Sequential Behavior Pattern Discovery with Frequent Episode Mining and Wireless Sensor Network

Li Li[1], Xin Li[2]*, Zhihan Lu[3]*, Jaime Lloret[4], Houbing Song[5]*
1. School of Architecture, Southeast University, China
2. School of Urban Design, Wuhan University, China
3. Department of Computer Science, University College London, UK
4. Department of Communications, Universitat Politecnica de Valencia, Spain
5. Department of Electrical and Computer Engineering, West Virginia University, Montgomery, WV 25136 USA
Email: lili_arch@163.com, li-xin@whu.edu.cn, Z.Lu@cs.ucl.ac.uk, jlloret@dcom.upv.es, h.song@ieee.org

***Abstract*—By recognizing patterns in occupants' daily activities, building systems are able to optimize and personalize services. Established technologies are available for data collection and pattern mining, but they all share the drawback that the methodology used for data collection tends to be ill suited for pattern recognition. For this research, we developed a bespoke wireless sensor network (WSN) and combined it with a compact data format for frequent episode mining (FEM) to overcome this obstacle. The proposed framework has been evaluated with both synthetic data from a smart home simulator and with real data from a self-organizing WSN in a student's home. We are able to demonstrate that the framework is capable of discovering sequential patterns in heterogeneous sensor data. With corresponding scenarios, patterns in daily activities can be deduced. The framework is self-contained, scalable and energy efficient and is thus applicable in multiple building system settings.**

***Index Terms*—Smart City, Smart Building, Frequent Episode Mining, Wireless Sensor Network**

I. INTRODUCTION

From a technical point of view, a smart city tries to improve the quality of life of its citizens in terms of urban services by utilizing information and communication technology (ICT), data mining (DM), and other new technologies to improve urban services. Patterns of daily behavior are a decisive factor in many aspects of the urban environment, such as traffic, air quality, energy cost, and so on. Considering that urban dwellers spend approximately 90 percent of their lives indoors, it's no surprise that buildings are responsible for about two-thirds of all electrical energy consumption. Making cities smarter begins indoors. Utilizing data collection to reveal occupants' behavior patterns enables data-driven decisions for smarter buildings. Usage can be anticipated, thus reducing the consumption while improving the experience[1].

The data-driven decision for buildings is largely based on data acquisition and data mining technologies. Wireless sensor networks are widely used for data acquisition since wireless is low cost and more flexible than wired solutions [2]. Extensive research has been performed on efficiency [3][4] and mobility [5-7]. Since it is impossible for the system designer to envision all possible contexts beforehand, decision making control systems largely rely on data mining and machine learning techniques such as an artificial neural network (ANN), a support vector machine (SVM), a self-organizing map (SOM), a hidden Markov model (HMM), and frequent pattern mining (FPM)[8]. Artificial intelligence provides many benefits for data gathering systems [9]

A. Existing problem

In spite of all of the studies conducted on data acquisition and data mining for building systems, no practical solution has been provided. There are several reasons for this:

- Too complex. Most of the research was conducted in an experimental environment that required expert installation, maintenance, and upgrade of the system. Some mining algorithms require extensive parameter settings that are not intuitive and need professional and prior knowledge of the environment. Supervisory algorithms need additional training data that is hard to obtain in a real world application.
- Too simple. There are two main types of data that can be recorded in a building system: numerical (discrete and continuous sensor values) and categorical (such as weather conditions: windy, snowy, sunny, etc.). Unfortunately, most algorithms can only address one type at a time. Although datatypes are interchangeable, additional parameters and prior knowledge of the dataset are required.
- Gap in research fields. Established technologies are available for data collection and data mining, but they all

share the drawback that methodology used for data collection tends to be ill suited for purposes of data mining. WSN developers continue to improve efficiency, regardless of the type of data and measuring frequency actually needed. DM researchers focus on accuracy of the algorithm, without regard for the data source or collection efficiency.

B. *Solution*

This research integrates data acquisition and data mining techniques efficiently and practically to discover behavior patterns. We first propose a compact data format that encompasses both sensor data about spontaneous events and periodic environmental readings. This format requires less data and transmission from WSN. The environmental information can be used to reduce redundancy and deduce behavior patterns. For data acquisition, a mesh WSN called a self-organizing WSN has been designed that requires neither planning nor configuration. A frequent episode mining (FEM) algorithm is used for mining sequential patterns. It is adapted to mine both categorical and numerical data in the dataset by introducing a DBSCAN clustering algorithm. Finally, a framework is proposed to seamlessly combine all of these technologies. Evaluation uses both synthetic data from a smart home simulator and real data from a self-organizing WSN in a student home. We are able to demonstrate that the framework is capable of discovering sequential patterns in heterogeneous sensor data. By applying corresponding scenarios, patterns of daily activities can be deduced. The framework is self-contained, scalable and energy efficient and is thus applicable to different building system settings.

II. THE COMPACT DATA FORMAT

A standard building sensor data format has not been established. There are two main types of data recording: event based and interval based. The event based approach only records when the sensor is triggered. Collected data size is small, but slowly changing environment parameters may be missed. The interval based approach does sampling using a fixed rate. Frequency may be increased to avoid missing short events, but data size will grow correspondingly. In our approach, we mix these two approaches. Building sensor data are divided into two categories: event data and ambient data. Event data means the sensor records activity triggered by the occupant, such as open/close the door, turn on/off the light. Ambient data refer to environmental parameters, such as temperature and light intensity. It provides the context and scenario for events. Time is also regarded as ambient data. Using this approach, we keep the sampling rate as low as possible without missing an event.

III. DATA ACQUISITION

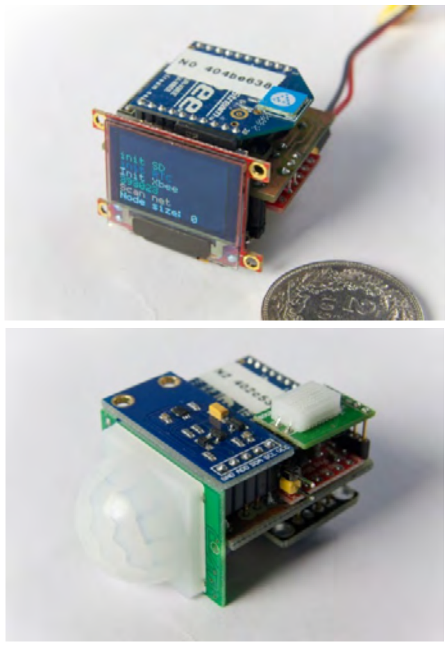
Self-organizing WSN is a mesh network designed to minimize settings, configurations and dependence on infrastructure during installation so that it can be easily deployed or removed. The self-organizing ability enables the network to form the mesh network automatically at installation or when a new node is added.

A. *Sensor Nodes*

The network consists of three types of nodes: gateway nodes, active sensor nodes, and passive sensor nodes. Each sensor node is equipped with an ATmega328P micro controller for processing and an XBee DigiMesh 2.4 Wireless RF Module for communication. Additional components may be added depending on the node's task.

The gateway node is primarily used to record data collected from the network. It is also responsible for management of network operations, including node discovery, adding, deleting, and error control. It is equipped with a DS1307 real-time clock module that provides a time stamp for each sample. It uses an SD card for data and log file storage, and an LCD for displaying real time information (Figure 1, a).

The passive node is used to collect ambient data and will send data after receiving an upload request from the gateway node. The active node is designed to detect events. It will send data whenever specific events occur. This design can reduce data transmission and energy consumption by running on a relative low sampling rate and can capture instantaneous events that occur in sampling intervals. In this experiment, the passive node is equipped with temperature, humidity, luminous intensity, and passive infrared (PIR) sensors (Figure 1, b); the active node is equipped with a reed switch to detect the opening and closing of doors and windows.



a.

b.

Figure 1. Sensor nodes: a, gateway node, b, passive sensor node

B. *Database*

A data managing program is provided, which is a JAVA program with MySQL database. It parses raw data stored on the gateway node and separates it into a table according to the sensor node that is designated by the low 32-bit address of the Xbee address. In the table, the first column is the timestamp of the sample, and the rest of the columns save sensor values. The database managing program can read historical data from the database on demand and present the data in time series charts. This type of chart helps provide an understanding of correlations between different sensor data.

IV. PATTERN MINING

The proposed algorithm provides a data mining technique that requires fewer parameters and non priori knowledge and provides more meaningful sequential patterns. FEM was selected as the core algorithm. The advantages of using FEM are: (1) It needs only one simple parameter, i.e., minimum support. (2) It allows gaps in the pattern, which gives it tolerance to randomness and noise in real life data. (3) The mining process is unsupervised, no training data are needed, and no pre-segmentation is needed. (4) There is no randomness in output patterns.

It also has some disadvantages in mining building sensor data: (1) It only works with categorical data. (2) There could be many redundancy and meaningless patterns in the mining result. In this research, a preprocessing module is introduced to convert numerical ambient sensor data into categorical data without prior knowledge. By introducing the associated ambient sensor data and complex temporal database, redundancy can be significantly reduced. With environment information in the mining result, patterns of daily activities can be deduced.

By introducing a more complex data structure and visualization module, it is also possible to display occurrences of patterns in the mining process.

A. *Preliminary*

In 1993, Agrawal invented the Apriori algorithm to find all co-occurrence relationships, called associations, among data items [10]. He first applied downward-closure property to find frequent itemsets in the transaction database. This property narrows the search space drastically and enables the mining of a large scale database.

FEM is an Apriori based algorithm mining temporal database. For example: ⟨(2 4)(1 3 4)(6)(5)(3 7)(6)(2 3)(1 3 6)(5 8)(7)(6)(2)(1 2 3)(4 5)⟩ is a temporal database. Each number inside is called an item. Numbers inside one pair of brackets form an itemset. The position of the itemset in the database indicates the sequence of their appearances. Items in the same itemset appear at the same time. Given a minimum support 3, <(2)(1 3)(5)> is one of the frequent episodes because it appears ≥ 3 times in the database. Similarly, <(6)>:4 and <(2)(1 3)>:3 are frequent episodes (the value after the colon represents the frequency of its appearance in the database). <(2)(1 3)(5)> and <(6)> are called closed frequent episodes because no super sequence has the same frequency, <(2)(1 3)> is not closed because <(2)(1 3)(5)> have the same frequency. A more formal definition can be found in [11]. Our algorithm is able to extract closed frequent episodes on the complex temporal database consisting of heterogeneous sensor data types.

B. *Preprocessing module*

Both numerical and categorical sensor data have to be converted into categorical values (i.e., items) for the temporal database. In this case, natural numbers are used to represent the item ID. It is not difficult to assign categorical values with the item ID. For example, the on/off of one light can be designated with "1" and "2", respectively. For numerical values, such as room temperature, ranging from 5 to 30 degrees, it is impossible to assign a symbol for each value recoded. However, only the ambient values at the time some event happens are of interest. Such ambient data are called associated ambient sensor data, for example, the

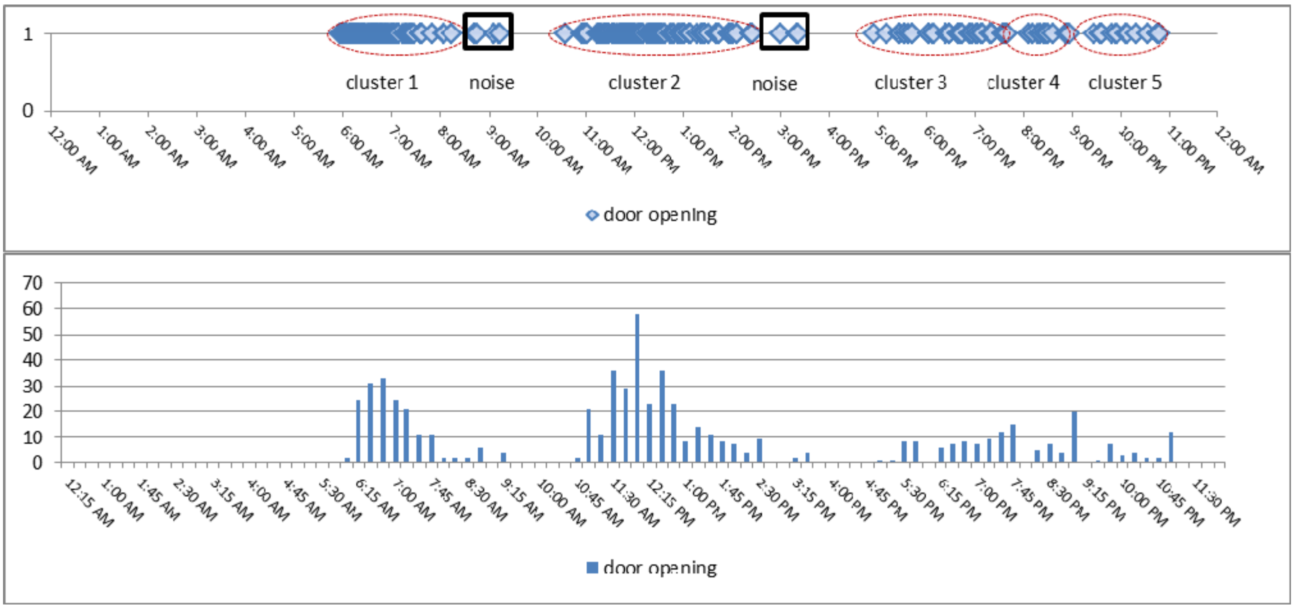


Figure 2 The 1)occurrences and 2)distribution of door opening time in one day based on 15 days observation

temperature when a certain window is opened.

People usually perform their daily activities at similar times and in similar circumstances. This assumption can be demonstrated by data collected for a real life experiment. Figure 2 shows the occurrences and distribution of door openings in one day based on 15 days of observation. It is clear that there are several high density time periods.

DBSCAN [12] is chosen for clustering associated ambient sensor data because: (1) It is a density-based clustering algorithm that fits our need. (2) It needs one parameter, max distance, and this value can be calculated based on historical data. (3) It recognizes isolated data points as noise rather than tries to cluster them.

The creation of the temporal database consists of 3 main steps, Figure 3 shows the process:

First, each event sensor data are assigned an item ID. For example, the window open event is assigned ID 1. This will be recorded in table b, item ID mapping table.

Second, select all associated ambient data for each event and try to cluster with the DBSCAN algorithm. In table a, there are two types of ambient data: time and temperature. For example, in sub-table 2, the window-opening event occurs at 6:39 A.M., 7:24 A.M., 8:05 A.M., and 6:33 A.M. each day. In sub-table 3, we can obtain the temperatures at these time points, which are 27.5℃, 26.8℃, 28.4℃, and 22.5℃. By clustering these data with DBSCAN and 1℃ as max distance, the first three temperatures, 27.5℃, 26.8℃, 28.4℃, can be clustered into one cluster, while the fourth one, 22.5℃, will be defined as noise. No cluster can be found for a window-opening and closing event of 15 minutes. The new found temperature cluster is also assigned item ID 5.

After all associated ambient data clusters are found and assigned item IDs, all events will be sorted in an array by timestamp. Their associated ambient data will be added to the same itemset (in table c). In this case, the temporal database will be :⟨(1 5)(3)(4 8)(2 6)(1 5)(3)(4 8)(2 6)(1 5)(3)(4 8)(2 7)(1)(3)(4 8)(2 7)⟩.

C. Frequent episode mining module

Although FEM derives from FPM and shares many similarities, there are still some differences in the algorithm.

a, sensor data tables

1 Window sensor	
Time stamp	state
08/18 6:39 a.m.	open
08/18 5:21 p.m.	close
08/19 7:24 a.m.	open
08/19 8:14 p.m.	close
08/20 8:05 a.m.	open
08/20 10:09 p.m.	close
08/21 6:33 a.m.	open
08/21 9:36 p.m.	close

2 Door sensor	
Time stamp	state
08/18 7:30 a.m.	open
08/18 4:22 p.m.	close
08/19 8:15 a.m.	open
08/19 4:16 p.m.	close
08/20 8:45 a.m.	open
08/20 4:19 p.m.	close
08/21 7:50 a.m.	open
08/21 4:26 p.m.	close

3 Temperature sensor data	
Time stamp	value
08/18 6:39 a.m.	27.5
08/18 7:30 a.m.	28.8
08/18 4:22 p.m.	19.9
08/18 5:21 p.m.	18.3
08/19 7:24 a.m.	26.8
08/19 8:15 a.m.	23.7
08/19 4:16 p.m.	21.9
08/19 8:14 p.m.	17.5
08/20 8:05 a.m.	28.4
08/20 8:45 a.m.	25.5
08/20 4:19 p.m.	12.4
08/20 10:09 p.m.	22.9
08/21 6:33 a.m.	22.5
08/21 7:50 a.m.	20.4
08/21 4:26 p.m.	15.7
08/21 9:36 p.m.	22.2

b, item ID mapping table

Events and clusters	Item ID
Window open	1
Window close	2
Door open	3
Door close	4
Cluster (27.5°C, 26.8°C, 28.4°C)	5
Cluster (18.3°C,17.5°C)	6
Cluster (22.9°C, 22,2°C)	7
Cluster (4:22 p.m., 4:16 p.m., 4:19 p.m., 4:26 p.m.)	8

c, create temporal database with item IDs

Date		08/18				08/19				08/20				08/21			
Time		6:39 a.m.	8:30 a.m.	4:22 p.m.	5:21 p.m.	7:24 a.m.	8:15 a.m.	4:16 p.m.	8:14 p.m.	8:05 a.m.	8:25 a.m.	4:19 p.m.	10:09 p.m.	6:33 a.m.	8:33 a.m.	4:26 p.m.	9:36 p.m.
Event		Win open	Door open	Door close	Win close	Win open	Door open	Door close	Win close	Win open	Door open	Door close	Win close	Win open	Door open	Door close	Win close
Temporal database	Item ID (event)	1	3	4	2	1	3	4	2	1	3	4	2	1	3	4	2
	Item ID (ambient)	5		8	6	5		8	6	5		8	7			8	7

Figure 3. Convert sensor data tables into temporal database

One essential difference in mining the temporal database and sequence database is the frequency metric, i.e., the frequency of occurrence of a pattern. No measures have been commonly accepted in temporal database mining. In this paper, a metric called LMaxnR-freq [13], which is short for the leftmost maximal non-redundant set of occurrences, is adopted.

In the mining process, the length of each frequent episode grows by iteration. Each length L frequent episode searches for the L+1 episode in the projected database with the metric defined in LMaxnR-freq. All found episodes are maintained in a tree structure called the enumeration tree. There, episodes will be pruned or kept for the next iteration. The actual implementation of the algorithm is described in these papers [14, 15].

In Figure 3 for example, with a min_sup = 3, a frequent episode, ⟨(1 5)(3)(4 8)(2)⟩, can be found. This episode can be translated into a behavior pattern: in the morning the window will be opened when the temperature is approximately 26.5 ℃ and then the door will be opened. Around 4:20 P.M. the door will be closed and afterwards the window.

D. *Visualization module*

The visualization module visualizes data created during the whole mining process. Unlike most of the algorithms that just provide frequent episodes and their supports, the visualization module enables researchers to track occurrences of patterns and understand actual meaning of the patterns. There are three different charts: the sensor data table chart, the enumeration tree chart, and the frequent episode chart.

The sensor data table chart displays content of the table, metadata from the sensor metadata table, and some statistics of the table, such as data type of the sensor value, number of different values and size of the table, and so on.

The enumeration tree chart and frequent episode chart work together to display information about the frequent episode. In the enumeration tree chart, each circle represents a node in the tree structure. Different colors represent different states of the node: black is pruned and yellow is not pruned. The number in the circle on the left side of the colon is the item ID of the node, right side is frequency of the episode. The link between the nodes represents the type of extension: the solid line is horizontal extension and the dashed line is vertical extension. The frequent episode chart is a 2D-grid, each row contains occurrences of a certain item ID, and each column represents a timestamp in the temporal database. When a certain node is selected in the enumeration tree chart, the corresponding occurrences of this episode will be marked in the frequent episode chart. Each item in one occurrence will be connected by a red line (Figure 4).

V. TEST OF THE FRAMEWORK

A. *The framework of the pattern discovery process*

The framework consists of three parts: data acquisition, data managing, and data mining (Figure 5). The data acquisition task is performed by the WSN that contains both active sensor and passive sensor nodes. The ambient sensor periodically records environment parameters. The event sensor records events triggered by occupants. Data are stored and maintained in the form of tables in the database. There are event sensor data and ambient sensor data for different recordings. The metadata table provides spatial relationships for linking event sensor tables with ambient sensor tables. With the data and relationships in the database, the temporal database can be created for mining. Then, WSN data can be converted into a normal FEM problem. After the mining process, discovered frequent episodes can be translated into behavior patterns with information in the database.

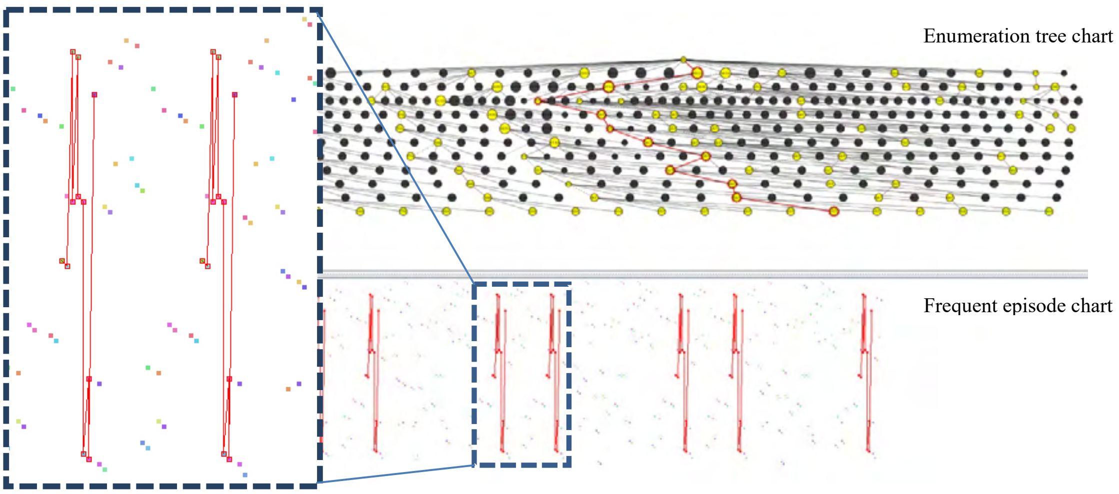


Figure 4. Enumeration tree chart and frequent episode chart

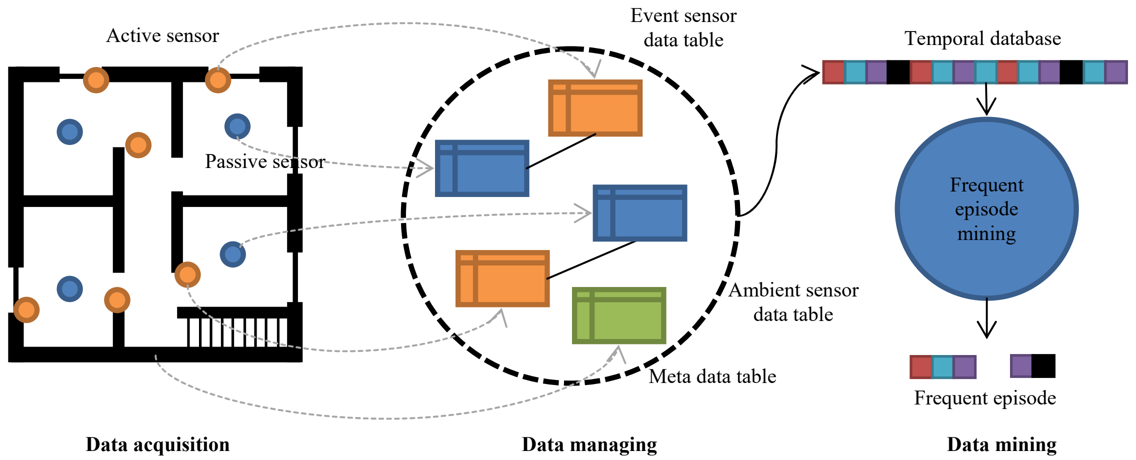


Figure 5. Diagram for the pattern discovery process

There are several items that can affect output patterns. The first is features of input sensor data, including density of the sequential pattern, and the average length of patterns. The second is parameters for clustering ambient sensor data, i.e., the max distance between each data point. The third is parameters for the FEM mining algorithm, including minimum support of the pattern (min_sup), size of the window constrain (max_gap), and degree of approximation for pruning the sub-sequence (max_err_bound). These parameters can be determined by evaluating the data source without additional knowledge.

Two tests have been conducted. The first is based on synthetic data generated by a simulator that can generate sensor data records mixed with patterns and noises. The second is based on real life data collected from a student dormitory that provides a demonstration and evaluation of the proposed framework in a real world application.

B. Experiment with synthetic data

In this experiment, one long daily pattern is predefined. The virtual environment generated 10 days sensor data with 10% noise data; 320 data samples were collected, 48 items were generated to represent all events and associated ambient data groups. With minimum support set to 10, the mining process lasted for 93 iterations. In the final iteration, 14026 tree nodes were found, and 14019 of them were pruned. Only 7 nodes were left. They were:

1, <18 15 >:29

2, <36 33 >:19

3, <24 30 27 21 >:19

4, < (18 19) (15 16) >:20

5, <18 (4 15) 1 >:19

6, <46 43 >:19

7, <(24 25) (30 31) (27 28) (18 19 21 22) (4 5 15 16) (1 2 18 19) (15 16 46 47) (36 37 43 44) (33 34) (36 38) (41 42) (33 35 46 48) (18 20 43 45) (4 6 15 17) (9 10) (1 3 13 14) (11 12 24 26) (30 32) (27 29) (21 23) (24 25) (30 31) (27 28) (18 19 21 22) (4 5 15 16) (7 8) (1 2 18 19) (15 16 46 47) (36 37 43 44) (39 40) (33 34) >:9

Figure 6-a shows occurrences of the seventh pattern. It is a length-93 episode that covers the whole daily routine as expected. This experiment shows that the algorithm is able to detect the entire predefined pattern with noise.

C. Experiment with real life data

In this experiment, data collected with the self-organizing WSN installed in a room of a student dormitory for a one month period is used for testing.

The test environment was a 12 ㎡ single room. A self-organizing WSN with eight nodes was installed in the room, including one data logging node, five passive sensor nodes, and two active sensor nodes. Two passive sensor nodes were placed outdoors to monitor the outdoor and corridor environment. The remaining three were placed next to the desk, bed, and basin. The active sensor nodes were installed on the window frame and door frame to monitor the opening and closing states.

In the 30 day period, 732 data sample were recorded. A total of 75 items were generated. The min_sup was set to 20 (there were 20 working days), the constraint window size was set to 3, and the max_error_bound was set to 5. The search ended with 13 iterations. There were 1613 nodes in the enumeration tree (Figure 6-b), 1450 of them were pruned, and 163 were left.

a.

b.

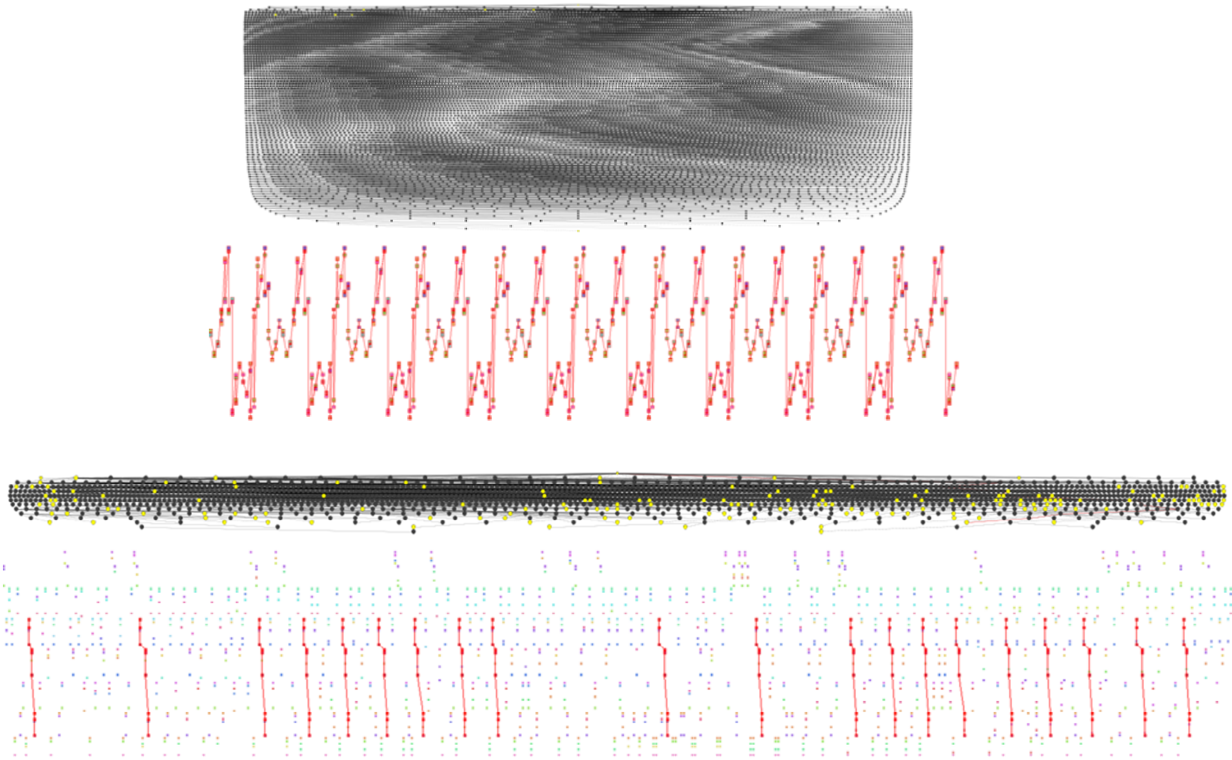


Figure 6. a. Enumeration tree and occurance of pattern 7 with synthetic data,

b.Enumeration tree and pattern <(25 28 34) (36 37 45) (59 61 67) > with real life data

Frequent episodes with associated ambient value clusters can be translated into behavior patterns. For example, the pattern <(25 28 34) (36 37 45) (59 61 67) >:20 (Figure 14) can be interpreted as desk PIR sensor activated around 6:35 with room light on, bed PIR sensor deactivated approximately 6:45 with room light, lamp turned off approximately 7:15, room light off. This was a record of the period when the occupant got up in the morning and turned off the lamp before leaving the room. The desk PIR sensor activated before the bed PIR sensor because there was a delay before the PIR sensor deactivated.

VI. Conclusion and Future Work

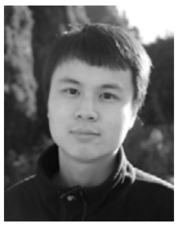
In conclusion, the proposed framework is able to collect sensor data from a building and discover behavior patterns. The advantage of the proposed framework is: (1) It is easy to deploy. (2) It requires less data collection and calculation. (3) It needs very few settings and parameters. (4) It can work with both numerical and categorical data, and the output pattern contains both sensor events and corresponding ambient values. (5) It can first visualize the mining processing and result.

There is some possible future work. First, in the real life data test, there is still redundancy. It is caused by parallel patterns where some items shifted in sequence. By introducing parallel episodes, the output can be condensed. Second, the framework is not limited to building sensors; it can easily be extended to discover daily routines in the city by adding mobile location data, among others.

Acknowledgment

The authors gratefully acknowledge financial support from the National Natural Science Foundation of China (No. 51408442).

Biographies



Li LI received the Ph.D. degree in CAAD from the ETH Zurich, Switzerland, in 2016. In February 2016, he joined the Lab. of Architectural Algorithm & Application, Southeast University, Nanjing, China. He is also the co-founder and manager of an IOT company called Nexiot AG, based in Zurich. His research interests lie in the areas of internet of things, wireless sensor networks, indoor tracking, and data mining.

Xin LI received his Ph.D. degree in Architecture from Wuhan University, China, in 2016. He has joined the faculty of School of Urban Design at Wuhan University in 2011. From 2013 to 2014, he was a visiting scholar at the chair of information architecture of ETH Zurich. His main research interests focus on urban design, digital architecture, GIS aided design, and information architecture.

Zhihan LV is an engineer and researcher of virtual/augmented reality and multimedia major in mathematics and computer science, having plenty of work experience on virtual reality and augmented reality projects, engage in application of computer visualization and computer vision. His research application fields widely range from everyday life to traditional research fields (i.e. geography, biology, medicine). During the past years, he has completed several projects successfully on PC, Website, Smartphone and Smartglasses.

Jaime LLORET [M'07-SM'10] is associate professor at Politechnic University of Valencia, Spain. He was Internet Technical Committee chair during 2014-2015 and he is chair of IEEE 1907.1. He is head of the Research Group "Communications and Networks" at the Research Institute IGIC and head of the Innovation Group EITACURTE. He is director of the University Master "Digital Post Production". He is a Cisco Certified Network Professional Instructor. He is co-editor-in-Chief of Ad Hoc and Sensor Wireless Networks and "Network Protocols and Algorithms".

Houbing SONG received his Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, VA, 2012. In August 2012, he joined the Department of Electrical and Computer Engineering, West Virginia University, Montgomery, WV, where he is currently the Golden Bear Scholar, an Assistant Professor and the Founding Director of the Security and Optimization for Networked Globe Laboratory (SONG Lab, www.SONGLab.us). His research interests lie in the areas of internet of things, edge computing, big data analytics, wireless communications and networking.

REFERENCES

[1] Xu, K., et al., *Toward software defined smart home.* IEEE Communications Magazine, 2016. **54**(5): p. 116-122.

[2] Diallo, O., Rodrigues, J. J. P. C., Sene, M., and Lloret, J., Distributed Database Management Techniques for Wireless Sensor Networks, IEEE Transactions on Parallel and Distributed Systems, Vol. 26, Issue 2, Pp. 604 - 620. February 2015

[3] Mehmood, A. and H. Song, *Smart Energy Efficient Hierarchical Data Gathering Protocols for Wireless Sensor Networks.* Smart Computing Review, 2015. **5**(5): p. 425-462.

[4] Ahmadi, A., et al., *An efficient routing algorithm to preserve k - coverage in wireless sensor networks.* The Journal of Supercomputing, 2014. **68**(2): p. 599-623.

[5] Shojafar, M., Cordeschi, N., Baccarelli, E., *Energy-efficient Adaptive Resource Management for Real-time Vehicular Cloud Services.* IEEE Transactions on Cloud Computing, 2016..

[6] Baek, J., et al., *On a moving direction pattern based MAP selection model for HMIPv6 networks.* Computer Communications, 2011. **34**(2): p. 150-158.

[7] Sun, Y., Q. Jiang, and M. Singhal, *A Pre-Processed Cross Link Detection Protocol for geographic routing in mobile ad hoc and sensor networks under realistic environments with obstacles.* Journal of Parallel and Distributed Computing, 2011. **71**(7): p. 1047-1054.

[8] Rashid, M.M., I. Gondal, and J. Kamruzzaman, *Dependable large scale behavioral patterns mining from sensor data using Hadoop platform.* Information Sciences, 2016.

[9] Zhu, C., Wang, Y., Han, G., Rodrigues, J. J. P. C., and Lloret, J., LPTA: Location Predictive and Time Adaptive Data Gathering Scheme with Mobile Sink for Wireless Sensor Networks, Scientific World Journal, Vol. 2014, Article ID 476253, Pp. 13. 2014

[10] Agrawal, R., et al., *Mining association rules between sets of items in large databases.* SIGMOD Rec., 1993. **22**(2): p. 207-216.

[11] Gan, M. and H. Dai, *Fast Mining of Non-derivable Episode Rules in Complex Sequences*, in *Modeling Decision for Artificial Intelligence*, V. Torra, et al., Editors. 2011, Springer Berlin Heidelberg. p. 67-78.

[12] Ester, M., et al. *A density-based algorithm for discovering clusters in large spatial databases with noise*. in *Kdd*. 1996.

[13] Gan, M. and H. Dai, *Subsequence Frequency Measurement and its Impact on Reliability of Knowledge Discovery in Single Sequences*, in *Reliable Knowledge Discovery*, H. Dai, J.N.K. Liu, and E. Smirnov, Editors. 2012, Springer US. p. 239-255.

[14] Han, J., et al., *Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach.* Data Mining and Knowledge Discovery, 2004. **8**(1): p. 53-87.

[15] Min, G. and D. Honghua. *A Study on the Accuracy of Frequency Measures and Its Impact on Knowledge Discovery in Single Sequences*. in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*. 20

a.

b.

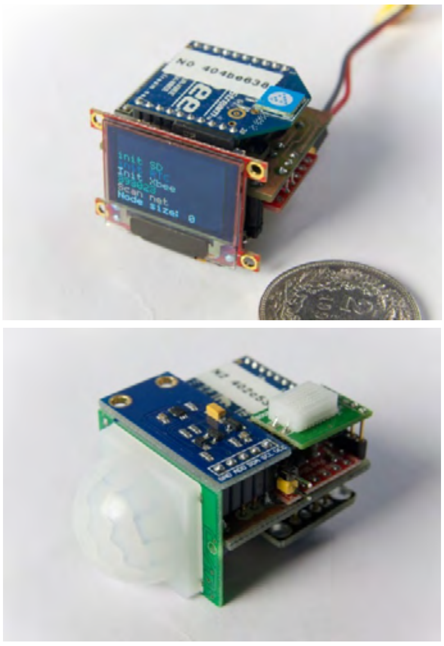


Figure 1 Sensor nodes: a, gateway node, b, passive sensor node

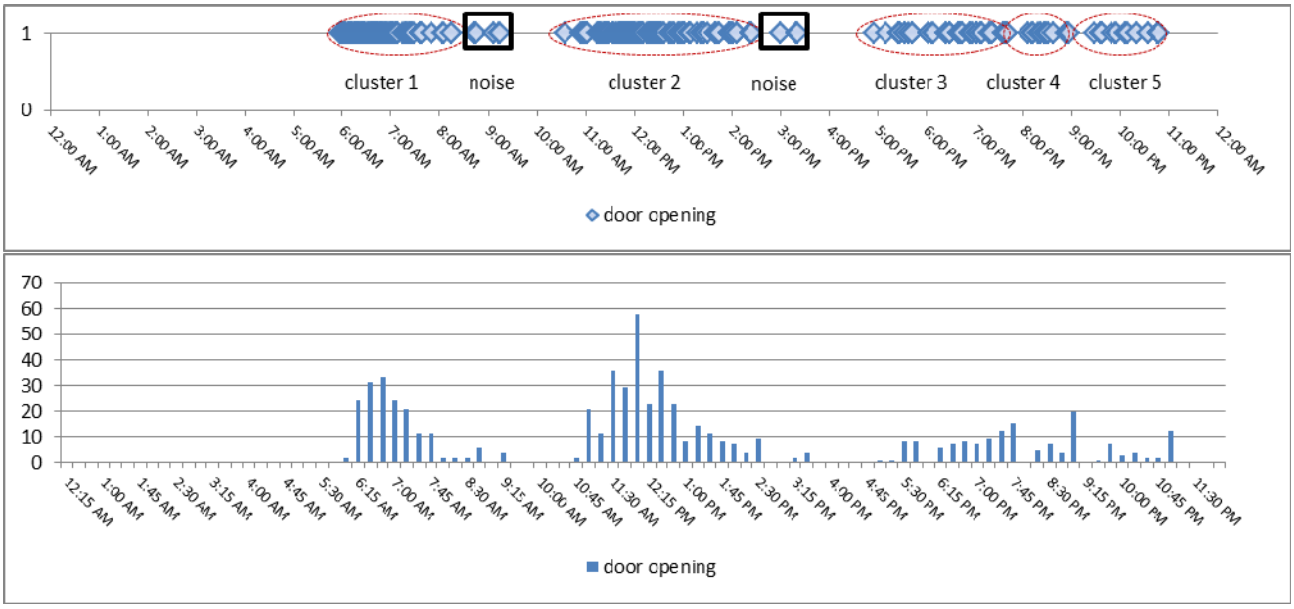


Figure 2 The 1)occurrences and 2)distribution of door opening time in one day based on 15 days observation

a, sensor data tables

1 Window sensor	
Time stamp	state
08/18 6:39 a.m.	open
08/18 5:21 p.m.	close
08/19 7:24 a.m.	open
08/19 8:14 p.m.	close
08/20 8:05 a.m.	open
08/20 10:09 p.m.	close
08/21 6:33 a.m.	open
08/21 9:36 p.m.	close

2 Door sensor	
Time stamp	state
08/18 7:30 a.m.	open
08/18 4:22 p.m.	close
08/19 8:15 a.m.	open
08/19 4:16 p.m.	close
08/20 8:45 a.m.	open
08/20 4:19 p.m.	close
08/21 7:50 a.m.	open
08/21 4:26 p.m.	close

3 Temperature sensor data	
Time stamp	value
08/18 6:39 a.m.	27.5
08/18 7:30 a.m.	28.8
08/18 4:22 p.m.	19.9
08/18 5:21 p.m.	18.3
08/19 7:24 a.m.	26.8
08/19 8:15 a.m.	23.7
08/19 4:16 p.m.	21.9
08/19 8:14 p.m.	17.5
08/20 8:05 a.m.	28.4
08/20 8:45 a.m.	25.5
08/20 4:19 p.m.	12.4
08/20 10:09 p.m.	22.9
08/21 6:33 a.m.	22.5
08/21 7:50 a.m.	20.4
08/21 4:26 p.m.	15.7
08/21 9:36 p.m.	22.2

b, item ID mapping table

Events and clusters	Item ID
Window open	1
Window close	2
Door open	3
Door close	4
Cluster (27.5℃, 26.8℃, 28.4℃)	5
Cluster (18.3℃,17.5℃)	6
Cluster (22.9℃, 22,2℃)	7
Cluster (4:22 p.m., 4:16 p.m., 4:19 p.m., 4:26 p.m.)	8

c, create temporal database with item IDs

Date		08/18				08/19				08/20				08/21			
Time		6:39 a.m.	8:30 a.m.	4:22 p.m.	5:21 p.m.	7:24 a.m.	8:15 a.m.	4:16 p.m.	8:14 p.m.	8:05 a.m.	8:25 a.m.	4:19 p.m.	10:09 p.m.	6:33 a.m.	8:33 a.m.	4:26 p.m.	9:36 p.m.
Event		Win open	Door open	Door close	Win close	Win open	Door open	Door close	Win close	Win open	Door open	Door close	Win close	Win open	Door open	Door close	Win close
Temporal database	Item ID (event)	1	3	4	2	1	3	4	2	1	3	4	2	1	3	4	2
	Item ID (ambient)	5		8	6	5		8	6	5		8	7			8	7

Figure 3. Convert sensor data tables into temporal database

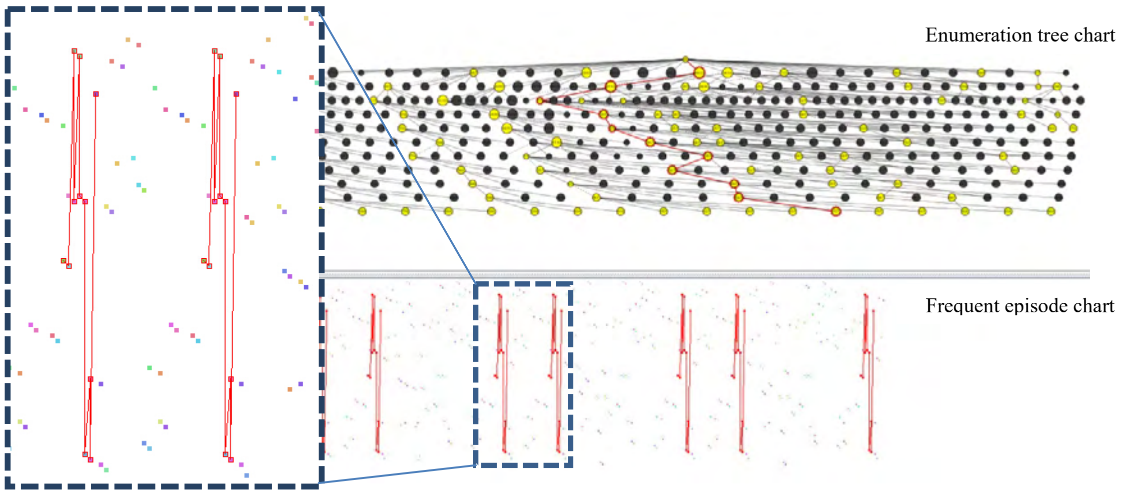


Figure 4. Enumeration tree chart and frequent episode chart

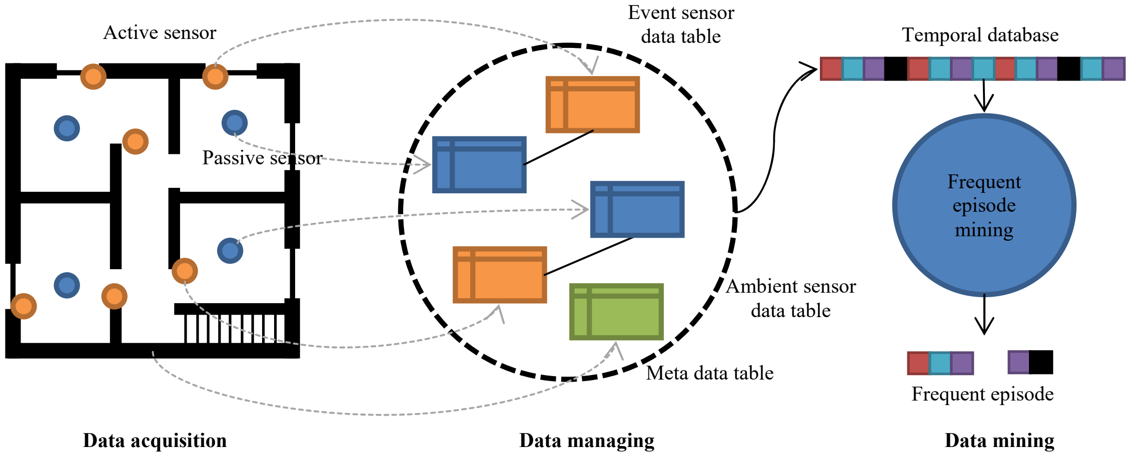
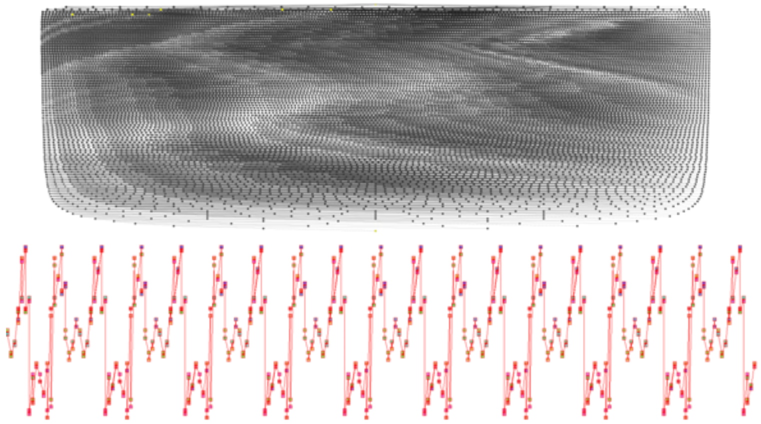


Figure 5. Diagram for the pattern discovery process

a.



b.

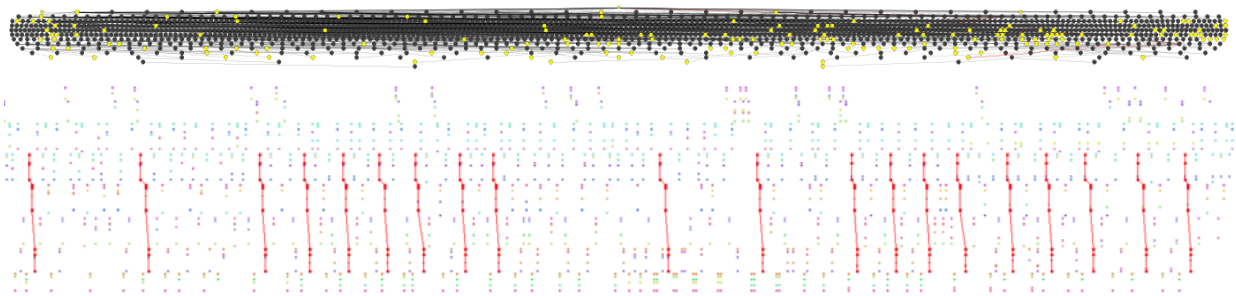


Figure 6. a. Enumeration tree and occurance of pattern 7 with synthetic data,
b.Enumeration tree and pattern <(25 28 34) (36 37 45) (59 61 67) > with real life data