Egy. J. Pure & Appl. Sci. 2022; 60(3):28-38

Egyptian Journal of Pure and Applied Science

A new algorithm for mining correct sequences of a specific behaviour for smart monitoring daily life activities

Amir Farouk [1], Fayed F.M. Ghaleb [1], Mohammad H. Abdel-Rahman[1], Wael Zakaria [1]

[1] *Department of Mathematics, Faculty of Science, Ain Shams University, Cairo, Egypt.*

ARTICLE INFO

Received 11 October 2022
Accepted 28 November 2022

Keywords

Internet of Things,
Frequent itemset,
Activities' sequences mining,
Daily life activities

Correspondence

Wael Zakaria

E-mail

wael.zakaria@sci.asu.edu.eg

ABSTRACT

In smart homes, mining frequent/correct activities' sequences, $\boldsymbol{AS}$, of specific behaviour, plays a vital role in building smart monitoring systems analyzing daily life activities (DLA), from which, the system can identify anomalies and automatically send alerts to users to remember them regarding any missing activity. Some researchers developed an intelligent system based on the Apriori algorithm, where all frequent k-Activities' sets mined by Apriori are used to identify all their permutations, which are then filtered out to extract just the frequent/correct k-Activities' sequences. However, because of using the Apriori algorithm, this system suffers from repeatedly scanning the DLA dataset and generating a huge number of candidates. As well as the exponential complexity of finding all permutations of all frequent k-activities' set to find the frequent k-activities' sequences$\boldsymbol{AS^k}$. In this paper, a new Positional Representation-based Frequent $\boldsymbol{AS^k}$ Mining algorithm, PR-FASM has been proposed, which is based on a new representation called Positional Representation (PR) of each activities' sequence of a specific behaviour. PR reflects the correct orders of each $\boldsymbol{AS^k}$ across all possible $\boldsymbol{AS}$ of a specific behaviour. PR-FASM overcomes the drawbacks of the mentioned system by scanning the DLA dataset only once and reducing the search space and time for finding the frequent $\boldsymbol{AS^k}$. On a CHESS dataset and a real smart home dataset called CASAS, the experimental results show that the system that is based on the PR-FASM algorithm is more efficient and scalable than the systems based on the Apriori algorithm and other sequence mining algorithms.

1. Introduction

The Internet of Things (IoT) is a new paradigm that makes it possible for electrical gadgets and sensors to communicate with one another through the internet to make our lives easier [1]. Simply, this technology could connect the living and non-living objects "Things" via the internet. In the object-oriented paradigm, it is known that everything in the world is considered as an object, but in the technology of the IoT paradigm, everything in the world is assumed as a smart object and it is able to communicate with one another via the internet technologies, either virtually or physically [2]. IoT technology is used in many applications such as smart cities, industry, healthcare, monitoring daily life activities (DLA), and providing some services.

One of the problems that attracts the researchers is monitoring and analyzing DLA to make life safe and easier. Therefore, this paper focuses on the problem of monitoring and analyzing DLA in an efficient way. Section 1.1 introduces the meaning of DLA. Section 1.2 formally presents the problem statement of DLA and the aim of this paper.

1.1 Daily Life Activities (DLA)

In smart homes, there are a lot of sensors attached to home objects such as doors, keys, mobiles, medicine, etc [3]. These sensors have readable tags that can be detected by radio-frequency identification (RFID) technology. Once the RFID detects or reads a sensor, it reflects that the user touches the corresponding object [4]. In another word, any touched object is considered a user's activity. Daily Life Activities (DLA) is one of the basics of smart homes, in which, the user's activities' sequences ($\boldsymbol{AS}$) have been collected daily while he is doing a specific behaviour such as going out home, sleeping, etc. Simply, the user's behaviour is a sequence of activities that happened together. For building a monitoring DLA system, a huge amount of data should be collected and analyzed. This system can mine all frequent activities' sequences ($\boldsymbol{FAS}$) from which, the system can easily detect any user's missing activity such as sleeping before having medicine, going out home before taking key, etc. Consequently, the monitoring system can automatically alert the user to remember him to do some forgotten activities.

Example 1. Fig. 1 is a simulation for DLA of an elderly woman, in which the objects: slipper, bed, burner and medicine are attached with readable sensors. In case, the woman touches any one of these objects, the attached sensor sends this activity to the monitoring system. The monitoring system is fully aware of the sequence of daily life activities of this woman and therefore able to detect any anomaly or unusual activity. Assume that the monitoring system is trained to know the usual behaviour of the woman such as sleeping behaviour. In particular, sleeping behaviour may contain several correct $\boldsymbol{AS}$ such as enters the bedroom, has medicine, sits on her bed, removes her slipper, then turns off the light or enters the bedroom, sits on her bed, removes her slipper, has medicine, then turns off the light. Therefore, if the woman turns off the light before having medicine, the system immediately alerts the woman to remember her to have her medicine firstly. On the other hand, if she sits on her bed before having medicine or have medicine before sitting on her bed the system doesn't detect any missing activity. Because both $\boldsymbol{AS}$ are correct.

Mining Association Rules (MAR) is one of the important data mining tasks which extract interesting correlations, frequent patterns, or associations among sets of items in the transaction databases or other data repositories [6, 7]. MAR plays an important role in analyzing DLA for extracting the frequent AS of a specific user's behaviour. In the following subsection, a problem statement will be formally introduced and how can MAR be used for mining $\boldsymbol{FAS}$ of a user.

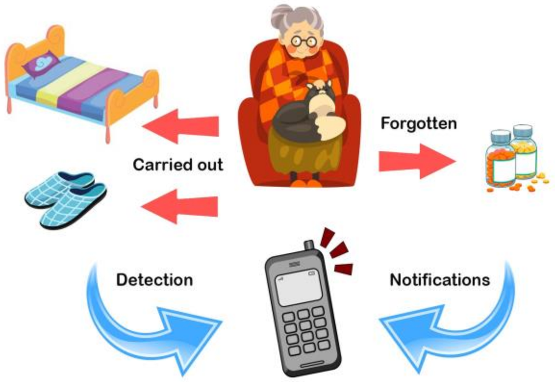


Fig. 1 An elderly woman is going to bed [5]**.**

1.2 Problem Statement

In this section, the problem of analyzing DLA is formally introduced to easily apply MAR for extracting frequent activities'sequences $\boldsymbol{FAS}$.

Consider A = $\{a_1, a_2, \dots, a_n\}$ is a set of n possible activities that the user performs in his daily life such as getting a key, opening a door, catching a mobile, etc. For simplicity, each activity is represented by the object's name that the user touched. For example, the object "ODoor" is used instead of the activity "opening the door". Therefore, a DLA dataset can be defined as in "Def. **1.1**"

Definition 1.1: (DLA Dataset *D*)

DLA dataset $D = \{d_1, d_2, \dots, d_m\}$ is set of daily life activities for *m* days; the activities on the j^{th} day, d_j, is a sequence of k pairs; $k > 0$. Each pair p=<*b, t*> can be described by the activity $b \in A$ and its happening time t.

Definition 1.2: (Behaviour Dataset $\boldsymbol{D_B}$)

Behaviour Dataset $D_{B,}$ of a specific behaviour *B,* is a set of all possible activities' sequence AS ⊆ d_j that take place near together. Formally D_B= {AS$_1$, AS$_2$, …, AS$_q$}; q>0. For example, for behaviour "go out home", B = {<mobile, key, door>, <mobile, glasses, key, door>, <key, mobile, door>, etc}. For instance, the first AS$_1$ means the user make the following activities in order: mobile, key, then door. It is worth noting that, one behaviour may have more than one possible $\boldsymbol{AS}$ in a single day.

As well as one day may contain several $\boldsymbol{AS}$ for many behaviours. Therefore $d_j \in D$ is split into several $\boldsymbol{AS}$. All $\boldsymbol{AS}$ regarding a specific behaviour B are combined in a behaviour dataset D_B.

Table 1. $D_{\text{go out home}}$ a dataset of all possible activities' sequences AS of a behaviour "go out home".

Activities' Sequences ID	Activities' Sequences (AS)
AS_1	<Clothes, Shoes, MobilePhone, Bag, Keys, ODoor, Elevator>
AS_2	<Shoes, Bag, Keys, MobilePhone, ODoor, Keys2, ODoor2, Elevator>
AS_3	<Tap, Towel, BathroomDoor, BedroomDoor, Clothes, Shoes, Bag, Keys, MobilePhone, ODoor, Keys2, ODoor2, Elevator>
AS_4	<Shoes, Keys, MobilePhone, Bag, ODoor, Elevator, Keys2, ODoor2>
AS_5	<Shoes, Bag, MobilePhone, Kyes, ODoor, Keys2, ODoor2, Elevator>

Example 2. Consider A = {mobile, shoes, key, door} and for instance $d_i \in D$ = < <mobile, 10:00 am>, < shoes, 10:01 am>, < key, 10:02 am>, < door, 10:10 am>, <key, 5:00 pm>, <door, 5:01 pm >, < shoes, 5:02 pm >>. As noted, there is a time gap between the happening activities door and key. In this case, the sequence d_i is split into two sub-sequences. The first one is < mobile, shoes, key, door > is assigned to $D_{\text{go out home}}$, which means that the person gets a mobile, puts shoes on, gets a key, then closes the door. While the second sequence is < key, door, shoes > is assigned to $D_{\text{return back home}}$, which means get the key, open the door, then take off shoes.

Table 1. shows the behaviour dataset D_B; which contains all possible $\boldsymbol{AS}$ of a behaviour B "go out home". As shown, AS_1 is one of the possible activities' sequences of a behaviour "go out home". It means that to go out home the user may firstly wear his clothes, wear his shoes, catch his mobile phone, carry on his bag, catch his keys, open a door, and finally open an elevator. Similarly, $AS_2, \dots, AS_5$ represent the other possible $\boldsymbol{AS}$ of "go out home".

In this paper, the two symbols {} and < > are used for describing a set of *k* activities (*k*-activities set) and a sequence of *k* activities (*k*-activities sequence (AS^k)) respectively. Moreover, the term activity-set is used instead of item-set as known in the mining association rules algorithms. For example, in Table 1, {Keys, MobilePhone} is considered as 2-activities' set that appears in $AS_i; i = 1,2, \dots ,5$, but the AS^2<kyes, MobilePhone> appears in {AS_2, AS_3, AS_4} only, while the AS^2 <MobilePhone, Keys> appears in {AS_1, AS_5}. The aim of this paper is to introduce a new efficient algorithm for mining all $\boldsymbol{FAS}$ of the behaviour dataset D_B for analyzing and monitoring DLA.

The remainder of this paper is organized as follows. Section 2 presents some closed related work to DLA. Section 3 introduces the proposed algorithm for mining $\boldsymbol{FAS}$ of a specific behaviour for DLA. Section 4 gives a comparative study between the Apriori-based sequences mining algorithm employed by the Delgado system [5] and the proposed PR-FASM algorithm. Finally, the paper is summarized in section 5.

2. Related Work

The related work can be categorized into the following two subsections:

2.1 Sequence mining algorithms

A lot of study has been done on mining frequent sequences of itemsets from sequences' dataset. The structure of the traditional sequence dataset, that is used by these algorithms, is slightly different from the DLA sequence dataset. Where the traditional dataset contains m customers instead of m days. Each customer contains a sequence of k itemsets b; b may be one or more items. On the other hand, b in the DLA dataset is only one activity/item. So, to employ these algorithms for building DLA monitoring systems, we restrict the itemset b to only one item/activity. Following is a brief discussion of a few of the well-known sequence mining algorithms.

The algorithms Spade [8], Spam [9], prefixspane, [10] CloFast [11], and Fast [12] have been proposed for fast finding frequent items' sequences. Some of them enhance the Apriori algorithm and others enhance FP-grows. However, these algorithms must be tuned to meet the characteristics of the DLA dataset. These sequence mining algorithms pruned any sequence with support less than minsup. In DLA, ignoring a sequence with less than minsup is not correct as will be discussed later. Therefore, as a result of these algorithms, a lot of interesting sequences are missed. For example, the itemset X={abc} with support 80% may be produce six sequences <abc>, <acb>, <bac>, <bca>, <cab>, <cba> with supports 20%, 20%, 0%, 0%, 10%, and 30% respectively.

Consider minsup=50%, although the itemset X is frequent, the traditional sequence mining algorithms pruned all the produced sequences. In real applications such as DLA, all the sequences produced from frequent itemsets X should be considered frequent without considering the sequence's support.

The only sequences which have been pruned are those of sequence's support equals zero. Therefore, in DLA, we must pertain to all sequences extracted from frequent itemsets even if their supports are less than minsup. This requires mining firstly the frequent itemsets, from which we can extract the frequent sequences. This restriction will make sequence mining require many computations compared with the traditional one.

2.2 Sequence mining algorithms-based DLA monitoring systems

Delgado et al (2009) [5] proposed a system that uses the Apriori algorithm [6] to identify $\boldsymbol{FAS}$ from the behaviour dataset D_B. In which, the Apriori algorithm [6] generates all frequent activities sets, from which all possible permutations are generated to extract only the correct activities' sequences. For instance, by applying the Apriori algorithm [6] with support = 0.9, over the database in Table 1, one of the Delgado outputs is the 3-activities' set {Keys, ODoor, MobilePhone}. From this activities' set all permutation is obtained as shown in Table 2. However, not all sequences that were obtained are frequent/correct. According to the order of activities in Table 1 the sequences AS_2 and AS_6 are frequent/correct, while the other sequences are infrequent/incorrect, therefore, they are deleted.

This system overcomes the problem of missing valuable sequences as in the mentioned algorithms (Section 2.1) by considering all sequences that are generated from frequent itemsets. However, this system suffers from multiple scanning of D_B and using large memory space and exponential computations for generating all permutations.

Table 2. All possible permutation of 3-activities' set { Keys, ODoor, MobilePhone }

Sequence ID	Sequences
AS_1	<Keys, ODoor,MobilePhone>
AS_2	<Keys, MobilePhone, ODoor>
AS_3	<ODoor, Keys, MobilePhone>
AS_4	<ODoor, MobilePhone, Keys>
AS_5	<MobilePhone, ODoor, Keys>
AS_6	<MobilePhone, Keys, ODoor>

Ros et al (2011) [13] overcomes the drawback of neglecting time in [5] by using fuzzy temporal windows. However, the algorithm still suffers from memory costs and high computations. Various learning approaches have been used in smart home applications in recent years. For instance, Fahad et al. (2015) [14] proposed a binary solution strategy based on a support vector machine (SVM) for classifying correct/incorrect assignments to simplify the problem. However, according to the nature of SVM, we cannot interpret why activates' sequence in DLA is incorrect. In other words, we cannot identify what is the missing activity to remember the user.

Yaqing et al (2020) [15] presents a new strategy for DLA recognition using feature selection based on the Pearson Correlation Coefficient. Three classifiers were used for activity recognition, namely, Naive Bayes (NB), C4.5, and Random Forest (RF). Although these models perform better when dealing with uncertain and incomplete data, but the drawback of this technique that optimization requires an extremely massive training dataset. Also, it has the same problem as in SVM-based approaches.

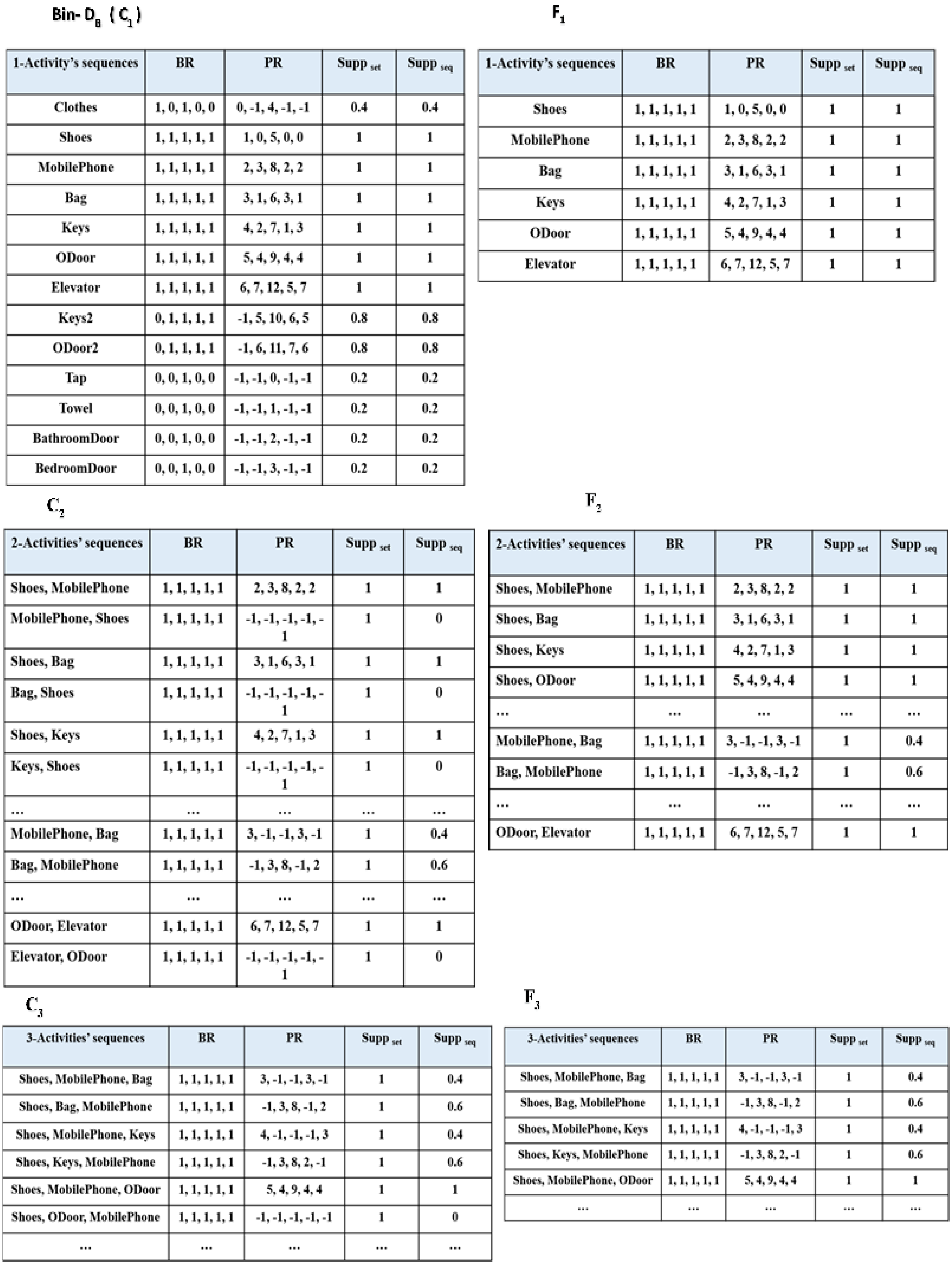
In this paper, a new algorithm called Positional Representation-based Frequent Activities' Sequences Mining algorithm, PR-FASM, is proposed for mining all activities sequences of a specific behaviour$B$. Compared to the Delgado algorithm, PR-FASM scans the dataset only once and reduces the complexity of finding $\boldsymbol{FAS}$. Moreover, compared to the SVM-based systems, PR-FASM algorithm is interpretable which makes the algorithm easy to detect incorrect activities' sequence and consequently detect any missing activity for warning the user.

3. The proposed PR-FASM algorithm

In this section, using the problem statement (Section 1.2), several definitions have been introduced to formally introduce the proposed PR-FASM algorithm. The behaviour dataset D_B is re-represented using column enumeration [16, 17] as shown in Fig. 2, in which each activity is represented as a binary representation.

Definition 3.1 [16,17] (k-activities' set binary representation) Every k-activities' set $A^k \subseteq A$ is represented by a binary vector BR_{A^k} of length $|D_B|$, where $BR_{A^k}(i) = 1$ if A^k happened at AS_i, otherwise $BR_{A^k}(i) = 0$; i=1,2, 3,..., $|D_B|$. For example, as shown in Table 1, the 1- activity's set {keys2} happened in AS_2, AS_3, AS_4, and AS_5, but not happened in AS_1. Therefore, $BR_{\{keys2\}}$ = [0,1,1,1,1]. Similarly, the 2-activities' set {Keys, MobilePhone} happened in AS_1, AS_2, AS_3, AS_4, and AS_5 (Table 1), therefore $BR_{\{Keys,\ MobilePhone\}}$ = [1,1,1,1,1]. It is worth noting that, a binary representation has been used for k-activities' set, not k- activities' sequence. Therefore, all that must be known are the AS_i that the k-activities' set occurred during.

Definition 3.2 (1-Activity's Sequence Positional Representation). Every 1-activity's sequence AS^1 is represented by a vector PR_{AS^1} of length $|D_B|$. $PR_{AS^1}(i) = p_i$; where p_i is the occurred position of AS^1 at AS_i. p_i is set to −1, if AS^1 does not happen at AS_i.



Bin- D_B (C_1)

1-Activity's sequences	BR	PR	Supp $_{set}$	Supp $_{seq}$
Clothes	1, 0, 1, 0, 0	0, -1, 4, -1, -1	0.4	0.4
Shoes	1, 1, 1, 1, 1	1, 0, 5, 0, 0	1	1
MobilePhone	1, 1, 1, 1, 1	2, 3, 8, 2, 2	1	1
Bag	1, 1, 1, 1, 1	3, 1, 6, 3, 1	1	1
Keys	1, 1, 1, 1, 1	4, 2, 7, 1, 3	1	1
ODoor	1, 1, 1, 1, 1	5, 4, 9, 4, 4	1	1
Elevator	1, 1, 1, 1, 1	6, 7, 12, 5, 7	1	1
Keys2	0, 1, 1, 1, 1	-1, 5, 10, 6, 5	0.8	0.8
ODoor2	0, 1, 1, 1, 1	-1, 6, 11, 7, 6	0.8	0.8
Tap	0, 0, 1, 0, 0	-1, -1, 0, -1, -1	0.2	0.2
Towel	0, 0, 1, 0, 0	-1, -1, 1, -1, -1	0.2	0.2
BathroomDoor	0, 0, 1, 0, 0	-1, -1, 2, -1, -1	0.2	0.2
BedroomDoor	0, 0, 1, 0, 0	-1, -1, 3, -1, -1	0.2	0.2

F_1

1-Activity's sequences	BR	PR	Supp $_{set}$	Supp $_{seq}$
Shoes	1, 1, 1, 1, 1	1, 0, 5, 0, 0	1	1
MobilePhone	1, 1, 1, 1, 1	2, 3, 8, 2, 2	1	1
Bag	1, 1, 1, 1, 1	3, 1, 6, 3, 1	1	1
Keys	1, 1, 1, 1, 1	4, 2, 7, 1, 3	1	1
ODoor	1, 1, 1, 1, 1	5, 4, 9, 4, 4	1	1
Elevator	1, 1, 1, 1, 1	6, 7, 12, 5, 7	1	1

C_2

2-Activities' sequences	BR	PR	Supp $_{set}$	Supp $_{seq}$
Shoes, MobilePhone	1, 1, 1, 1, 1	2, 3, 8, 2, 2	1	1
MobilePhone, Shoes	1, 1, 1, 1, 1	-1, -1, -1, -1, -1	1	0
Shoes, Bag	1, 1, 1, 1, 1	3, 1, 6, 3, 1	1	1
Bag, Shoes	1, 1, 1, 1, 1	-1, -1, -1, -1, -1	1	0
Shoes, Keys	1, 1, 1, 1, 1	4, 2, 7, 1, 3	1	1
Keys, Shoes	1, 1, 1, 1, 1	-1, -1, -1, -1, -1	1	0
…	…	…	…	…
MobilePhone, Bag	1, 1, 1, 1, 1	3, -1, -1, 3, -1	1	0.4
Bag, MobilePhone	1, 1, 1, 1, 1	-1, 3, 8, -1, 2	1	0.6
…	…	…	…	…
ODoor, Elevator	1, 1, 1, 1, 1	6, 7, 12, 5, 7	1	1
Elevator, ODoor	1, 1, 1, 1, 1	-1, -1, -1, -1, -1	1	0

F_2

2-Activities' sequences	BR	PR	Supp $_{set}$	Supp $_{seq}$
Shoes, MobilePhone	1, 1, 1, 1, 1	2, 3, 8, 2, 2	1	1
Shoes, Bag	1, 1, 1, 1, 1	3, 1, 6, 3, 1	1	1
Shoes, Keys	1, 1, 1, 1, 1	4, 2, 7, 1, 3	1	1
Shoes, ODoor	1, 1, 1, 1, 1	5, 4, 9, 4, 4	1	1
…	…	…	…	…
MobilePhone, Bag	1, 1, 1, 1, 1	3, -1, -1, 3, -1	1	0.4
Bag, MobilePhone	1, 1, 1, 1, 1	-1, 3, 8, -1, 2	1	0.6
…	…	…	…	…
ODoor, Elevator	1, 1, 1, 1, 1	6, 7, 12, 5, 7	1	1

C_3

3-Activities' sequences	BR	PR	Supp $_{set}$	Supp $_{seq}$
Shoes, MobilePhone, Bag	1, 1, 1, 1, 1	3, -1, -1, 3, -1	1	0.4
Shoes, Bag, MobilePhone	1, 1, 1, 1, 1	-1, 3, 8, -1, 2	1	0.6
Shoes, MobilePhone, Keys	1, 1, 1, 1, 1	4, -1, -1, -1, 3	1	0.4
Shoes, Keys, MobilePhone	1, 1, 1, 1, 1	-1, 3, 8, 2, -1	1	0.6
Shoes, MobilePhone, ODoor	1, 1, 1, 1, 1	5, 4, 9, 4, 4	1	1
Shoes, ODoor, MobilePhone	1, 1, 1, 1, 1	-1, -1, -1, -1, -1	1	0
…	…	…	…	…

F_3

3-Activities' sequences	BR	PR	Supp $_{set}$	Supp $_{seq}$
Shoes, MobilePhone, Bag	1, 1, 1, 1, 1	3, -1, -1, 3, -1	1	0.4
Shoes, Bag, MobilePhone	1, 1, 1, 1, 1	-1, 3, 8, -1, 2	1	0.6
Shoes, MobilePhone, Keys	1, 1, 1, 1, 1	4, -1, -1, -1, 3	1	0.4
Shoes, Keys, MobilePhone	1, 1, 1, 1, 1	-1, 3, 8, 2, -1	1	0.6
Shoes, MobilePhone, ODoor	1, 1, 1, 1, 1	5, 4, 9, 4, 4	1	1
…	…	…	…	…

Fig. 2 Running example of generation process of mining frequent k- activities' sequences mining $\boldsymbol{FASM}$ using PR-FASM. minsup=0.9.

Example 3. In Table 1, the 1- activity's sequence AS^1 = <keys> happened in positions 4th, 2nd, 7th, 1st, and 3rd at {AS$_1$, AS$_2$, AS$_3$, AS$_4$, AS$_5$} respectively, then the PR_{AS^1} = {4, 2, 7, 1, 3}. While the 1- activity's sequence AS^1 = <keys2> happened in positions 5th, 10th, 7th, and 5th at AS$_2$, AS$_3$, AS$_4$, and AS$_5$ respectively and not happened in AS$_1$. Therefore, PR_{AS^1}= {−1, 5, 10, 7, 5} (as shown in the first table in Fig. 2). Therefore, the behaviour dataset D$_B$ is compressed into binary-based behaviour dataset Bin − D$_B$ as shown in Fig. 2.

In this paper, the following definitions of the used data structure have been introduced for efficiently storing the information of the behaviour dataset Bin − D$_B$ which is the backbone of the proposed algorithm for mining frequent activities' sequence in an efficient manner.

Definition 3.3 (k-activities' sequences Positional Representation). Every k-activities' sequences AS^k; k ≥ 2 is represented by a vector PR_{AS^k} of length |D$_B$|. $PR_{AS^k}$is recursively calculated using the PR of the first k − 1 activities (AS^{k-1}) and PR of the kth activity (AS^1) of AS^k; Where PR_{AS^k} (i) is calculated according to the following equation:

$$PR_{AS^k}(i) = \begin{cases} PR_{AS^1}(i) & if\ PR_{AS^1}(i) \neq -1\ and \\ & PR_{AS^1}(i) > PR_{AS^{k-1}}(i) \\ -1 & otherwise \end{cases}$$

Example 3.4. In Fig. 2, AS^3 = <Shoes, Bag, MobilePhone>, from which AS^2 = < Shoes, Bag> and its PR_{AS^2}= {3, 1, 6, 3, 1} and AS^1 = <MobilePhone> and its PR_{AS^1}= {2, 3, 8, 2, 2}, therefore $PR_{AS^3}$is calculated as follows:

- PR_{AS^3} (1) = −1; PR_{AS^1} (1) = 2 < PR_{AS^2} (1) = 3
- PR_{AS^3} (2) = 3; PR_{AS^1} (2) = 3 > PR_{AS^2} (2) = 1
- PR_{AS^3} (3) = 8; PR_{AS^1} (3) = 8 > PR_{AS^2} (3) = 6
- PR_{AS^3} (4) = −1; PR_{AS^1} (4) = 2 < PR_{AS^2} (4) = 3
- PR_{AS^3} (5) = 2; PR_{AS^1} (5) = 2 > PR_{AS^2} (5) = 1

Therefore, PR_{AS^3}= {−1, 3, 8, −1, 2}, which means that the sequence < Shoes, Bag, MobilePhone > is correct in AS$_2$, AS$_3$, and AS$_5$. because the MobilePhone comes after <Shoes, Bag> in AS$_2$, AS$_3$, and AS$_5$. While PR of AS^3 <Shoes, MobilePhone, Bag > is {3, −1, −1, 3, −1}.

Definition 3.4 [16, 17] (Support of k- activities' set). The support of a k-activities' set A^k ⊆ A, supp − set$_{A^k}$, is its relative occurrence in Bin − D$_B$ that is determined by:

$$supp - set_{A^k} = \frac{\#\ ones\ in\ BR_{A^k}}{|D_B|} \qquad (1)$$

Based on example 4, supp − set $_{A^3}$ = supp-set$_{(\{Shoes, Bag, MobilePhone\})}$ = (5/5) = 1 as shown in Fig. 2.

Definition 3.5 (Support of k-activities' sequences). The support of a k-activities' sequences AS^k , supp − seq $_{AS^k}$, is the frequency of its occurrence in Bin − D$_B$ that is determined by:

$$\text{supp} - \text{seq}_{AS^k} = \frac{\#\ non\text{−}negative\ values\ in\ _{AS^k}}{|D_B|} \qquad (2)$$

Based on example 4, supp − seq $_{AS^3}$ = supp-seq$_{(<Shoes,Bag,MobilePhone>)}$ = (3/5) = 0.6 as shown in Fig. 2. While, supp − seq $_{AS^3}$ = supp-seq$_{(<Shoes,MobilePhone,Bag>)}$=(2/5)= 0.4 as shown in Fig. 2.

Definition 3.6 (Frequent k-activities' sequence). The k-activities' sequence AS^k , is called frequent if supp − set_{AS^k} ≥ minsup and supp − seq$_{AS^k}$ ≠ 0. Therefore, frequent k-activities' sequences FAS$_k$={ AS^k; AS^k $is\ frequent$ }. For simplicity, the terms FAS$_k$ and F$_k$ are similar. Example AS^3 =< Shoes, Bag, MobilePhone > is frequent sequences' activity because supp − set$_{AS^3}$ = 1 > $minsup$ and supp − seq$_{AS^3}$ = 0.6 ≠ 0 . While AS^3 = < Shoes, ODoor, MobilePhone > is infrequent because supp − set$_{AS^3}$ = 1 and supp − seq$_{AS^3}$ = 0.

Using the mentioned definitions, the proposed PR-FASM algorithm (Algorithm 1) is applied on the compressed dataset Bin − D$_B$ (Fig. 2) instead of the original dataset (Table 1) for mining all ***FAS*** (Lines 1,2). Recalling that, the dataset focuses on the behaviour "go out home". Therefore, any mined frequent AS^k tells us about the correct ***AS*** for behaviour "go out home".

The mined correct ***AS*** are used for remembering the user if any activity is missed. Algorithm 1 works as follows: Consider the dataset Bin − D$_B$ is C$_1$ (Fig. 2) that contains all the candidates of 1- activity's sequences (C$_1$) (Line 3), from which the frequent AS^1 (F$_1$) is extracted according to the conditions in Line 4. Iteratively (Lines 5-13), for k ≥ 2, the set C$_k$ is extracted by joining the F$_{k-1}$ by itself and when k>2, F$_2$ is used only for checking if the extracted sequence is valid (correct order), from which the F$_k$ are extracted according to (**Definition 3.6**). The iteration is stopped when F$_{k-1}$ = φ. Using the positional representation, the joining process between two sequences is performed according to (**Definition 3.7**).

Definition 3.7. (Joining process of two sequences). Consider the two sequences of length k-1, $\mathrm{S_i^{k-1}} = \langle \mathrm{a_1, a_2, \ldots, a_{k-2}, a_{k-1}} \rangle$ and $\mathrm{S_j^{k-1}} = \langle \mathrm{a_1, a_2, \ldots, a_{k-2}, b_{k-1}} \rangle$ which are the same of all the first k−2 activities and differ only in the (k −1)th activity.

The joining between $\mathrm{S}_\mathrm{i}^{\mathrm{k}-1}$ and $\mathrm{S}_\mathrm{j}^{\mathrm{k}-1}$ may form the following new sequences:

1. $\mathrm{S}_\mathrm{p}^\mathrm{k} = \langle \mathrm{a}_1, \mathrm{a}_2, \dots, \mathrm{a}_{\mathrm{k}-2}, \mathrm{a}_{\mathrm{k}-1}, \mathrm{b}_{\mathrm{k}-1} \rangle$ if the sequence $\langle \mathrm{a}_{\mathrm{k}-1}, \mathrm{b}_{\mathrm{k}-1} \rangle \in F_2$.
2. $\mathrm{S}_\mathrm{q}^\mathrm{k} = \langle \mathrm{a}_1, \mathrm{a}_2, \dots, \mathrm{a}_{\mathrm{k}-2}, \mathrm{b}_{\mathrm{k}-1}, \mathrm{a}_{\mathrm{k}-1} \rangle$ if the sequence $\langle \mathrm{b}_{\mathrm{k}-1}, \mathrm{a}_{\mathrm{k}-1} \rangle \in F_2$.
3. NULL, otherwise.

Therefore, as shown in Algorithm 1, the joining function at k ≥ 3 requires the F_2 as an input (Line 9) for forming valid sequences. While at k=2 (Line 7), the F_2 does not formed yet, therefor, the null is passed instead. Based on (**Definition 3.7**), Algorithm 2 shows how the joining process is applied between two sequences to form only frequent/correct sequences.

Algorithm 1 Mining Frequent k-activities' sequences

```
1: Input: Bin – D_B, minsup.
2:Output: FAS is a set of frequent k-activities' sequences
3: C_1 ← Bin – D_B, FAS = {}, k=1
4: F_1 = {c ∈ C_1; supp-set_c ≥ minsup}
5: do
6:    k=k+1
7:    if k == 2 then
8:        F_k ← Join(F_k-1, null, minsup) // F_1 joins itself
9:    else
10:       F_k ← Join(F_k-1, F_2, minsup) //Given F_2,
                                         // F_k-1 joins itself
11:   end if
12:   add F_k to FAS
13: while F_k ≠ ∅
```

Example 3.5. Consider the dataset Bin – D_B as shown in (Fig. 2) and minsupp = 0.9, Algorithm 1 considers Bin – D_B as the set of 1-activity's candidates C_1 (Fig. 2), from which, 1- activity's sequences F_1 has been extracted. The next step is to form the set of 2- activities' candidates C_2 by joining F_1 by itself from which the frequent 2-***AS*** F_2 has been extracted. For instance, consider, $\mathrm{S}_1^1 = \langle Shoes \rangle$, $\mathrm{S}_2^1 = \langle MobilePhone \rangle$; $\mathrm{S}_1^1, \mathrm{S}_2^1 \in F_1$ $BR_{S_1^1} = [1,1,1,1,1]$ and $PR_{S_1^1} = [1,0,5,0,0]$, $BR_{S_2^1} = [1,1,1,1,1]$ and $PR_{S_2^1} = [2,3,8,2,2]$ as shown in (Fig. 2).

According to (Algorithm 2), $\mathrm{S}_1^1, \mathrm{S}_2^1$ are joined to generate two sequences $\mathrm{S}_\mathrm{p}^2 = \langle Shoes, MobilePhone \rangle$, $\mathrm{S}_\mathrm{q}^2 = \langle MobilePhone,\ Shoes \rangle$ with orders $PR_{S_p^2} = [2,3,8,2,2]$, $PR_{S_q^2} = [-1,-1,-1,-1,-1]$ and $BR_{S_p^2} = BR_{S_q^2} = [1,1,1,1,1]$ Since supp-set$_{\text{shoes,MobilePhone}}$ = 1 which is greater than minsupp = 0.9, then 2- activities set {Shoes, MobilePhone} is frequent. Since $supp - seq_{S_p^2} = 1$ and $supp - seq_{S_q^2} = 0$, then S_p^2 is only added to F_2 (Fig. 2).

Algorithm 2 F_k ← Join(F_{k-1}, F_2, minsup)

```
for ($i = 0; i < F_{k-1}.size - 1;\ i = i + 1$) do
    $S_i^{k-1} \leftarrow F_{k-1}(i)$ // $S_i^{k-1}$ is the i^th frequent
                                   //sequence of k-1 activities.
    for ($j = i + 1;\ j < F_{k-1}.size;\ j = j + 1$) do
        $S_j^{k-1} \leftarrow F_{k-1}(j)$ // $S_j^{k-1}$ is the j^th
                              // frequent sequence of k-1 activities.
        if $S_i^{k-1}$ and $S_j^{k-1}$ have the same
            first k-2 actions (Def. 3.7) then
            $S_p^k \leftarrow< S_i^{k-1}(1), \dots, S_i^{k-1}(k-2), S_i^{k-1}(k-1),$
                                            $S_j^{k-1}(k-1) >$
            $S_q^k \leftarrow< S_i^{k-1}(1), \dots, S_i^{k-1}(k-2), S_j^{k-1}(k-1),$
                                            $S_i^{k-1}(k-1) >$
            $BR_{S_p^k} = BR_{S_i^{k-1}} \cap BR_{S_j^{k-1}}$
            $BR_{S_q^k} = BR_{S_p^k}$
            $PR_{S_q^k}, PR_{S_p^k}$ //are calculated according
                                   //to (Def. 3.3).
            if $supp - set_{S^k} \geq minsup$ then
                add $S_q^k, S_p^k$ to F_k if $supp - seq_{S_q^k} \neq 0$
                and $supp - seq_{S_p^k} \neq 0$ respectively.
            end if
        end if
    end for
end for
```

- ***Complexity Analysis***

The complexity and the memory consumption of mining all frequently occurring itemsets or sequences is O (2^n), where n is the total number of items/activities [6].

Numerous studies have been conducted to reduce this cost, but none of them accomplish so by reducing complexity. As a result, the goal is to shorten key essential phases to lower the mining process' overall processing time. The below Delgado algorithm challenges are ones we try to overcome:

1. It utilizes the row-enumeration-based Apriori method, which results in multiple scanning of the dataset.
2. Generate every common itemset X, from which every permutation is produced to obtain every possible sequence of X. A huge number of candidates are produced by this stage. Additionally, it rescans the dataset to determine its correctness.

To prevent repeatedly scanning the dataset, we first employ column enumeration rather than row enumeration. Secondly, the proposed positional representation (PR) of sequences is used for including all the information required for further processes. To prevent the massive production of candidates, we use the Apriori algorithm property, which states that "if itemset x is frequent, then its superset is often". It is possible to apply this property to sequences, which indicates that if sequence x is infrequent, its superset is also infrequent. Delgado do not consider this property for mining all correct sequences, while the proposed system considers this property and has the potential to eliminate numerous possibilities in early phases. The suggested representation, known as positional representation (PR), is crucial in this regard.

4. Results and Discussion

In this section, we compare the proposed PR-FASM algorithm with two kinds of algorithms; first, the algorithm employed by DLA systems and second the sequence mining algorithms.

4.1 Sequence mining algorithm-based DLA monitoring systems

Delgado system [5] relies on the Apriori algorithm [6] for finding all FAS. The algorithm used for mining FAS is named Delgado algorithm. As known, The Apriori algorithm is based on row enumeration which suffers from multiple scanning of a dataset. It is worth noting that, there are many improvements have been made on Apriori algorithm [6] to overcome its drawbacks such as the series of column-based enumeration algorithms Eclat [16] and Quick-Apriori [17], etc.

Unfortunately, Delgado did not consider these improvements. In this paper, we implement two versions of Delgado algorithm, the first one is based on row enumeration and the other is based on column enumeration.

Therefore, a comparative study has been conducted between three algorithms: Delgado, Modified Delgado, and the proposed novel PR-FASM algorithm. All these algorithms are applied on two datasets. The first one is Delgado dataset [5]. The second one has been gathered by CASAS smart home technology [18]. Table 3 shows the characteristics of the datasets. In the CASAS dataset, some pre-processing is made to be suitable to the problem statement. The experiments are applied on PC of the following specification: operating system windows 10, Core (TM)i5-2450M CPU @ 2.50 GHz, and 8 GB ram memory. The algorithms have been implemented using Java language.

Table 3. The characteristics of the datasets.

Dataset	Sequences count	Items count
Delgado	5	13
CASAS	24	11
CHESS	3196	75

Using Delgado dataset [5], Fig. 3 Execution time in seconds between the three algorithms applied on Delgado Dataset [5]. shows the running time of the mentioned three algorithms in seconds for extracting all FAS. Using minsup = 0.2, 0.3, 0.9 the modified Delgado algorithm outperforms Delgado algorithm which means that using columns enumeration-based algorithms is better than row enumeration-based algorithm. While, the proposed PR-FASM algorithm is efficient and scalable than Delgado and modified Delgado. Moreover, at minsup = 0.2 the proposed algorithm success to achieve the goal in around one second, while the Delgado and modified Delgado did not respond and go into heap memory exception.

Using CASAS dataset [18], Fig. 4 Execution time in seconds between the three algorithms applied on CASAS Dataset [18]. shows the running time of the mentioned three algorithms in seconds for extracting all FAS. Using minsup = 0.2, 0.3, ..., 0.9 the difference in performance between the PR-FASM and the other two algorithms is very noticeable.

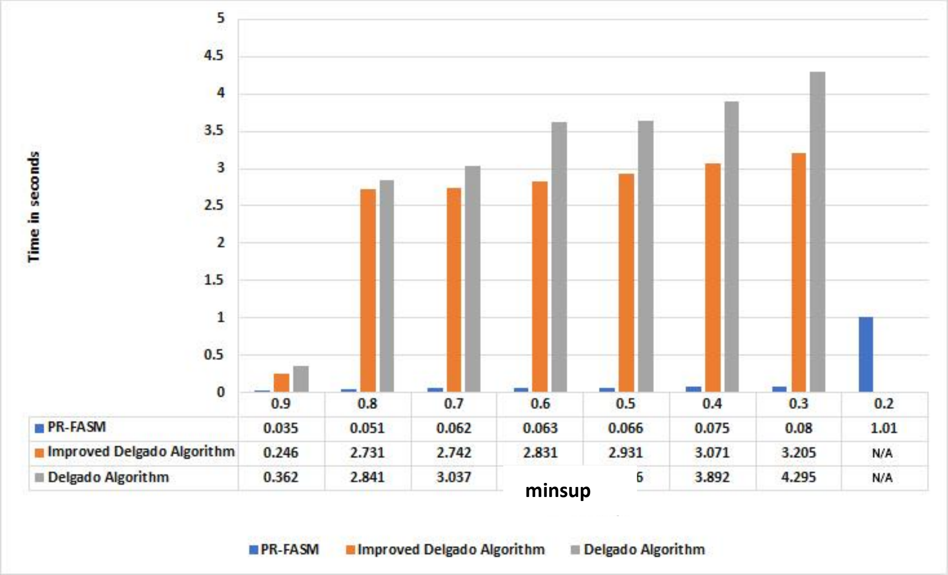


Fig. 3 Execution time in seconds between the three algorithms applied on Delgado Dataset [5].

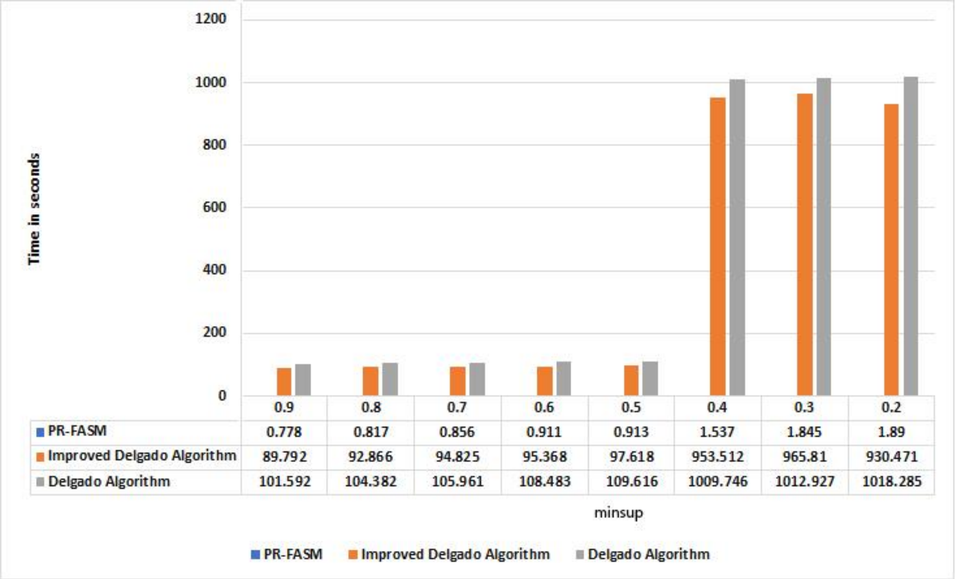


Fig. 4 Execution time in seconds between the three algorithms applied on CASAS Dataset [18].

4.2 Sequence mining algorithms

In this section, using the two datasets Chess[19] and CASAS[12] (Table 3), a comparative study has been conducted between the proposed PR-FASM and two well-known sequence mining algorithms CloFast [11] and Fast [12]. Using chess dataset Fig. 5. shows the running time of the mentioned three algorithms in seconds for extracting all ***FAS***. Using minsup = 0.6, 0.7, ..., 0.9 the PR-FASM algorithm outperforms both CloFast and Fast algorithms. As noted in the figure at minsup = 0.6 CloFast and Fast algorithms fails to find ***FAS*** according to heap memory exception.

Using CASAS dataset Fig. 6. shows the running time of the mentioned three algorithms in seconds for extracting all ***FAS***.Using minsup = 0.2, 0.3, ..., 0.9 the PR-FASM algorithm outperforms both CloFast and Fast algorithms. Although CloFast and Fast algorithms do not get all ***FAS*** as mentioned in section 2.1, the proposed PR-FASM, that finds all ***FAS*** is efficient and scalable. Finally, using the Positional Relation PR plays an important role for efficiently extract all FAS for monitoring DLA. Table 4. shows the memory consumption differences between the proposed PR-FASM algorithm and the other algorithms CloFast and Fast algorithms applied on CHESS dataset.

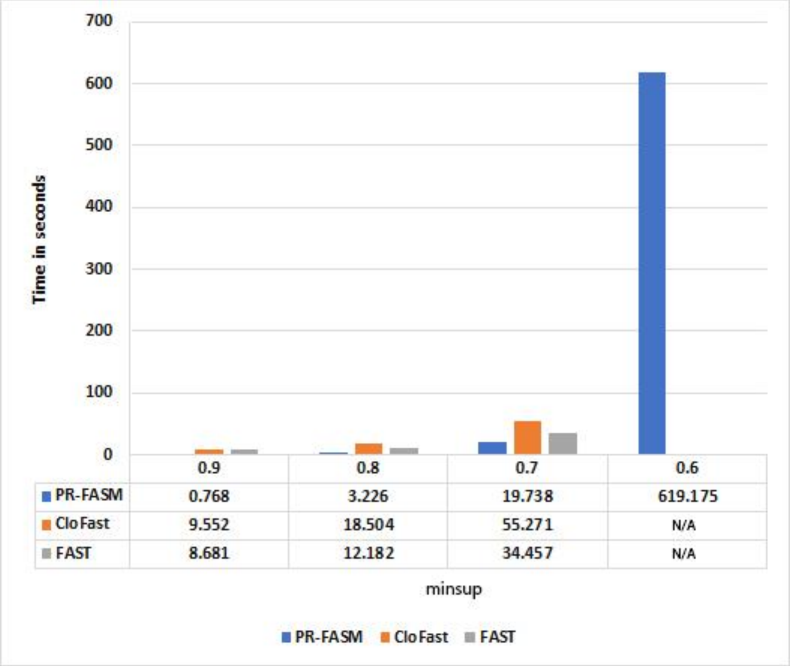


Fig. 5 Execution time in seconds between PR-FASM , CloFast [11] and Fast [12] algorithms based on CHESS dataset [19].

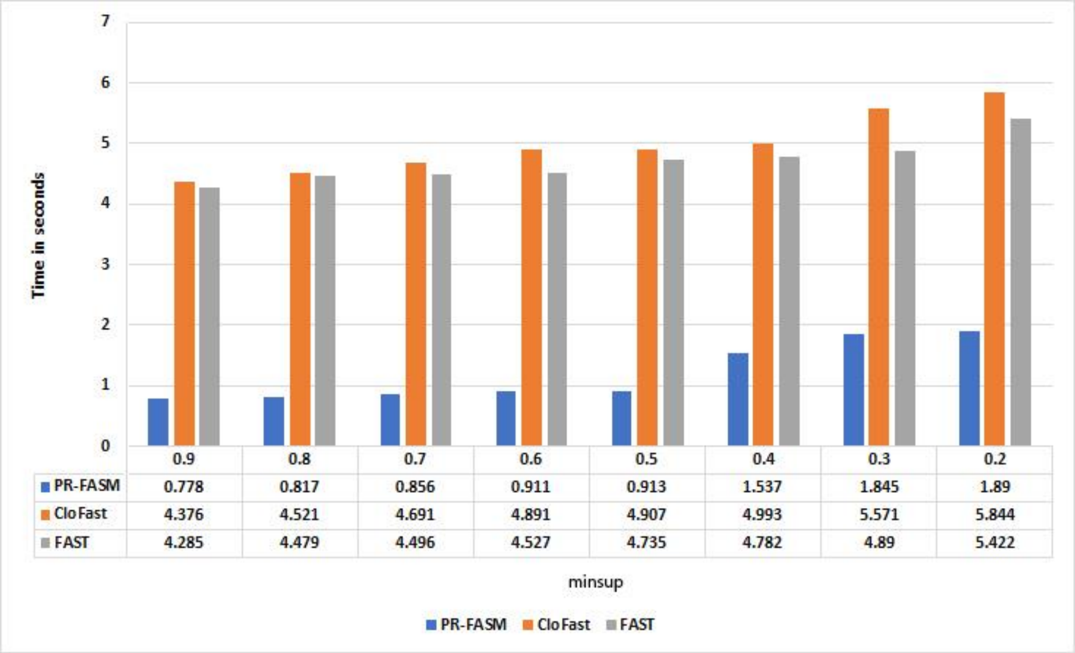


Fig. 6 Execution time in seconds between PR-FASM , CloFast [11] and Fast [12] algorithms based on CASAS dataset [18].

Table 4. Memory consumption between PR-FASM, CloFast [11] and Fast [12] algorithms applied on CHESS dataset [19].

Algorithm	Memory consumption at different minsup in MB			
	minsup = 0.9	minsup = 0.8	minsup = 0.7	minsup = 0.6
PR-FASM	19	189	874	4160
CloFast	388	499	999	heap memory exception.
Fast	118	413	990	heap memory exception.

5. Conclusion

For smart monitoring DLA, a positional representation-based frequent activities' sequences mining algorithm, PR-FASM, has been proposed. The algorithm used column-enumeration representation of the DLA dataset instead of row-enumeration as in Apriori. PR-FASM algorithm scanned the DLA dataset only once instead of multiple scanning as in Apriori. The proposed algorithm used a new representation method called positional representation PR. PR plays a crucial task for efficiently joining two activities' sequences of length k ($\boldsymbol{AS^k}$) to get new activities' sequence of length $k+1$ ($\boldsymbol{AS^{k+1}}$) that make the proposed algorithm scalable and efficient. Moreover, the proposed algorithm is interpretable and easy to detect any missing activity. As shown in the experimental results, Delgado algorithm, CloFast, and Fast failed in many cases to get the desired output or take a large time to achieve the goal, while the proposed algorithm PR-FASM achieved the goal in a few seconds. In addition, compared to the proposed PR-FASM algorithm, CloFast and Fast algorithms discard many valuable activities' sequences.

6. Acknowledgement

The authors acknowledge the role of the Artificial Intelligence and Cognitive Science Lab at the Department of Mathematics, Faculty of Science, Ain Shams University, Cairo, Egypt.

7. References

1. **Kumar, S., Tiwari, P., & Zymbler, M. (2019).** Internet of Things is a revolutionary approach for future technology enhancement: a review. Journal of Big data, **6(1)**, **1-21**.
2. **Dian, F. J., Vahidnia, R., & Rahmati, A. (2020).** Wearables and the Internet of Things (IoT), applications, opportunities, and challenges: A Survey. IEEE Access, **8**, **69200-69211.**
3. **Qolomany, B., Al-Fuqaha, A., Gupta, A., Benhaddou, D., Alwajidi, S., Qadir, J., & Fong, A. C. (2019).** Leveraging machine learning and big data for smart buildings: A comprehensive survey. IEEE Access, **7**, **90316-90356**.
4. **Mezzanotte, P., Palazzi, V., Alimenti, F., & Roselli, L. (2021).** Innovative RFID sensors for Internet of Things applications. IEEE Journal of Microwaves, **1(1)**, **55-65**.
5. **Delgado, M., Ros, M., & Vila, M. A. (2009).** Correct behavior identification system in a tagged world. Expert systems with applications, **36(6)**, **9899-9906**.
6. **Agrawal, R., Faloutsos, C., & Swami, A. (1993, October).** Efficient similarity search in sequence databases. In International conference on foundations of data organization and algorithms (**pp. 69-84**). Springer, Berlin, Heidelberg.
7. **Pujari, A. K. (2001).** Data mining techniques. Universities press.
8. **Zaki, M. J. (2001).** SPADE: An efficient algorithm for mining frequent sequences. Machine learning, **42(1)**, **31-60**.
9. **Ayres, J., Flannick, J., Gehrke, J., & Yiu, T. (2002, July).** Sequential pattern mining using a bitmap representation. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (**pp. 429-435**).
10. **Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., & Hsu, M. (2001, April).** Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In proceedings of the 17th international conference on data engineering (**pp. 215-224**). IEEE.
11. **Fumarola, F., Lanotte, P. F., Ceci, M., & Malerba, D. (2016).** CloFAST: closed sequential pattern mining using sparse and vertical id-lists. Knowledge and Information Systems, **48(2)**, **429-463**.
12. **Salvemini, E., Fumarola, F., Malerba, D., & Han, J. (2011, June)**. Fast sequence mining based on sparse id-lists. In International Symposium on Methodologies for Intelligent Systems (**pp. 316-325**). Springer, Berlin, Heidelberg.
13. **Ros, M., Delgado, M., & Vila, A. (2011).** Fuzzy method to disclose behaviour patterns in a tagged world. Expert Systems with Applications, **38(4)**, **3600-3612**.
14. **Fahad, L. G., Khan, A., & Rajarajan, M. (2015).** Activity recognition in smart homes with self-verification of assignments. Neurocomputing, **149**, **1286-1298.**
15. **Liu, Y., Mu, Y., Chen, K., Li, Y., & Guo, J. (2020).** Daily activity feature selection in smart homes based on pearson correlation coefficient. Neural Processing Letters, **51(2)**, **1771-1787**.
16. **Zaki, M. J. (2000).** Scalable algorithms for association mining. IEEE transactions on knowledge and data engineering, **12(3)**, **372-390**.
17. **Zakaria, W., Kotb, Y., & Ghaleb, F. (2014).** MCR-Miner: Maximal confident association rules miner algorithm for up/down-expressed genes. Applied Mathematics & Information Sciences, **8(2)**, **799**.
18. **"Casas shared smart home datasets repository". (2011).** http://casas.wsu.edu/datasets/.
19. **"SPMF An Open-Source Data Mining Library",** https://www.philippe-fournier-viger.com/spmf/