International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 3, June 2023, pp. 3192~3203
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i3.pp3192-3203

Context-aware recommender system for multi-user smart home

Shymaa Sobhy, Eman M. Mohamed, Arabi Keshk, Mahmoud Hussein
Department of Computer Science, Menoufia University, Menofia, Egypt

Article Info

Article history:

Received Aug 4, 2022
Revised Sep 16, 2022
Accepted Oct 1, 2022

Keywords:

Data mining
Internet of things
Parallel computing
Recommender system
Smart homes
Unsupervised algorithms

ABSTRACT

Smart home is one of the most important applications of the internet of things (IoT). Smart home makes life simpler, easier to control, saves energy based on user's behavior and interaction with the home appliances. Many existing approaches have designed a smart home system using data mining algorithms. However, these approaches do not consider multiusers that exist in the same location and time (which needs a complex control). They also use centralized mining algorithm, then the system's efficiency is reduced when datasets increase. Therefore, in this paper, we firstly build a context-aware recommender system that considers multi-user's preferences and solves their conflicts by using unsupervised algorithms to deliver useful recommendation services. Secondly, we improve smart home's responsive using parallel computing. The results reveal that the proposed method is better than existing approaches.

Corresponding Author:

Shymaa Sobhy
Department of Computer Science, Faculty of Computers and Information, Menofia University
Menofia, Egypt
Email: shaymaa.abdelaal@ci.menofia.edu.eg

1. INTRODUCTION

Internet of things (IoT) world is a network of connected devices for exchanging information data and to control devices [1]. Devices with sensors and actuators combine objects on the Internet through human-to-tool interactions. A main application of IoT is the smart home that is powered with various IoT devices connected together [2]. The user can control the connected devices with smart home applications on any electronic devices such as smartphones. The usage behavior of the IoT devices in the smart home is important for collecting recurrent patterns relevant to the user. It can be viewed as a sequence of activities and interactions with various devices to reflect the user's intentions, actions and interactions patterns that need analysis [3], [4].

Recommender systems can exploit knowledge collected from the users' usage and offers some helping recommendations to them [5]. They give more comfort, help older and disabled individuals, and facilitate user's life. They have methods that facilitate users ' choices based on their preferences [6], [7]. Smart home recommendation applications become available to analysis the actions that matches user's interaction profile. They provide recommend services that user may need to perform. The goal is to help user in controlling their life in best and comfortable manner.

Recently, a number of recommender system that could be delivered to user in their smart life has been introduced. In [8] Quijano-Sánchez *et al.* introduce a system takes data of IoT devices to derive the most frequent patterns and recommends the other possible active patterns. The patterns are most related to the users and their usage of the IoT devices. In [9] Amatriain *et al.* introduce recommend actions that help elderly people by providing smart agents that assist them. In [10] Thakur and Han introduce a recommender system that address frequent choices of automation available for the user in his environment. In [11] Rasch introduce a recommender system offers services to user based on the interaction profiles offered by the real existing sensors. In [12] Belghini *et al.* introduce a recommender system with supervised learning algorithms to predict user action in the current contextual profile and provide recommendations. In [13] Gupta develop a predictive model for location-aware optimizing resources in smart homes user that enhance his comfort. In [14] Roy *et al.* introduce a method for context-aware services that adapt all corners of living part of home. In [4] Nawara and Kashef introduce a recommender system help user when requires recommendations of nearby places based on the ranking function. In [15] Miraoui introduce recommendations in a smart kitchen by construct contextual profiles provide useful recipe options for the user.

The related works mentioned before provide useful services like helping elderly people by providing smart agents, location-aware resources optimization in smart homes, predict next action and construct interaction profiles that help user in his smart kitchen or living room or any part of home. However, they have limitations when multiple users exist in same time in the same location. It results in complex scenarios with conflicts between users in using different devices in home and how to deliver useful services for the proper user related their priority on using devices. Another limitation is the use of traditional centralized data mining algorithms in learning preferences. Such centralized algorithms are efficient in small datasets but when datasets increased their efficiency is reduced.

This paper proposes a method to improve smart homes performance to be more interactive with multiusers. It builds a context-aware recommender system that learns useful multi-user's preferences by using unsupervised algorithms and delivers useful recommendation services. It starts by simulating complex multi user's environment contexts in a virtual smart home tool to capture sensors data and construct multiusers history of interactions is in the datasets. Then, it preprocesses multiusers datasets to manage conflicts occurred by removing any redundancy. Finally, it constructs contextual information such as location, time, and type of the day that are extracted from the previously collected data, and merge it with multiusers preferences to provide useful context aware recommend for multiuser home devices. This paper also proposes an approach to improve smart homes performance to be more responsive by using parallel computing that solve challenges of traditional data mining algorithms in term of time and memory. These algorithms used decentralized in learning process to find the most frequent activities and association rules of multi users in smart home. The experimental results of the proposed method yielded good results in providing useful recommendation for multiuser smart homes without conflicts between users complex situations. The smart homes also become more responsive by reducing execution time.

In brief, the contributions of this paper are: i) making smart homes more interactive by build context-aware recommendation system that provides useful recommendation for multiuser smart homes; and ii) make smart homes more responsive using parallel computing that accelerate performance of learning algorithms. The remainder of this paper is ordered as. The method is introduced in section 2. Section 3 provides the results and discussion. Section 4 presents the conclusion.

2. METHOD

This paper introduces a method that enables smart homes to be more responsive and interactive with users. It recommends services for the user according to their current situation such as: turning off lights when user is sleeping, turning on television when user exist in couch on the living room, turning on kitchen light when user opens the kitchen door. In the following, the related work to the proposed method introduced then the proposed method in brief is explained followed by method steps explanation in detail.

2.1. Related work to the proposed method

The recommender system starts by the training phase in which smart home learns a model of the user activities. After the training phase, the system analyzes user's situation and generates personalized recommendations [16]. The system recommends services that help users in the future actions. There are many algorithms for learning user's actions like deep learning, machine learning, collaborative filtering, content-based, knowledge-based and context-aware approach (that define the context related to the application by different users) [17]. Context-aware recommendations create better user experiences to improve performance in carrying out activities based on contexts such as location, identity, and actions of users and physical sensors on devices with applications [18], [19]. They are aware of actions, their attributes and possible activities that a user may perform by analyzing the user's behavior and nature of doing various actions [20], [21]. Data mining algorithms are classification; recommendation systems, clustering, and association rule mining are used frequently to save useful or relevant information from massive datasets [22]. In the following, we review some of the existing techniques for smart home's recommender system.

An action recommender system for IoT smart homes has been introduced [8]. The system takes data of IoT devices as input then applies a pattern-mining algorithm to derive the most frequent and active recurrent patterns of single user. Finally, it recommends the other possible actions from any active patterns.

The resulted active patterns are most related to the users and their usage of the IoT devices making the recommendations personalized. The system uses association rule mining for mining out recurrent patterns.

A framework was developed for generating a database that capture all possible user actions in any activity and tries to find the activity goal [9]. They present an activity performance supervised recommender system to recommend actions related to various activities of single user. They help elderly people by providing environments with smart agents that can effectively assist them in user interactions and provides useful services and gives recommend help in their current action.

The method proposed in [10] identified challenges that are useful to user of the smart homes like difficult interfaces, configuration problems. It solved the challenges by proposing a recommender system that addresses frequent choices of automation available for the user in the current situation. In [11], Rasch offer the most important actions to user based on the interaction profile provided by the real sensors. They recommend to use unsupervised learning such as association analysis to mine the rules from a frequent item set by setting a minimum frequency threshold on the recurring items found.

A framework for finding useful user preferences and delivers relevant context-aware recommendations is designed [12]. It transforms data into useful format by applying unsupervised algorithms to find frequently occurring activities and to learn useful rules over those activities. Then, it constructs contextual profile from the raw data and adding it to the rules mined. Finally, it applies supervised learning algorithms to predict user action in the current contextual profile and provide recommendations.

A model that specifies location-aware optimizing resources in smart homes has been introduced [13]. It offers efficient learning of user location's context. Efficient prediction helps in controlling devices with operations of automated services along user next location. Simulation results of the model on virtual smart home indicates high prediction success and gives high optimization of energy consumption, operations, and time of users to enhance their comfort.

A framework for context-aware services that simulate design of home living part has been proposed [14]. In this framework, algorithms of machine learning such as naïve Bayes, fuzzy logic and case-based reasoning techniques are used. Results show they improve the smart home user life and optimize their life.

A location-based context aware recommender method that finds a ranking function to offer top-k services to the user has been introduced [4]. The contextual information is constructed by users as generated rules using a rule-based language (RuleML). When user requires recommendations of the nearby places, contextual data are constructed and rules is extracted. The top-k recommendations of nearby places based on the ranking function are presented to the user.

The method proposed in [15] defines the context as sources to build a recommender model to offer recommendations in a smart kitchen. It uses recorded interaction data to predict and recommend services to the user. The recommendation model has three processes: food kind's taxonomy, recipe taxonomy, and condition feature data. The first process is to decide the food sections exist in refrigerator and cupboard in the smart kitchen. Second, the model finds only those recipes which can be provided from the existing kinds in the smart kitchen out of all the possible recipes. Finally, contextual profiles provide useful recipe options for the user.

Table 1 provides a summary of related works. The summary includes the used mining algorithm, IoT application, collect data is real or from smart home's simulation, contextual data are considered or not, parallel computing or centralized approach is used, and multi users in smart environment are handled or not. The table shows that related work uses data unsupervised or supervised algorithms for learning and predictions in real or simulated smart home environment. But, few of them add contextual information.

The related works focus on building a model for recommender system that help user in his smart environment. However, none of the models deals with a high number of users and handles cases where lot of activities is interleaved (i.e., complex activity definitions for multiusers in smart home). Also, they use centralized data mining algorithms that find frequent user actions and gives useful recommendations for next actions. Such algorithms are not efficient in larger context.

Table 1. The analysis of related works

RefNum	Mining algorithm	Application	Focus	Context information	Multiusers	Parallel
[8]	Unsupervised learning	smart home	Real	No	No	No
[9]	Supervised learning	smart home	Real	No	No	No
[10]	Unsupervised learning	smart home	Real	No	No	No
[11]	Unsupervised learning	smart home	Real	Yes	No	No
[12]	Association prediction	smart homes	Simulation	Yes	No	No
[13]	Prediction algorithm	smart homes	Visualization	No	No	No
[14]	Machine learning	living room	Visualization	No	No	No
[4]	Ranking function	smart homes	Real	Yes	No	No
[15]	Machine learning	smart kitchen	Visualization	Yes	No	No

2.2. The processes of the proposed method

The processes of the proposed method are shown in Figure 1. It starts with capturing sensors data. Then, history of interaction formalization is recorded in the datasets and data is pre-processed to address the complex situation of multi users. After that, conflicts are managed; frequent items and association rules are learned with Apriori algorithm in parallel. Finally, contextual profile like location and time of the day are added to the rules to provide useful services for smart applications.

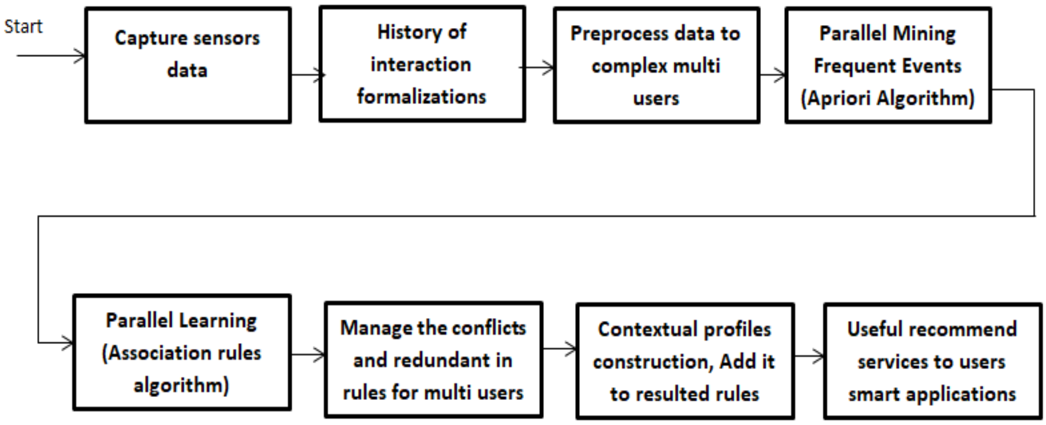


Figure 1. Process of the proposed method

2.2.1. Capture sensors data

Serving the user in an effective manner in his smart environment requires datasets for recording the interaction captured from a smart home. Figure 2 shows the techniques for generating datasets for smart home for data analysis. It shows two main methods: real and simulation. When building real smart home, many challenges are faced like i) having a robust and continuous collecting techniques for the sensors data and ii) finding the best annotation techniques for user actions. Simulation tool handle these limitations of real datasets by dataset generation with robust algorithms to capture sensor data. The simulation tools design can be done using three ways: model-based, interactive and hybrid approaches. The model-based uses statistical models to generate datasets. The interactive method relies on real-time using an avatar navigates by simulated person. The hybrid combines model-based and interactive approaches. The targets from simulation tool are visualization, dataset generation and context-awareness.

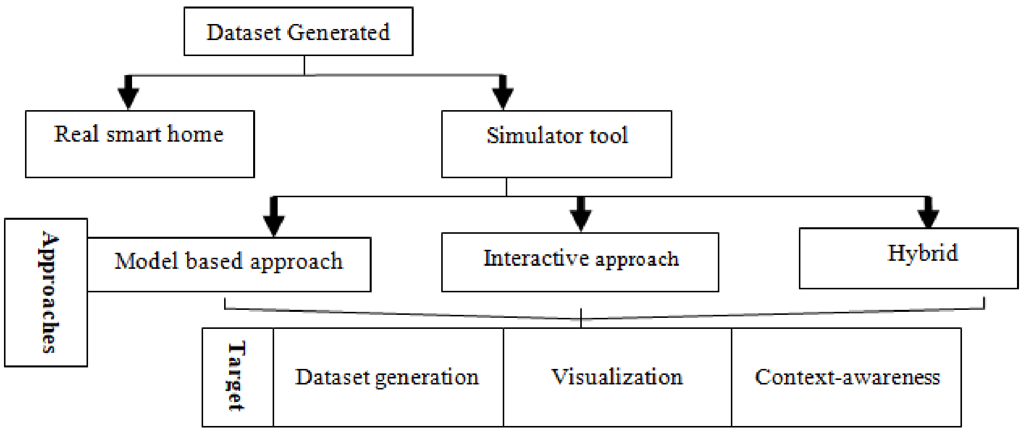


Figure 2. Techniques for datasets generation

Table 2 compares smart home simulation tools. They are compared by simulation technique, free access, scripting availability in simulation, multi users or single user and time related considering. As shown in the table, most of the existing tools problems are not open source and none of them handle multiuser.

In our proposed method, personal data are collected in a virtual environment as real datasets are limited in numbers. The dataset has been collected by an open source simulator called OpenSHS that attempt to create a realistic dataset to facilitate research in smart home design [3]. The users simulate various actions of their day and simulator records the state of sensors placed in different rooms of the home. The OpenSHS tool design is a bedroom, kitchen, living, hallway, bathroom, and office. The smart home is installed with twenty-nine binary sensors places at different objects such as television, lights which offers the state of the object on which is placed. List of sensors name, type and description in OpenSHS datasets are showed in Table 3. The simulator unfortunately is designed for single user as it has one dataset for all interactions of that user. Thus, we simulate the interaction for multi users separately. Then, the collected datasets are edited to handle location of each user in the time of that context and inject some conflicts between all users (i.e., the users exist at the same time and location). Finally, we convert all datasets to indicate the actions of all users (see the details in the preprocessing step).

Table 2. Smart home simulation tools comparison

RefNum	Simulation tool	Simulation technique	Open source	scripting	Multiusers	Time related
[23]	Home I/O	interactive	NO	NO	NO	YES
[3]	OpenSHS	Hybrid	YES	NO	NO	YES
[4]	SIMACT	Model based	YES	YES	NO	NO
[24]	IE Sim	interactive	NO	YES	NO	YES
[25]	SE SiM	Model based	YES	YES	NO	NO
[26]	Persim 3D	Model based	NO	NO	NO	NO

Table 3. Design of OpenSHS simulator

Sensor_Num	Sensor name	Sensor type	Sensor description
1	Wardrobe	binary	bedroom wardrobe sensor
2	tv	binary	living TV sensor
3	oven	binary	kitchen oven sensor
4	officeLight	binary	office ceiling light sensor
5	officeDoorLock	binary	office door lock sensor
6	officeDoor	binary	office door sensor
7	officeCarp	binary	office carpet sensor
8	office	binary	office room office sensor
9	mainDoorLock	binary	main door lock office
10	mainDoor	binary	main door sensor
11	livingLight	binary	living ceiling light sensor
12	livingCarp	binary	living carpet sensor
13	kitchenLight	binary	kitchen ceiling light sensor
14	kitchenDoorLock	binary	kitchen door lock sensor
15	kitchenDoor	binary	kitchen door sensor
16	kitchenCarp	binary	kitchen carpet sensor
17	hallwayLight	binary	hallway ceiling light sensor
18	fridge	binary	kitchen fridge sensor
19	couch	binary	living couch sensor
20	bedroomLight	binary	bedroom ceiling light sensor
21	bedroomDoorLock	binary	bedroom door lock sensor
22	bedroomDoor	binary	bedroom door sensor
23	bedroomCarp	binary	bedroom carpet sensor
24	bedTableLamp	binary	bathroom bed Table lamp
25	bed	binary	bathroom bed sensor
26	bathroomLight	binary	bathroom ceiling light sensor
27	bathroomDoorLock	binary	bathroom door lock sensor
28	bathroomDoor	binary	bathroom door sensor
29	bathroomCarp	binary	bathroom carpet sensor
30	Activity	String	current user activity sensor
31	Timestamp	String	Timestamp every second

2.2.2. History of interaction formalization

In this step, the history of interactions is constructed and saved on databases. The proposed approach uses OpenSHS tool to save all users interactions by sensor names to datasets that has a timestamp column of datetime. Figure 3 present the design phase at the top and simulation phase at the bottom with time stamp. The datasets also has an activity column which divides the activity of the user such as eat, sleep, walk and others. OpenSHS datasets generation is hybrid model that creates rich realistic datasets depends on replication algorithm [3]. Two elements were used, the researcher and the participant. Most of the work done by OpenSHS is done by researchers. Participants are anyone who voluntarily simulates their activities.

Three main phases: the design phase, the simulation phase, and the aggregation phase. In design Phase researchers create a virtual environment, import smart devices, assign actions to tags, and develop context. After developing the smart home model, the researcher creates a context for the simulation. Context is the specific time range that researchers are interested in modeling, for example, situations in the morning, afternoon, and evening. In simulation phase the researcher to specify which context to simulate. Each context has the default state of the sensor and position of the participant. Then the participants started to simulate their Activities in this situation. During the simulation, the sensor output and the status of various devices are recorded and saved in a temporary data set. The modeling or simulation phase aims to capture the details of the actual interaction between participants. In aggregation phase the researcher can aggregate the participants' generated sample activities such as create the final recorded event. The result of the modeling phase forms a series of typical actions for each context.

The aggregation stage aims to provide a solution for generating large data sets in a short modeling time. The replication algorithm used to reproduce the results of the simulation phase by extracting enough samples for each expected context. This feature allows OpenSHS to combine the advantages of both approaches, a hybrid approach. OpenSHS depend on game engine of on Blender with Python editors. Blender is available for all three major operating systems. Blender uses OpenGL for its Game Engine which is also, a cross-platform 3D technology available for the major operating systems. The Blender Game Engine: The physics engine facilitates the simulation of different types of real sensors and devices. For example, Blender has a (Near) sensor that only triggers when the user-controlled 3D avatar is physically close to other objects in the scene. All logic and interactions between the avatar and the virtual environment are developed with it. In addition, all OpenSHS modules are programmed in Python [27].

Figure 3. The design phase and simulation phase

2.2.3. Preprocessing datasets for multiusers environment

The proposed approach preprocesses OpenSHS row data to work with multiusers without conflicts between users in the same time and location. Table 4 shows parts of datasets contents after preprocessing with devices off. For example, first user actions on devices like television in living and kitchen light in kitchen can be either on or off. Datasets represented in binary format as 1 means devices state is open and 0 means that devices is off. Also, on locations like living or kitchen 1 mean user are exit on that location and 0 mean that he is not exist. The steps of dataset preprocessing are: i) edit sensor names for all users' datasets with user's id; ii) insert location column based on five sensors called kitchen, bathroom, bedroom, office, living, and hallway. This captured from the sensors states located in the carpets of every part of the house, representing presence of the users in that part; and iii) convert sensor states that are saved in binary format of zeros and ones to represent the user's actions on smart home devices like kitchen light is on and kitchen light is off by add more columns to some sensors in dataset that represent the off action of devices.

Table 4. The datasets of first user preprocess with devices off and on and location

u1 tv off	u1 tv on	u1 oven off	u1 oven on	u1office Light on	u1office Light off	u1kitchen Light on	u1kitchen Light off	u1 Office	u1 Living	u1 Kitchen	u1 Bedroom	u1 Bathroom
1	0	0	0	1	0	0	1	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	0	1	0	1	1	0

2.2.4. Find frequent events in parallel

The purpose of data mining techniques is to extract a valuable information and knowledge from huge amounts of data stored in databases. It is the process of finding out formerly unknown, useful and valuable patterns from a large amount of data stored in a database. Data mining examines databases for identifying hidden patterns and valuable information that sometimes experts may not observe as it occurs outside their expectations [13]. The discovered patterns are accessible to the user and could be stored as new information in the information database. Classical Apriori algorithm is for learning data mining association rule and discovers association among different set of data [14].

Searching for most occurring itemsets is one of the most targets for data mining fields [28]. Apriori algorithm is the most efficiently used technique for specifying most occurring itemsets from a transactional dataset [29], [30]. However, the Apriori algorithm searches the dataset a lot of times for specifying candidate itemsets. But, when the dataset becomes very big, both space used and computational cost can be more expensive. Memory and CPU resources are unfortunately very limited, which make execution of the algorithm inefficient with low performance. Thus, the proposed method use multi-processing to execute Apriori algorithm in parallel [31], [32] with the following steps of parallel mining to get frequent events for each user datasets: i) define number of parallel process to run as N process, ii) divide the input file into different N parts for parallel processing, iii) start function that calculated the frequent dataset parallel, and iv) create the pool object to start mapping the input to different process object and join parts.

2.2.5. Learning association rules in parallel

To find rules that specify relation between frequent items with association rules in parallel as frequent items list size is huge [33]–[35]. In the following steps the algorithm use multi-processing to execute association algorithm. This is performed in parallel are used for each resulted frequent items from previous step by Apriori algorithm. We present here the steps of association rule parallel learning: i) splitting the input frequent itemsets resulted from Apriori algorithm into a list of subset data frames; ii) map the list of subset data frames with pool map function that take frequent patterns as input and the min threshold to calculate the rules and return rules; and iii) join resulted rules lists in one list and print rules with antecedents and consequences.

Figure 4 shows a flowchart for Apriori and association rule algorithms for single data set, while the parallelization of these algorithms is shown in Figure 5. The Apriori algorithm set a minimum value for support and confidence. It then reads each transaction and finds rules for the items that have certain default existence support and have a minimum value for co-occurrence with other items confidence. Later on, it extracts all the subsets having higher value of support than minimum threshold, and select all the rules from the subsets with confidence value higher than minimum threshold. This is done for all datasets in parallel by multiprocessing as explained before.

2.2.6. Manage conflicts between resulted rules

Merging all users' rules in one data frame results in conflicts between users exists in same locations at the same time. The proposed method handles it by the following steps: i) search merged rules to specify locations for the different users exist on same time; ii) separate the rules into smaller ones based on locations founded in step one; iii) for all resulted sets of rules convert it from many antecedents and many consequents to one to one; iv) remove resulted duplication in rules; v) set for all rules different priority for active devices that indicates who can access it if multi users exist at the same time; and vi) select user with high priority on each active device on the specified location.

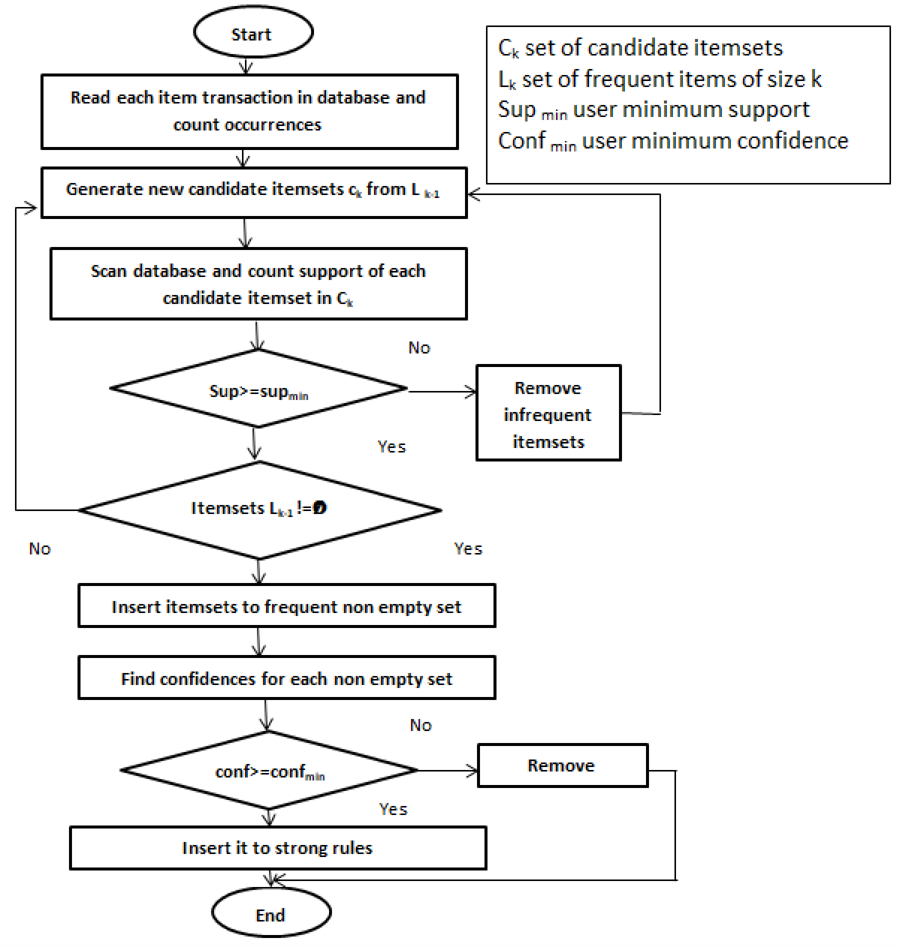


Figure 4. Flowchart for Apriori and association rule algorithms for single dataset

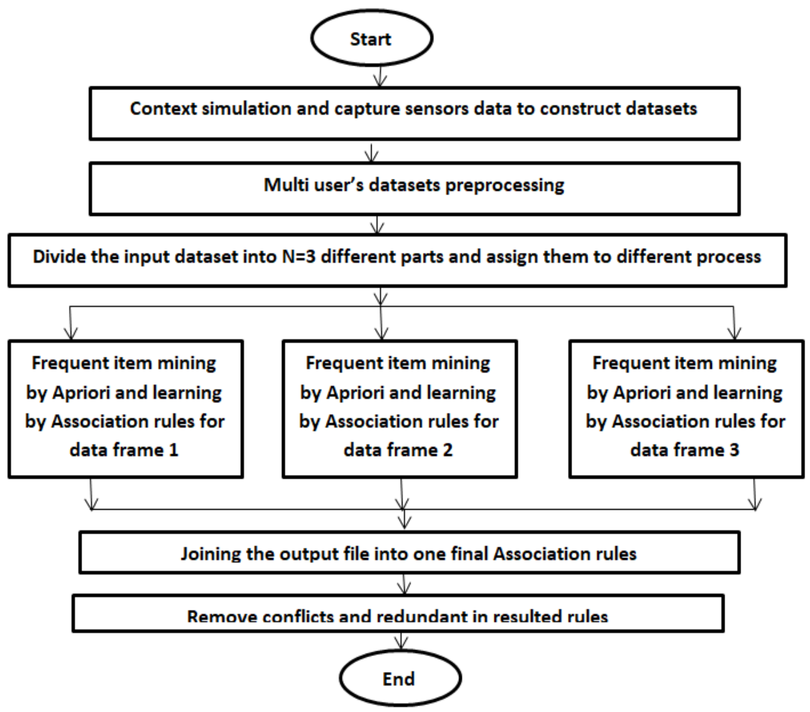


Figure 5. Flowchart for parallel computing of Apriori and association rule algorithms

2.2.7. Merge multiusers rules and contextual profile

Many related works ignore contextual profile construction. But it is very important to make the simulation aware of environment and offer information useful to the users that help him in his life. The proposed method uses OpenSHS row data and analysis to construct context profiles. Along with the activity column already placed in OpenSHS row dataset, three contexts proposed: i) time of the day as hours of the day are classified into six parts of the day, ii) type of the day specifying all days on the week by separated it to a weekday or weekend, iii) user location calculated by the sensors states located in the carpets of every part of the home rooms.

2.2.8. Recommending useful context-aware services for smart applications

After adding contextual information to row data and get frequent rules with consequent and antecedent. They are collected in one file indicates each data set columns with resulted consequent. This file used to automate services like IF (U1ovenON) Then (U1kitchenlighton). Table 5 show subset of resulted rules antecedents and consequents for kitchen location after running parallel Apriori and association rules of multi users of tree different users and managing conflict between then as found that user1 and user2 found in the same time in kitchen so we give priority to user on kitchen light and oven. Then, the consequent column adds to the original data to be used in service automation and prediction of supervised algorithms.

Table 5. The resulted rule for kitchen location

RuleNum	Antecedents	Consequents
0	u1ovenon	u1kitchenLighton
1	u1Kitchen	u1kitchenLighton
2	u1Kitchen	u1kitchenLightoff
3	u1ovenoff	u1kitchenLightoff
6	u3mainDoorLockon	u3kitchenDooron
7	u3Kitchen	u3kitchenDooron
8	u3kitchenDooron	u3mainDoorLockon
9	u3Kitchen	u3mainDoorLockon
10	u3mainDoorLockon	u3fridgeoff
11	u3Kitchen	u3fridgeoff
12	u3mainDoorLockon	u3bedroomDooron
13	u3Kitchen	u3bedroomDooron
14	u3bedroomDooron	u3mainDoorLockon
20	u3kitchenDooron	u3fridgeoff
22	u3bedroomDooron	u3kitchenDooron
24	u3kitchenDooron	u3bedroomDooron
26	u3bedroomDooron	u3fridgeoff
51	u3fridgeoff	u3mainDoorLockon
76	u3fridgeoff	u3kitchenDooron
114	u3fridgeoff	u3bedroomDooron

3. RESULTS AND DISCUSSION

To evaluate the proposed method, the experimental setting is Intel core i5 CPU and 8.00 GB RAM on 64-bit operating system to build a context-aware recommendation system for multiuser smart homes in simulation environment tool called OpenSHS (built by Python and blender). The anaconda jupyter notebook Python are used to implement mining algorithms and Python multi-processing was used to improve performance with parallel execution. This section firstly introduces datasets resulted for multiusers. Secondly, experimental results for parallel computing of Apriori and association rules are introduced.

3.1. Datasets for multiusers

Table 6 show results of applying the proposed method of converting simulation row data into multiuser data, and remove any duplication or conflict if two or more user exist in the same location. Three users in home and dataset records them in four different locations so conflict is found between users as Table 6 shows. The proposed method separates the rules of three users and for each location it specifies user's priority on active devices. Table 6 includes 168 frequent items and 1,811 rules for user one, and for user two there are 104 frequent items and 526 rules, and 76 frequent items and 486 rules for user three. The total merged rules for all users are 2,823 rules.

Table 7 Show results of conflict removing for merged rules for all users that equal 2,823 rules. When they are separated into four different locations, the rules are 130 for office, 641 for living, 152 for kitchen, and 900 for bedroom with total 1,823 the other rules appears when one user exists. The rules are then cleaned by removing duplication. The rules become 17 for office, 66 for living, 29 for kitchen, and 177 for bedroom with total 289 rules. Finally, after specifying which user can access devices by setting priority, the final rules are 12 for office, 42 for living, 20 for kitchen, 80 for bedroom (i.e., the total rules are 154) as shown in Table 7 so results indicates reducing rules from 1,823 to 154 rule without conflicts.

Table 6. The merged rules for three users

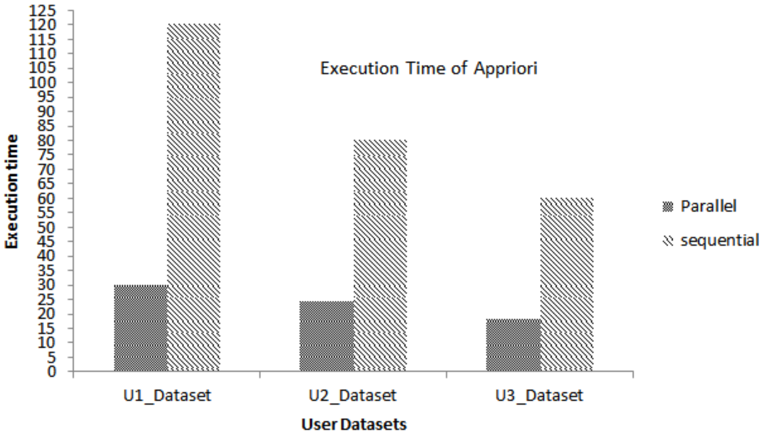
NUM users	U1	U2	U3
Frequent items	168	104	76
Association rules	1811	526	486
Merged rules for all users=2823			

Table 7. The conflict removing for merged rules

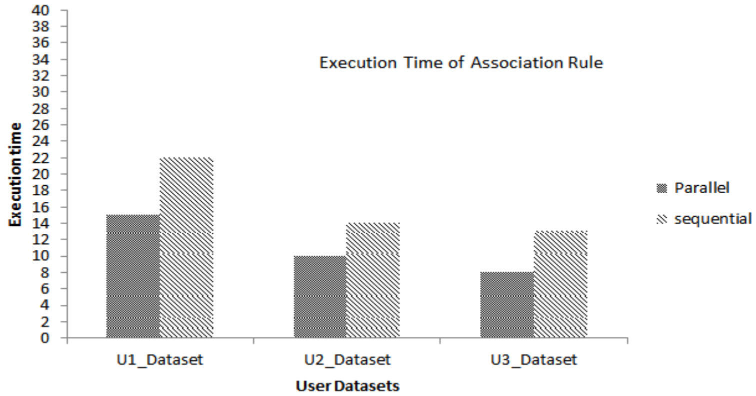
	Location office	Location living	Location kitchen	Location bedroom	Total
Merged rules number	130	641	152	900	1823
Rules number after duplication remove	17	66	29	177	289
Rules number after setting users priority	12	42	20	80	154

3.2. Parallel and sequential execution time of Apriori and association rules

This section shows results for the execution time for Apriori and association rules of the three different datasets for different context with three users. The execution time is 45 ms in the parallel case while it is 142 ms in sequential for dataset one. Second dataset requires 34.2 ms during the parallel execution, while 94 ms in the sequential case. Third dataset has in total 26 ms in parallel case and 73 ms during sequential execution. Figure 6 comparing three different users datasets for parallel and sequential execution time in (a) Apriori algorithm and (b) association rules.



(a)



(b)

Figure 6. Comparing three different users datasets for parallel and sequential execution time in (a) Apriori algorithm and (b) association rules

4. CONCLUSION

This paper proposed method to improves smart home performance to be more responsive to his users. It built a model that extracts useful user-preferences and delivers context-aware recommendations. The proposed method makes smart home more interactive/responsive with users by the use of parallel computation of association rule algorithm. The proposed method transforms OpenSHS simulator data into suitable format and detects frequently occurring events. Then, it extracts and combines the context profile of interaction to associate it with resulted rules. The proposed method introduces method for learning multi-user preferences with multi device that allows smart home users to efficiently use the appliances. The proposed method finds frequent items and association rules for multiple datasets for all users. It extracts meaningful rules without conflict between the multi-users in the same time and location in the smart home. Experiments on a massive dataset demonstrated outstanding applicability of this algorithm. As a future work, we plan to improve a security model for smart home that is remotely controlled via web application. This can be done through authorization by access control and considering contextual information like location, time of request, and history of user action.

REFERENCES

[1] P. Sethi and S. R. Sarangi, "Internet of things: architectures, protocols, and applications," *Journal of Electrical and Computer Engineering*, pp. 1–25, 2017, doi: 10.1155/2017/9324035.

[2] M. Domb, "Smart home systems based on internet of things," in *IoT and Smart Home Automation [Working Title]*, IntechOpen, 2019.

[3] N. Alshammari, T. Alshammari, M. Sedky, J. Champion, and C. Bauer, "OpenSHS: open smart home simulator," *Sensors*, vol. 17, no. 5, May 2017, doi: 10.3390/s17051003.

[4] D. Nawara and R. Kashef, "IoT-based recommendation systems-an overview," in *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, Sep. 2020, pp. 1–7, doi: 10.1109/IEMTRONICS51293.2020.9216391.

[5] K. Rasch, "Smart assistants for smart homes," KTH Royal Institute of Technology, 2013.

[6] Y. Himeur *et al.*, "A survey of recommender systems for energy efficiency in buildings: Principles, challenges and prospects," *Information Fusion*, vol. 72, pp. 1–21, Aug. 2021, doi: 10.1016/j.inffus.2021.02.002.

[7] L. Yao, X. Wang, Q. Z. Sheng, S. Dustdar, and S. Zhang, "Recommendations on the internet of things: requirements, challenges, and directions," *IEEE Internet Computing*, vol. 23, no. 3, pp. 46–54, May 2019, doi: 10.1109/MIC.2019.2909607.

[8] L. Quijano-Sánchez, I. Cantador, M. E. Cortés-Cediel, and O. Gil, "Recommender systems for smart cities," *Information Systems*, vol. 92, Sep. 2020, doi: 10.1016/j.is.2020.101545.

[9] X. Amatriain, N. Oliver, and J. M. Pujol, "Data mining methods for recommender systems," in *Recommender systems handbook*, Springer, 2011, pp. 39–71.

[10] N. Thakur and C. Y. Han, "A context-driven complex activity framework for smart home," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Nov. 2018, pp. 801–806, doi: 10.1109/IEMCON.2018.8615079.

[11] K. Rasch, "An unsupervised recommender system for smart homes," *Journal of Ambient Intelligence and Smart Environments*, vol. 6, no. 1, pp. 21–37, 2014, doi: 10.3233/AIS-130242.

[12] N. Belghini, N. Gouttaya, W. Bouab, and A. Sayouti, "Pervasive recommender system for smart home environment," *International Journal of Applied Information Systems*, vol. 10, no. 9, pp. 1–7, May 2016, doi: 10.5120/ijais2016451528.

[13] S. Gupta, "A context-aware personalized recommender system for automation in IoT based smart home environment," Dublin Business School, 2020.

[14] A. Roy, S. K. Das Bhaumik, A. Bhattacharya, K. Basu, D. J. Cook, and S. K. Das, "Location aware resource management in smart homes," in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003).*, 2003, pp. 481–488, doi: 10.1109/PERCOM.2003.1192773.

[15] M. Miraoui, S. El-etriby, C. Tadj, and A. Zaid Abid, "A hybrid modular context-aware services adaptation for a smart living room," *Intelligent Automation and Soft Computing*, vol. 24, no. 2, pp. 299–308, Jun. 2018, doi: 10.1080/10798587.2017.1281565.

[16] Pratibha and P. D. Kaur, "A context-aware recommender engine for smart kitchen," in *Smart Innovations in Communication and Computational Sciences*, Springer Singapore, 2019, pp. 161–170.

[17] P. Mishra, S. K. Gudla, A. D. ShanBhag, and J. Bose, "Alternate action recommender system using recurrent patterns of smart home users," in *2020 IEEE 17th Annual Consumer Communications and Networking Conference (CCNC)*, Jan. 2020, pp. 1–6, doi: 10.1109/CCNC46108.2020.9045128.

[18] H. F. Abdulkarem, G. Y. Abozaid, and M. I. Soliman, "Context-aware recommender system frameworks, techniques, and applications: a survey," in *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*, Feb. 2019, pp. 180–185, doi: 10.1109/ITCE.2019.8646564.

[19] M. Altulyan, L. Yao, X. Wang, C. Huang, S. S. Kanhere, and Q. Z. Sheng, "Recommender systems for the internet of things: a survey," 2020, *arXiv:2007.06758*.

[20] A. Alotaibi and C. Perera, "Smart home human activity simulation tool for openhab-based research," *Tech. Rep. ACM*, 2019.

[21] S. Faieq, A. Front, R. Saidi, H. El Ghazi, and M. D. Rahmani, "A context-aware recommendation-based system for service composition in smart environments," *Service Oriented Computing and Applications*, vol. 13, no. 4, pp. 341–355, Dec. 2019, doi: 10.1007/s11761-019-00277-7.

[22] A.-K. Moon, H.-H. Kim, J.-Y. Park, and Y.-I. Choi, "Context based user profile for personalization in ubiquitous computing environments," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 34, no. 58, pp. 542–551, 2009.

[23] I. B. B. Gottfried and H. Aghajan, "Learning about preferences and common behaviours of the user in an intelligent environment," *Behaviour Monitoring and Interpretation-BMI: Smart Environments*, vol. 3, 2009.

[24] B. Riera and B. Vigário, "HOME I/O and FACTORY I/O: a virtual house and a virtual plant for control education," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9144–9149, Jul. 2017, doi: 10.1016/j.ifacol.2017.08.1719.

[25] J. Synnott, L. Chen, C. Nugent, and G. Moore, "IE Sim-A flexible tool for the simulation of data generated within intelligent environments," in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2012, pp. 373–378.

[26] B. Ho, D. Vogts, and J. Wesson, "A smart home simulation tool to support the recognition of activities of daily living," in *Proceedings of the South African Institute of Computer Scientists and Information Technologists 2019 on ZZZ - SAICSIT '19*, 2019, pp. 1–10, doi: 10.1145/3351108.3351132.

[27] J. W. Lee, S. Cho, S. Liu, K. Cho, and S. Helal, "Persim 3D: context-driven simulation and modeling of human activities in smart spaces," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1243–1256, Oct. 2015, doi: 10.1109/TASE.2015.2467353.

[28] J. Synnott, C. Nugent, and P. Jeffers, "Simulation of smart home activity datasets," *Sensors*, vol. 15, no. 6, pp. 14162–14179, Jun. 2015, doi: 10.3390/s150614162.

[29] X. Yuan, "An improved Apriori algorithm for mining association rules," 2017, doi: 10.1063/1.4977361.

[30] J. Dongre, G. L. Prajapati, and S. V Tokekar, "The role of Apriori algorithm for finding the association rules in data mining," in *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, Feb. 2014, pp. 657–660, doi: 10.1109/ICICICT.2014.6781357.

[31] P. Stanišić and S. Tomović, "Apriori multiple algorithm for mining association rules," *Information Technology and Control*, vol. 37, no. 4, 2008.

[32] S. Biswas, N. Biswas, and K. C. Mondal, "Parallel and distributed association rule mining algorithms: a recent survey," *Information Management and Computer Science*, vol. 2, no. 1, pp. 15–24, Sep. 2019, doi: 10.26480/imcs.01.2019.15.24.

[33] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Procedings 20th International Conference Very Large Data Bases*, vol. 1215, 1994, pp. 487–499.

[34] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of The 1993 ACM SIGMOD international Conference on Management of Data-SIGMOD '93*, 1993, pp. 207–216, doi: 10.1145/170035.170072.

[35] A. Savasere, E. R. Omiecinski, and S. B. Navathe, "An efficient algorithm for mining association rules in large databases," in *Proceedings of the 21st International Conference on Very Large Data Bases*, 1995, pp. 432–444.

BIOGRAPHIES OF AUTHOR

Shymaa Sobhy is a Ph.D. student at Menoufia University Faculty of Computers and Information, Egypt. She works Teaching Assistant at Computer Science Department, Menoufia University, Egypt. She received the B.S. Degree in Computer Science Faculty of Computers and Information, Menoufia University in May 2012 and an M.S. Degree in Computer Science from Menoufia University, Faculty of Computers and Information in 2018. Her main research interest is in software engineering, data mining, database, internet of things, data privacy, security, web application, artificial intelligence and recommender system. She can be contacted at email: sshh_yy_mmaa@yahoo.com.

Eman M. Mohamed received her B.Sc. in Computer Science from Menoufia University, Faculty of Computers, and Information in 2008. She received her M.Sc. in Computer Science from Menoufia University, Faculty of Computers, and Information in 2012. She received his Ph.D. in computer science from Menoufia University, Faculty of Computers, and Information in 2019. She works a Teacher at Computer Science Department, Menoufia University, Egypt. Her research interest includes cloud computing, big data, distributed system, database, bioinformatics, data privacy, and security. She can be contacted at email: eman.mohamed@ci.menofia.edu.eg.

Arabi Keshk received the B.Sc. in Electronic Engineering in Computer Science and Engineering from Menoufia University, Faculty of Electronic Engineering in 1987. He received the M.Sc in Electronic Engineering in Computer Science and Engineering from Menoufia University, Faculty of Electronic Engineering in 1995. He received his Ph.D. in Electronic Engineering from Osaka University, Japan in 2001. He works as Associate Professor Head of Computer Science Department 2009. He works as dean of Faculty of Computers and Information, Menoufia University at 2016. He works as the present of Delta Technological University in 2022. His research interest includes cloud computing, big data analytics, distributed systems, digital systems testing, software testing and data security. He can be contacted at email: arabikeshk@yahoo.com.

Mahmoud Hussein received his B.Sc. in Computer Science from Menoufia University, Faculty of Computers and Information in 2006. He received his M.Sc. in Computer Science from Menoufia University, Faculty of Computers and Information in 2009. He received his Ph.D. in Software Engineering from Swinburne University of Technology, Faculty of Information and Communications Technology in 2013. He works as Teaching Assistant at Menofia University in 2007. He works Assistant Lecturer at Menofia University in 2009. He does his Postdoctoral in Swinburne University of Technology. He works his research interest includes software engineering, data mining, machine learning, data privacy, and security. He can be contacted at email: mahmoud.hussein@ci.menofia.edu.eg.