events of a pattern or a variation are. It is computed in a bottom-up manner, such that for the variation continuity is defined in terms of the average number of infrequent sensor events separating each two successive events of the variation. For a general pattern continuity is defined as the average continuity of its variations.

$$\alpha_T(a) = \frac{L(B_T) * \Gamma_a}{L(a) + L(B_T|a)} \quad (4)$$

$$\beta_T(a_i) = \frac{(L(B_T|a) + L(a)) * \Gamma_{a_i}}{L(B_T|a_i) + L(a_i)} \quad (5)$$

Variation compression measures the capability of a variation to compress a general pattern compared to the other variations. Compression of a general pattern shows the overall capability of the general pattern to compress the dataset with respect to its length and continuity. Based on using the compression values and by using a maximum compression error, we define interesting, sub-interesting and uninteresting patterns and variations.

Definition 3. *Let the compression of a general pattern a be defined as in Equation 4 over a time period T. Also Let σ_g and ϵ_g denote the minimum compression and maximum compression error. The general pattern a is said to be interesting if its compression α is no less than σ_g. It is sub-interesting if its compression is less than σ_g, but no less than $\sigma_g - \epsilon_g$; otherwise it is uninteresting.*

We also give a similar definition for identifying interesting/sub-interesting variations of a pattern. Let the average variation compression of all variations of a general pattern a over a time period T be defined as in Equation 6. Here the number of variations of a general pattern is denoted by n_a.

$$\tilde{\beta}_T(a) = \frac{1}{n_a} * \sum_{i=1}^{n_a} \frac{(L(B_T|a) + L(a)) * \Gamma_{a_i}}{L(B_T|a_i) + L(a_i)} \quad (6)$$

Definition 4. *Let the compression of a variation a_i of a general pattern a be defined as in Equation 5 over a time period*

T. Also Let ϵ_v denote the maximum variation compression error. A variation a_i is said to be interesting over a time period T if its compression $\beta_T(a_i)$ is no less than $\tilde{\beta}(a)_T$. It is sub-interesting if its compression is less than $\tilde{\beta}(a)_T$, but no less than $\tilde{\beta}(a)_T - \epsilon_v$; otherwise it is uninteresting.

During each pattern growth iteration, based on the above definitions, the uninteresting patterns and variation are pruned, i.e. those patterns and variations that are either highly discontinuous or infrequent (with respect to their length). We also prune redundant non-maximal general patterns; i.e., those patterns that are totally contained in another larger pattern. To only maintain the very relevant variations of a pattern, also irrelevant variations of a pattern are discarded based on using mutual information [34] as in Equation 7. It allows us to find core sensors for each general pattern a. Finding the set of core sensors allows us to prune the irrelevant variations of a pattern which do not contain the core sensors. Here $P(s, a)$ is the joint probability distribution of a sensor s and general pattern a, while $P(s)$ and $P(a)$ are the marginal probability distributions. A high mutual information value indicates the sensor is a core sensor.

$$MI(s, a) = P(s, a) * \log \frac{P(s, a)}{P(s)P(a)} \qquad (7)$$

We continue extending the patterns by prefix and suffix at each iteration until no more interesting patterns are found. A post-processing step records attributes of the patterns, such as event durations and start times. We refer to the pruning process performed during the pattern growth on the current data batch as normal pruning. Note that it's different from the tail pruning process which is performed on the titled-time window to discard the unpromising patterns over time.

In the following subsection we will describe how the tilted-time window is updated after discovering patterns of current data batch.

C. Updating the Tilted-time Window

After discovering the patterns in the current data batch as described in the previous subsection, the tilted-time window will be updated. Each general pattern is associated with a tilted-time window. The tilted-time window keeps track of the general pattern's history as well as its variations. Whenever a new batch arrives, after discovering its interesting general patterns, we will replace the compressions at the finest level of granularity with the recently computed compressions. If a variation of a general pattern is not observed in the current batch, we will set its recent compression to 0. If none of the variations of a general patterns are perceived in the current batch, then the general pattern's recent compression is also set to 0.

In order to reduce the number of maintained records and to remove unpromising general patterns, we propose the following tail pruning mechanisms as an extension of the original tail pruning method in [13]. Let $\alpha_j(a)$ to denote the computed compression of general pattern a in time unit j.

Also let τ refer to the most recent time point. For some m, where $1 \le m \le \tau$, the compression records $\alpha_1(a), ..., \alpha_m(a)$ are pruned if Equations 8 and 9 hold.

$$\exists n \le \tau, \ \forall i, 1 \le i \le n, \ \alpha_i(a) < \sigma_g \qquad (8)$$

$$\forall l, 1 \le l \le m \le n, \ \sum_{j=1}^{l} \alpha_j(a) < l * (\sigma_g - \epsilon_g) \qquad (9)$$

Equation 8 finds a point n in the stream such that before that point, the computed compression of the general pattern a is always less than the minimum compression required. Equation 9 computes the time unit m, where $1 \le m \le n$, such that before that point, the sum of the computed compression of a is always less than the relaxed minimum compression threshold. In this case the compression records of a from 1 to m are considered as unpromising and are pruned.

We define a similar procedure for pruning the variations of a general pattern. We prune a variation a_k if the following conditions in Equations 10 and 11 hold.

$$\exists n \le \tau, \ \forall i, 1 \le i \le n, \ \beta_i(a_k) < \tilde{\beta}_i(a) \qquad (10)$$

$$\forall l, 1 \le l \le m \le n, \ \sum_{j=1}^{l} \beta_j(a_k) < l * (\tilde{\beta}_l(a) - \epsilon_v) \qquad (11)$$

Equation 10 finds a point in time where the computed compression of a variation is less than the average computed compression of all variations in that time unit. Equation 11 computes the time unit m, where $1 \le m \le n$, such that before that point the sum of the computed compression of a_i is always less than the relaxed minimum support threshold. In this case the compression records of a_i from 1 to m are considered as unpromising and are pruned.

V. EXPERIMENTS

The performance of the system was evaluated on the data collected from two smart apartments. The layout of the apartments including sensor placement and location tags are shown in Figure 5. We will refer to apartments in Figures 5a and 5b as apartments 1 and apartment 2. The apartments were equipped with infrared motion sensors installed on ceilings, infrared area sensors installed on walls, and switch contact sensors to detect open/close status of the doors and cabinets. The data was collected during 17 weeks for apartment 1, and during 19 weeks for apartment 2. During data collection, the resident in apartment 2 was away for approximately 20 days, once during week 12 and once during week 17. Also the last week of data collection in both apartments does not include a full cycle. In our experiments, we constrain each batch to contain approximately one week of data. In our experiments, we set the maximum errors ϵ_s, ϵ_g and ϵ_v to 0.1, as suggested in literature. Also σ_g was set to 0.75 based on several runs of experiments.

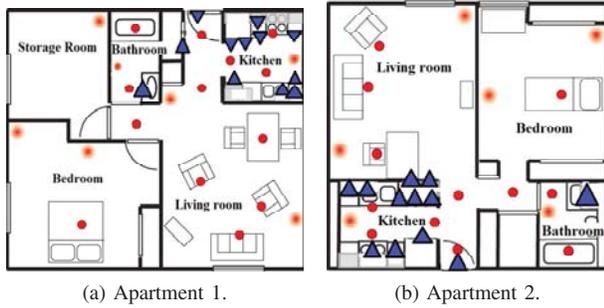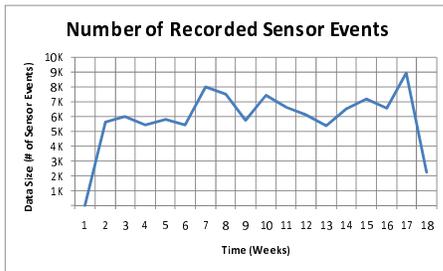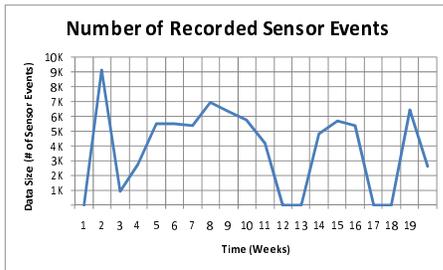(a) Apartment 1.　　　　　(b) Apartment 2.

Fig. 5: Sensor map and location tags for each apartment. On the map, circles show motion sensors while triangles show switch contact sensors.
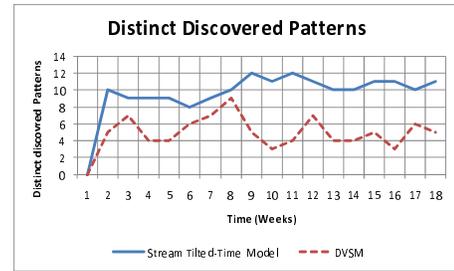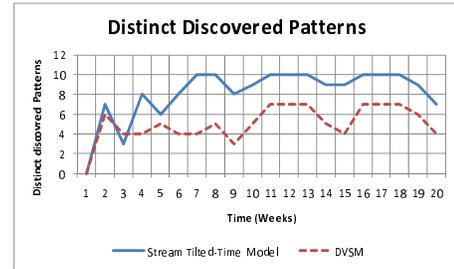

(a) Apartment 1.

(b) Apartment 2.

Fig. 6: Total number of recorded sensor event over time (time unit = weeks).

(a) Apartment 1 (time unit = weeks).

(b) Apartment 2 (time unit = weeks).

Fig. 7: Total number of distinct discovered patterns over time.

To be able to evaluate the results of our algorithms based on a ground truth, each one of the datasets was annotated with activities of interest. A total of 10 activities were noted for each apartment. Those activities included bathing, bed-toilet transition, eating, leave home/enter home, housekeeping, meal preparation(cooking), personal hygiene, sleeping in bed, sleeping not in bed (relaxing) and taking medicine. Apartment 1 includes $193,592$ sensor events and $3,384$ annotated activity instances. Apartment 2 includes $132,550$ sensor events and $2,602$ annotated activity instances. Figures 6a and 6b show the number of recorded sensor events over time. As we mentioned, resident of apartment 2 was not at home during two different time periods, hence we can see the gaps in Figure 6b.

We ran our algorithms on both apartments' datasets. Figures 7a and 7b show the number of distinct discovered patterns over time based on using a global support employed by DVSM versus using multiple regional support as in our proposed method. The results confirm our hypothesis that our proposed method is able to detect a higher percentage of interesting patterns using multiple regional support values. Some of the patterns that have not been discovered are indeed quite difficult to spot and also in some cases less frequent. For example the housekeeping activity happens every 2-4 weeks and is not associated with any specific sensor. Also some of the similar patterns are merged together, as they use the same set of sensors, such as eating and relaxing activities. It should be noted that some of the activities are discovered multiple times in form of different patterns, as the activity might be performed in a different motion trajectory using different sensors. One also can see that the number of discovered patterns increases at the beginning and then is adjusted over time depending on the perceived patterns in the data. The number of discovered patterns depends on perceived patterns in current data batch and previous batches, as well as the compression of patterns in tilted-time window records. Therefore, some of the patterns might disappear and reappear over time which can be a measure of how consistently the resident performs those activities.

As already mentioned, to reduce the number of discovered patterns over time, our algorithm performs two types of pruning. The first type of pruning called normal pruning, prunes patterns and variations while processing the current data batch. The second type of pruning is based on tail pruning to discard unpromising patterns and variations stored in titled-time window. Figures 8a and 8b show the results of both types of pruning on the first dataset. Figures 8d and 8e show the results of both types of pruning on the second dataset. Figures 8c and 8f show the tail pruning results in tilted-time window over time. Note that the gaps for apartment 2 results are due

to the 20 days when resident was away.

By comparing the results of normal pruning in Figures 8a and 8d against the number of recorded sensors in Figures 6a and 6b, one can see that the normal pruning somehow follows the pattern of recorded sensors. If more sensor events are available, more patterns would be obtained, and also more patterns would be pruned. For the tail pruning results, depicted in Figures 8b, 8e, 8c and 8f the number of tail pruned patterns at first increases in order to discard the many unpromising patterns at the beginning. Then the number of tail pruned patterns decreases over time as the algorithm is stabilized.

To better see an example of discovered patterns and variation, several patterns are shown in 9. We have developed a visualization software that allows us to visualize the patterns and variations along with their statistics. Figure 9a shows a "taking medication" activity in apartment 1 that was pruned at third week due to its low compression. Figure 9b shows two variations of "leave home" activity pattern in apartment 2. Note that we use a color coding to differentiate between different variations, if multiple variations are chosen to be shown simultaneously. Figure 9c shows "meal preparation" pattern in apartment 2.

To see how consistent the variations of a certain general pattern are, we used a measure called "variation consistency" based on using the externally provided labels. We define the varication consistency as in Equation 12. Here $|v_{11}|$ refers to the number of variations that have the same label as their general pattern, and $|v_{01}|$ and $|v_{10}|$ refer to the number of variations that have a different label other than their general pattern's label.

$$purity(P_i) = \frac{|v_{11}|}{|v_{11}| + |v_{01}| + |v_{10}|} \qquad (12)$$

Figures 10a and 10b show the average variation consistency for apartment 1 and 2. As mentioned already, for each current batch of data the irrelevant variations are discarded using a mutual information method. The result confirm that that the variation consistency increases at the beginning, and then it quickly stabilizes due to discarding irrelevant variations for each batch.

To show the changes of a specific pattern over time, we show the results of our algorithm for "taking medication" activity over time. Figure 11a shows the number of discovered variations over time for "taking medication" activity. Figure 11b shows the same results in the tilted-time window over time. We can clearly see that the number of discovered variations quickly drops due to the tail pruning process. This shows that despite the fact that we are maintaining the time records over time for all variations, yet many of the uninteresting, unpromising and irrelevant variations will be pruned over time, making our algorithm more efficient in practice.

We also show how the average duration of the "taking medication" pattern changes over time in Figure 11c. Presenting such information can be informative to caregivers to detect any anomalous events in the patterns. Figure 11d shows the consistency of the "taking medication" variations

over time. Similar to the results obtained for the average variation consistency of all patterns, we see that the variation consistency is increased and then stabilized quickly.

In summary, the results of our experiments confirm that we can find sequential patterns from a steam of sensor data over time. It also shows that using two types of pruning techniques allows for a large number of unpromising, uninteresting and irrelevant patterns and variation to be discarded, in order to achieve a more efficient solution that can be used in practice.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we showed a method for discovering sequential patterns over time from a stream of sensor data. We provided an extension of the tilted-time window model for continuity-based, varied order sequential patterns according to the special requirements of our application domain. Not only our proposed method can be used in an activity discovery and recognition system, but it also can be applied to other application domains, such as Web click mining. In the future, we intend to apply our method to other application domains, as well as to use our model in a fully functional system deployed in a real home. We also intend to design anomaly detection methods to detect anomalies in the observed data over time.

REFERENCES

[1] M. C. Mozer, R. H. Dodier, M. Anderson, L. Vidmar, R. F. C. Iii, and D. Miller, "The neural network house: An overview," in *Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments*, 1998, pp. 110–114.

[2] O. Brdiczka, J. Maisonnasse, and P. Reignier, "Automatic detection of interaction groups," in *Proceedings of the 7th international conference on Multimodal interfaces*, 2005, pp. 32–36.

[3] M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, and D. Hahnel, "Inferring activities from interactions with objects," *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 50–57, Oct.-Dec. 2004.

[4] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *BSN '06: Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, 2006, pp. 113–116.

[5] L. Liao, D. Fox, and H. Kautz, "Location-based activity recognition using relational markov networks," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2005, pp. 773–778.

[6] T. Inomata, F. Naya, N. Kuwahara, F. Hattori, and K. Kogure, "Activity recognition from interactions with objects using dynamic bayesian network," in *Casemans '09: Proceedings of the 3rd ACM International Workshop on Context-Awareness for Self-Managing Systems*, 2009, pp. 39–42.

[7] T. Gu, Z. Wu, X. Tao, H. Pung, , and J. Lu, "epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition." in *Proceedings of the IEEE International Conference on Pervasive Computing and Communication*, 2009.

[8] J. Pei, J. Han, and W. Wang, "Constraint-based sequential pattern mining: the pattern-growth methods," *Journal of Intelligent Information Systems*, vol. 28, no. 2, pp. 133–160, 2007.

[9] P. Rashidi and D. J. Cook, "the resident in the loop: Adapting the smart home to the user," *IEEE Transactions on Systems, Man, and Cybernetics journal, Part A*, vol. 39, no. 5, pp. 949–959, September 2009.

[10] B. Schiele, "Unsupervised discovery of structure in activity data using multiple eigenspaces," in *2nd International Workshop on Location and Context Awareness*. Springer, 2006.

[11] P. Rashidi, D. J. Cook, L. Holder, and M. Schmitter-Edgecombe, "Discovering activities to recognize and track in a smart environment," *IEEE Transaction on Knowledge and Data Engineering*, 2010.

[12] J. Ren and C. Huo, "Mining closed frequent itemsets in sliding window over data streams," in *Innovative Computing Information and Control, 2008. ICICIC '08. 3rd International Conference on*, 18-20 2008, pp. 76 –76.
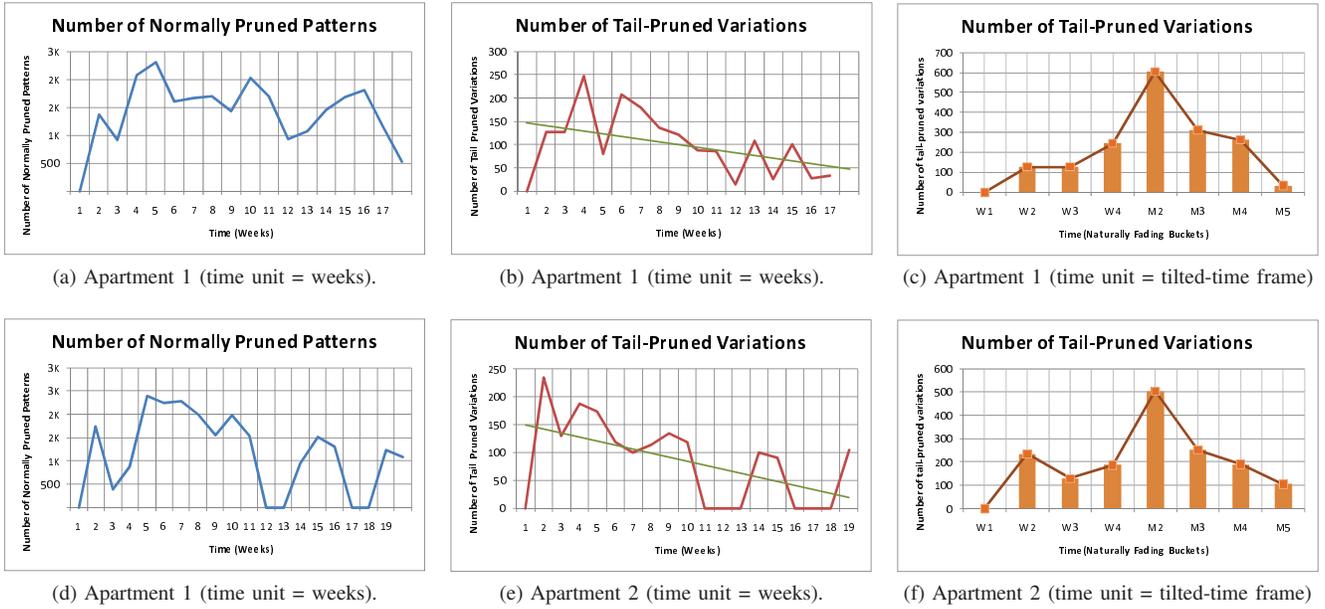
(a) Apartment 1 (time unit = weeks).

(b) Apartment 1 (time unit = weeks).

(c) Apartment 1 (time unit = tilted-time frame)

(d) Apartment 1 (time unit = weeks).

(e) Apartment 2 (time unit = weeks).

(f) Apartment 2 (time unit = tilted-time frame)

Fig. 8: Total number of tail-pruned variations over time. For the bar charts, W1-W4 refers to the week 1-4, and M1-M5 refers to month 1-5.

(a) A visualization of the "taking medication" variation in apartment 1.

(b) Two variations of "leave home" pattern in apartment 1.

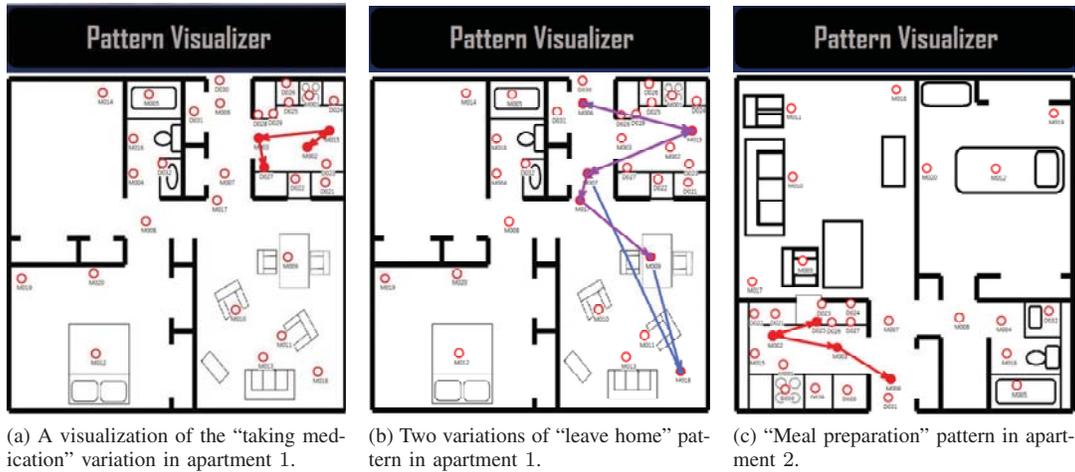(c) "Meal preparation" pattern in apartment 2.

Fig. 9: Visualization of patterns and variations.

[13] C. Giannella, J. Han, J. Pei, X. Yan, and P. S. Yu, *Mining Frequent Patterns in Data Streams at Multiple Time Granularities*. MIT Press, 2003, ch. 3.

[14] H. fu Li, S. yin Lee, and M. kwan Shan, "An efficient algorithm for mining frequent itemsets over the entire history of data streams," in *In Proc. of First International Workshop on Knowledge Discovery in Data Streams*, 2004.

[15] G. S. Manku and R. Motwani, "Approximate frequency counts over data streams," in *VLDB '02: Proceedings of the 28th international conference on Very Large Data Bases*. VLDB Endowment, 2002, pp. 346–357.

[16] G. Chen, X. Wu, and X. Zhu, "Sequential pattern mining in multiple streams," in *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 585–588.

[17] A. Marascu and F. Masseglia, "Mining sequential patterns from data streams: a centroid approach," *Journal of Intelligent Information Systems*, vol. 27, no. 3, pp. 291–307, 2006.

[18] C. Raïssi, P. Poncelet, and M. Teisseire, "Need for speed: Mining sequential pattens in data streams," in *BDA05: Actes des 21iemes Journees Bases de Donnees Avancees*, October 2005.

[19] T. L. Hayes, M. Pavel, N. Larimer, I. A. Tsay, J. Nutt, and A. G. Adami, "Distributed healthcare: Simultaneous assessment of multiple individuals," *IEEE Pervasive Computing*, vol. 6, no. 1, pp. 36–43, 2007.

[20] R. Agrawal and R. Srikant, "Mining sequential patterns," in *ICDE '95: Proceedings of the Eleventh International Conference on Data Engineering*. Washington, DC, USA: IEEE Computer Society, 1995, pp. 3–14.

[21] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "Prefixspan: Mining sequential patterns by prefix-projected growth," in *Proceedings of the 17th International Conference on Data Engineering*. Washington, DC, USA: IEEE Computer Society, 2001, pp. 215–224.

[22] J. Wang and J. Han, "Bide: Efficient mining of frequent closed sequences," in *ICDE '04: Proceedings of the 20th International Conference on Data Engineering*. Washington, DC, USA: IEEE Computer Society, 2004, p. 79.
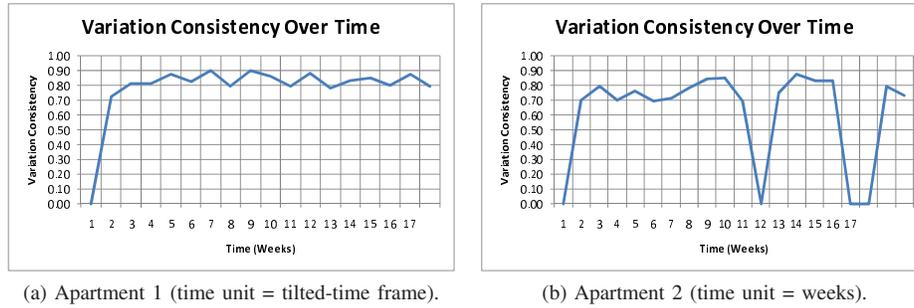
(a) Apartment 1 (time unit = tilted-time frame).

(b) Apartment 2 (time unit = weeks).

Fig. 10: Total number of distinct discovered patterns and their variation consistency over time.

(a) (Number of discovered variations (time unit = weeks).

(b) Number of discovered variations (time unit = tilted-time frame).

(c) Duration (time unit = weeks).
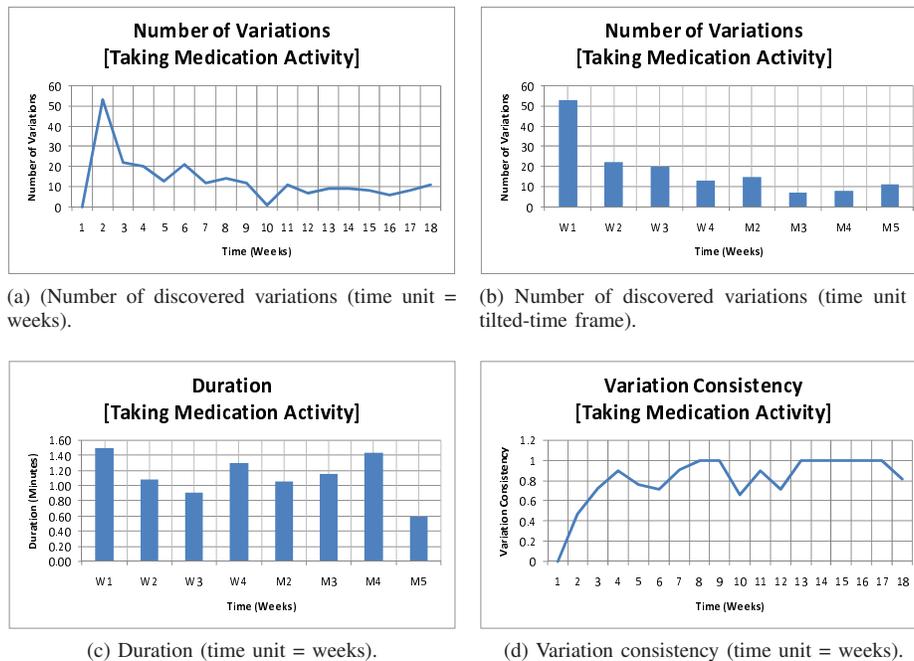

(d) Variation consistency (time unit = weeks).

Fig. 11: Number of discovered variations, duration and consistency for "taking medication" activity pattern over time.

[23] F. Masseglia, F. Cathala, and P. Poncelet, "The psp approach for mining sequential patterns," in *PKDD '98: Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*. London, UK: Springer-Verlag, 1998, pp. 176–184.

[24] M. Garofalakis, J. Gehrke, and R. Rastogi, "Querying and mining data streams: you only get one look a tutorial," in *SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2002, pp. 635–635.

[25] J. H. Chang and W. S. Lee, "Finding recent frequent itemsets adaptively over online data streams," in *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2003, pp. 487–492.

[26] W.-G. Teng, M.-S. Chen, and P. S. Yu, "A regression-based temporal pattern mining scheme for data streams," in *VLDB '2003: Proceedings of the 29th international conference on Very large data bases*. VLDB Endowment, 2003, pp. 93–104.

[27] J. Han, Y. Chen, G. Dong, J. Pei, B. W. Wah, J. Wang, and Y. D. Cai, "Stream cube: An architecture for multi-dimensional analysis of data streams," *Distrib. Parallel Databases*, vol. 18, no. 2, pp. 173–197, 2005.

[28] C. Raïssi and M. Plantevit, "Mining multidimensional sequential patterns over data streams," in *DaWaK '08: Proceedings of the 10th international conference on Data Warehousing and Knowledge Discovery*. Berlin,

Heidelberg: Springer-Verlag, 2008, pp. 263–272.

[29] S. Papadimitriou, A. Brockwell, and C. Faloutsos, "Adaptive, hands-off stream mining," in *VLDB '2003: Proceedings of the 29th international conference on Very large data bases*. VLDB Endowment, 2003, pp. 560–571.

[30] K. K. Loo, I. Tong, B. Kao, and D. Cheung, "Online algorithms for mining inter-stream associations from large sensor networks," in *PAKDD*, 2005.

[31] J. Cheng, Y. Ke, and W. Ng, "A survey on algorithms for mining frequent itemsets over data streams," *Knowledge and Information Systems*, vol. 16, no. 1, pp. 1–27, 2008.

[32] B. Liu, W. Hsu, and Y. Ma, "Mining association rules with multiple minimum supports," in *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 1999, pp. 337–341.

[33] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

[34] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.