Soft Computing
https://doi.org/10.1007/s00500-023-08572-4

FOCUS



# PredictionMiner: mining the latest individual behavioral rules for personalized contextual pattern predictions

**Muhammad Faizan Khan[1] · Lu Lu[1] · Humaira Afzal[2] · Jamal Abdul Nasir[3]**

Accepted: 18 May 2023




## Abstract

The user's behavior towards smartphone is not static and changes dynamically as the real world changes, depending on various contextual patterns. Finding the latest behavioral rules is challenging because the smartphone log is incremented with the user's behavior, which is not static and changes daily. Previously, for mining, the latest user behavioral rules researchers have used the recent log of static periods and considered it as the latest behavioral log, however those approaches create accuracy and reliability problems because with time behavioral log keeps on updating and some user behaviors become outdated. On the basis of user's volatile behaviors toward smartphones, this study devises the issue of modeling an individual's up-to-date behavioral rules with their smart-phone interaction co-occurring patterns by incorporating the dynamically changing log data. Proposed behavioral-based approach named "PredictionMiner" firstly, mines the dynamic log period which holds the latest behavior of individual users by neglecting the outdated behaviors. Secondly, it extracts the individual latest smartphone machine learning rules with co-occurring contextual patterns. By utilizing individualized co-occurring patterns with the corresponding behavioral rules, the personalized context-aware prediction model is built for predicting future smartphone contextual behavioral activities. The proposed approach dynamically mines the latest machine learning rules and removes the outdated rules making it more effective. To make this approach more relevant, real-world contextual notifications dataset has been used. Our experiments and comparisons on such contextual dataset show that proposed rule discovery approach is more adequate and accurate than base models.

**Keywords** Machine learning · Context-aware prediction · Behavioral modeling · Co-occurring patterns mining · Association rule mining

## 1 Introduction

Recent smartphone advancements have enabled us to collect users' various behavioral activities on smartphones along with their related contextual information. We can save all functions, such as mobile application usage, phone call activities, notifications usage, social networking, and user location, in the form of the corresponding log file. Using public APIs on most smartphone platforms (Lee et al. 2022; Belkhir et al. 2019), we can easily collect users' contextual data, which includes online activity, SMS and call behaviors, battery usage, and charging behavior. With the help of these fine grained APIs, smartphone safety features have also increased, making them less prone to different kinds of attacks (Jiang et al. 2020).

The advancements in smartphones and their capability to store such varied user activities (Sarker et al. 2018) with related contexts enable the study of information-driven smartphone usage behavior modeling and prediction. To design a productive, context-specific user behavioral prediction model, it is necessary to analyze an individual's diverse behaviors with their co-occurring patterns in multidimen-

✉ Lu Lu
lul@scut.edu.cn

Muhammad Faizan Khan
aukhan@mail.scut.edu.cn

Humaira Afzal
humairaafzal@bzu.edu.pk

Jamal Abdul Nasir
jamalnasir@gu.edu.pk

1 School of Computer Science and Engineering, South China University of Technology, University Town, Guangzhou, People's Republic of China

2 Institute of Computing, BZU, Multan, Pakistan

3 ICIT, Gomal University, DI Khan, Pakistan

sional contexts. But, user behavior is not idle with time, it can change depending on various factors, such as the user's location. The change in location can alter the context and affect the way in which the user interacts with their smartphone. Previously statistical correlation functions were used that demonstrate compelling correlations among different numerical contexts such as sleep caliber and activity level (Hao et al. 2013). The author has used static correlation factors, but the contexts and behavioral patterns do not remain static; the user's smartphone behaviors and contextual patterns change over time and differ from user to user.

In this article, we focus on the symmetric yet equally crucial problem of mining co-occurring patterns with a personalized set of contemporary latest behavioral rules. This modern approach utilizes an individual's smartphone context-based notifications data to structure a contextual customized notifications interactive-based prediction model. Using notifications data can enhance the effectiveness of our approach, because the device's notification log, is capable of holding the individual smartphone interaction behavioral activities in contrasting contexts.

### 1.1 Motivation

To create a user behavioral prediction model that is effective and context-specific, it is important to analyze the individual's various behaviors and patterns that occur together in different situations. The predictor should be based on the user's most recent smartphone behaviors. However, user behavior is not static over time. As time goes on, user behavioral rules become outdated and expire. This is because users' behaviors towards smartphones change as their daily routines change, rendering previous prediction rules obsolete which make prediction quiet inaccurate. It is imminent to find the latest behaviors of user's for accurate predictions. Researchers have tried to incorporate this problem and used the latest log of user's data and consider it as recent behavior of users. That approach works fine for some time but as time goes by new data is added inclemently which makes the old rules unrelated and prediction becomes wrong. Some researchers have also tried to use a whole period of static data for modeling smartphone user behavioral change but those approaches creates a large amount of redundant and unnecessary rules which makes the extraction process inefficient and incorrect. Therefore, to make accurate predictions, the predictor should work on the dynamic selection of latest period of data with the capacity of generating compelling up to date behavioral rules on the basis of user's behavioral change. To solve those problems in the proposed approach the dynamic latest set of dynamic behavior-based rules are extracted with their corresponding co-occurring patterns for different users.

## 2 Related work and problem statement

To develop a rule-based behavioral prediction model (Chen and Li 2021), current researchers only use a fixed period of time. In our previous work (Khan et al. 2022), we tried to predict the ideal occasions at which notifications are not interruptable for users. Many other authors have also raised the issue that interruptions during ongoing tasks have unfavorable impacts on user performance, such as reducing task performance and affecting individuals' emotional state (Adamczyk and Bailey 2004; Boehm-Davis and Remington 2009). On the other hand, some interruptions are acceptable; many studies show that different circumstances, such as the type of major task, level of task engagement, and the timeframe of interruptions, might influence the descriptiveness of notifications (Mark et al. 2008). However, these studies lose credibility because user behaviors change dynamically, making it crucial to identify the ideal occasions when notifications are not interruptible. In Walsh et al. (2022), the author explored smartphone notification content to place succeeding Call-to-Action associated with them. The association rule mining technique has been widely used to tackle these kinds of problems, but the redundancy in the rules makes it inefficient. In our study, we used the customized association rule learning approach, where the redundancy of rules is removed, and only high-quality rules are produced. We still use association rule mining because recent studies suggest that association and classification rule learning techniques are widely used for finding rules from given datasets in the field of machine learning and data mining (Han and Kamber 2011; Azuaje 2006). In Freitas (2000), it was said that decision trees could not ensure that the discovered classification rules have high predictive efficiency and accuracy. The most state-of-the-art algorithm for mining association rules was proposed by Agrawal and Srikant (1994). In extension to the proposed model, many other methods were suggested to mine association rules dynamically from databases, such as tree-based (Visuri et al. 2019), Probability-based (Amornchewin and Kreesuradej 2009; Thusaranon and Kreesuradej 2015), pattern-based (Ünvan 2021; Zhiang et al. 2020), and three-way decision-based (Zhang et al. 2014; Li et al. 2017). These different techniques were developed to solve problems like processing speed, for example, by making the mining procedures efficient by minimizing the number of scans rather than scanning the entire dataset, which contains the original log with the additional chunk of the dataset. However, these techniques did not take into account the relevance and newness of the rules, i.e., producing an entire set of upgraded rules depends on the user's current and latest behaviors employing their contextual smartphone notification log and neglecting the rules that are obsolete and do not represent the user's behaviors.

### 2.1 Static period dataset extraction problems

To model user behaviors from the static period of contextual smartphone logs, several authors have contributed by using different kinds of log files, such as smartphone application records (Srinivasan et al. 2014; Liao et al. 2013), notifications (Walsh et al. 2022; Ibrahim et al. 2022), call histories (Sarker and Kayes 2020), web browsing activity (Bordino et al. 2012), game usage data (Zhang et al. 2014), sensor data (Rawassizadeh et al. 2013; Mafrur et al. 2015) and smartphone contextual logs (Zhu et al. 2014) for different purposes. For prediction purposes that either a user attend or did not attend a call, a static data log starting in 2012 to June 2014 was used by Pielot (2014). In Sarker et al. (2016), researchers used a smartphone call log from July 2014 to mid-October 2015 to model distinct smartphone user behaviors. Khan et al. (2022) put forward a machine learning advanced user behavioral model named the behavioral adversarial traversal tree, experimenting on an individual's real-life notification log data collected for six months. Mafrur et al. (2015) used a life sensing log of smartphones for two months to unveil user behaviors for recognition services. For the modeling of user preferences for customized context-aware recommendations, Srinivasan et al. (2014), have also utilized the contextual data of a three-month static period. It predicts that under certain contexts, which app is preferred by an individual user. Mehrotra et al. (2016) used a static notification log that consists of 11,185 notifications. All the approaches we discussed were used to model user phone usage behaviors utilizing the entire smartphone dataset log for a static period. However, they did not incorporate the latestness of user behavioral patterns utilizing their smartphone log, which is crucial for predicting user behaviors accurately and in which we are interested.

### 2.2 Dynamic period dataset extraction problems

Many researchers have used the latest smartphone dataset to produce rules based on individuals' latest behavior to forecast their future behavior from entire historical logs. In Visuri et al. (2019), authors try to incorporate the notifications content and context information to develop the personalized prediction model. The problem is the same; they also use the latest static data log and consider it a user's current behavioral log. For instance, Sarker et al. (2016) utilized the previous three months' data as the current call log of a user to develop a call proposition method for a customizable speed call list. Likewise, in Sarker and Kayes (2020), the researchers used the last 24 h' call record as the latest record of the users to forecast the next calling pattern for 24 h for a specific user. Although taking the latest log of the data can show current behaviors, this approach may not reflect the latest behavioral activities of the users, as user behavior changes frequently over time in real life. *Problem statement* Providing a NotifyMiner dataset holding the detailed multidimentional contexts with relevant notification interaction behavioral activities of a specific smartphone user. Our aim is to develop an approach which automatically identify the user's smartphone interaction behavioral change and dynamically identify the ideal period of time which holds the latest user latest behavior. The Proposed should dynamically predict user behavior. So, unlike old methods, in PredictionMiner, we present the latest rule-based system that vigorously regulates the ideal period of the latest notification log of individual users according to their latest current behaviors. By using that latest behaviors-based log, our approach removes the obsolete rules and produces the current co-occurring patterns for individuals. Those co-occurring patterns are based upon user's latest smartphone interactions by incorporating the notification's log.

## 3 Research challenges and main contribution

One challenge we encountered while initializing this research was identifying the ideal period for behavioral extraction. The ideal period is the time period in which we can expect to find the user's latest behaviors. The duration of this time period is crucial for selecting valid latest rules. For instance, if we take a brief time interval (e.g., two weeks of data) as representative of an individual's current behavior, there may not be enough data occurrences to indicate significant behavioral rules. behavioral rules with insufficient support are not likely to be effective in representing user behavior (Sarker et al. 2016). On the contrary, if we take behaviors over a prolonged time interval (e.g., 8 months) as representative of individual behavior, the support would be high, but the behavioral variations will be greater. So, while these behavioral variations in different contexts may have high support value, they decrease the confidence value, and the rules may lose validity because of a lower confidence threshold. Therefore, the main challenge was to find the ideal time period that holds the latest log of data.

Another challenge is selecting the machine learning approach for extracting rules. Within the field of data mining and machine learning (Sarker and Salim 2018), the most commonly used approaches for extracting rules are classification rule mining and association rules mining. We have chosen the association rule mining approach as compared to the classification rule mining approach. The reason is that in the classification approach that is used for rule mining (e.g., decision-based tree), the set of rules produced does not consider individual user-to-user preferences, resulting in inflexible decision-making. Additionally, classification rule mining approaches have less reliability and cannot assure that the produced set of rules will have high prediction accuracy (Mehrotra et al. 2016; Freitas 2000).

Our key contribution is the development of a method, PredictionMiner, for pattern mining services. Using minimal time resources, PredictionMiner runs on the entire extracted dynamic latest notification data log and mines user co-occurring patterns. The proposed approach is flexible in permitting the addition of more newly generated contexts and makes it easy to extract the user's co-occurring patterns or provide more comprehensive patterns affiliated with a certain type of context (e.g., temporal location, application type, etc.). In the proposed approach, a favorable period of time is dynamically determined for identifying the changing behavior of individual users and setting a threshold for behavioral change. If the varying behavioral limit for a specified user is not reached, we consider it as no change in behavior, and the whole smartphone notification log data will be utilized to uncover the behavioral patterns. When the latest log data has been identified in the proposed method, the old, outdated rules are deleted, which do not represent the changing behavior of the user. The major contributions of this work are:

- We proposed an approach that dynamically identifies the co-occurring patterns of individual smartphone users on the basis of collected notification log data.
- We devise a mechanism with which we can dynamically find the Contemporary period of the latest log data established on the behavior patterns changes of separate smartphone users rather than the specific time slot.
- The proposed approach mines the compelling latest rules representing the most delinquent behaviors of the users towards their smartphone notifications.
- Unlike the old approached, in the proposed approach the predictions are being done on the basis of user's latest smartphone interaction behavior.
- The experiments are conducted on solitary real-world notification datasets to assess our behavior modeling-based method contrasting with subsisting core models to exhibit the efficacy of the PredictionMiner in making predictions.

The contents of the following article is organized in various sections. Next, in Sect. 4, the main notions regarding the proposed approach are presented. In Sect. 5, we describe our methodology and provide details on how the algorithm is employed. The results after the experiment will be discussed thoroughly in Sect. 6. Last, Sect. 7 holds the conclusion.

*Algorithm 1 Context Combination Algorithm*

**Input:** $DS_{week} = M_1, M_2, M_3, \ldots, M_n$ // week-wise data with each instance $M_i$ contains amount of Categorical contexts

Intitialize_assoc→Ø

**Procedure PRO** ($DS_{week}$, list_of_contexts);

**Step 1** **if** list_of_contexts is empty **then**
  **return** assoc_list
**end**

//for the identification of precedence of contexts calculate the entropy and information gain

**Step 2** $X_{best}$←the best context classification example

**Step 3** **foreach** value of context con ϵ X **do**

**Step 4** assoc←create_conext_combination(con,initialize_assoc)

**Step 5** Add association to assoc_list;
//division in to subsets and create corresponding tree for each subset

**Step 5** $D =_{sub} \leftarrow DS_{week}$ subset which has con value

**Step 7** **if** $D_{sub}$ is empty **then**

**Step 8** start_assoc← Ø

**Step 9** **else**
//delete $X_{best}$ befit context from list

**Step 10** list_of_context←context_list -$X_{best}$
//generation of remaining attributes association iteratively

**Step 11** PRO($D_{sub}$,list_of_contexts)
**end**
**end**

**STEP 12** ***Output*** ($asooc_{list}$ list of contextual combination)

## 4 Materials and methods

### 4.1 Definitions and discussions

The main notions regarding pattern mining with the latest rules are discussed below.

**Definition 1** Con ={ $con_1, con_2, ....con_n$ } *indicates a set of contexts occur together in different dimensions at a given timestamp with their corresponding decisions; e.g., the contexts* { Meeting,Messaging_app } → { Dismiss} could be followed by { Meeting,Messaging_app, Colleague, Wed[10:30-11:30] } → { Dismiss }The contexts which are output to base rules miner are the common co-occurring patterns denoted in the form of association rules.

**Definition 2** *Let, $c_1$ is the occasions count (logs) in the whole set of notifications data DS, which is sorted for the time being. A current notifications dataset* $\mathrm{DS}_{\mathrm{current}}$ *is the subset of the dataset which holds the utmost current details of DS on the basis of timestamps of the range $c_2$ whereas $c_2 \leq c_1$.* The dynamic ideal time period of the dataset can be utilized to identify the current rules for individual user's that's why the proposed method must hold the capability to point out individual behavioral changes from all the notifications logs without making any predetermined suppositions.

**Definition 3** A The Rule $R_1$ can be considered as an outdated rule if $B_1$ is changed (different behavior) for that particular context $A_1$ utilizing current log data $D_{\mathrm{current}}$i.e.,$A_1 \rightarrow B_2$ and $B_2 \neq B_1$where $B_1$ and $B_2$ represents the current and past behaviors .

**Definition 4** A new latest rule is not a rule which is created from an entire log of data; it is a rule which is created by a current/recent log of data $\mathrm{DS}_{\mathrm{current}}$. Let a rule X: M→N be developed by employing the current log of user's mobile notifications $\mathrm{DS}_{\mathrm{current}}$, in which M denotes the user's context-sensitive details, and N represents the notification interaction behaviors. Following rule X can only be acknowledged as the current behavioral rule $R_{\mathrm{current}}$ if that rule has not been discovered before from the entire notification data log. Although $\mathrm{DS}_{\mathrm{current}}$ is the subdivision of the entire DS, so those rule types cannot be uncovered by employing the whole notification data log for the reason of less confidence threshold (assume 80%).

**Definition 5** A robust rules governance is required to acquire the entire set of up-to-date rules by not considering a few suppositions regarding the time an individual user behavior can change. Let, $X_{\mathrm{basic}}$ is the set of rules originating from the whole notification dataset DS, and $Y_{\mathrm{current}}$ holds another ruleset which is originated through the current data log $\mathrm{DS}_{\mathrm{current}}$. An entire list of current upgraded rules $Z_{\mathrm{updated}}$ ought to be the combined result of those two rulesets, like, $Z_{\mathrm{updated}}$ combination ( $X_{\mathrm{basic}}$, $Y_{\mathrm{current}}$ ). $Z_{\mathrm{updated}}$ ruleset not only contains prominent rules of individual smartphone users but also shows the current updated behavioral patterns, which can be suitable for modeling smartphone usage behaviors in actual-world operations.

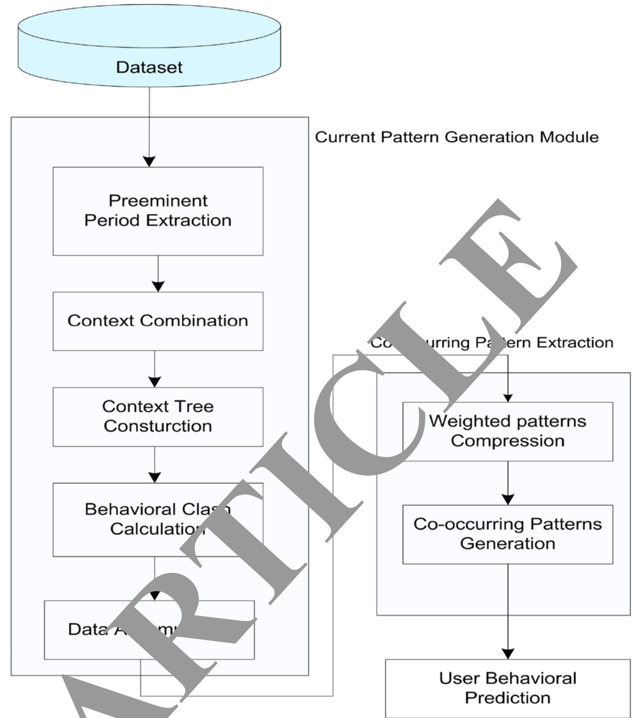



**Fig. 1** Prediction miner algorithm

## 5 PredictionMiner approach

In the following section, we will discuss our prediction miner approach step by step, starting with the procedure of extracting the latest user behaviors and then mining the co-occurring patterns to make individual smartphone user behavioral predictions using their notification log data. The proposed methodology is composed of two parts, as shown in Fig. 1. The first part involves the process of extracting the individual latest smartphone interactions, while the second part includes the extraction of co-occurring patterns that occur together.

The input to our approach is the real smartphone notification log DS. After going through several processing steps, the proposed method is capable of producing an entire set of upgraded rules for individual users. Later, duplicate patterns are removed using the pattern compression module.

### 5.1 Extracting preeminent period for latest data

The first step of the current pattern generation module is to extract the ideal period from the current log. Since time has a significant impact on user behaviors (Halvey et al. 2006), we initially divide the notification data log on a weekly basis. The reason for selecting the weekly division is because individuals' behavior is likely to remain the same during the same period of the week (Mon,..., Sunday).We assume that user behavior will keep repeating (e.g., the office timing of the user remains the same on all days of the week). The weeks are divided in the week-wise data division, where $Wk_1$ is initial week and $Wk_n$ is represented as the most current week, which holds the user's latest behaviors.

Algorithm 1 shows the context generation method. The input includes week wised data $D_{\text{week}} = m_1, m_2, ..., m_n$, holds the collection of occasions with definite contexts and resultant data in the form of an association-list $\text{assoc}_{\text{list}}$. The $\text{assoc}_{\text{list}}$ is a subset $D_{\text{sub}}$ that holds the value of context. In case the subset is not vacant, then we iteratively repeat it in favor of every context and create associations by considering each context based on the context precedence. After the list is empty algorithm generates the list $\text{assoc}_{\text{list}}$.

## 5.2 Behavioral clash calculation

Behavioral clash calculation is the most important part in the process of identifying the behavioral change. After the contexts are associated later, we calculate the behavioral clash for every association among adjacent weeks. We start from identifying the prominent behavior (which occurs maximum times) (Sarker et al. 2018). For a particular association (e.g., morning, lab); we do not expect 100% user's like behavior meanwhile, we can consider it as 80% click, 60% dismiss, and 50% deferred notifications so in this particular association, click ought to be the prominent behavior. For finding the clash behavior, scanning will be initialized by us from the utmost current week $Wk_n$ into preceding weeks. $Wk_{n-1}, Wk_{n-2}, ...., Wk_1$ individually determining the behavior change for every context in adjoining weeks. After determining whether there exists a behavioral clash or not for every association created in the previous part, we use Eq. 1, stated below, to compute the clash score. Here $\text{assoc}_{\text{total}}$ exhibits the complete count of associations determined from week $Wk_n$ and $\text{clash}_{\text{total}}$ denotes the complete count of behavioral clashes exhibited by comparing against arising associations in week $Wk_n$.

$$\text{ClashScore}(\%) = \frac{\text{clash}_{\text{total}}}{\text{assoc}_{\text{total}}} \times 100 \tag{1}$$

***Algorithm 2:** Behavioural Clash Score Calculation Algorithm*

```
Input   DS_week  Week-wise data DS_week1 and DS_week2 for W_n and W_n-1
        Procedure Start
        Step 1 assoc_list1←association_generation(DS_week1);
                  //by using algorithm 1
        Step 2 assoc_list2←association_generation(DS_week2);
                  //clash score count initialization
        Step 3 initialize c←1
            Step4   foreach assoc in assoc_list1 do
            Step 5    if   same kind of association is found in assoc_list2 then
                                //prominent behavior identification for assoc
            Step 6         B1←prominant behavioridentification(assoc,DS_week1)
            Step 7         B2←prominant behavioridentification(assoc,DS_week2)
                            //prominent behavior checking
            Step 8         if   B1≠B2 then
            Step 9                  Increment c
                           end
                      end
                    end
            //behavioral clash score percentage(%) calculation
            Step 10 score←Percentage_calculation (c,assoc_list1)
            return score
        Step 11   Output (Behavioral clash score: score)
```

Algorithm 2 shows the process for calculating the behavioral clash score. For input, we take the adjoining weeks' data into account: $\mathrm{DT}_{\mathrm{week}}$ for week $Wk_n$ and $\mathrm{DT}_{\mathrm{week2}}$ for week $Wk_{(n-1)}$, each of them holds a set of instances $M_1, M_2, ...., M_n$. The data output would be a behavioral clash score in the form of parentage. From Algorithm 1, we generate the contextual association; later, from Algorithm 2, for every association, we examine whether prominent behavior is similar or changing. If various prominent behaviors changes are found, the behavioral clash score rises. The percentage (%) is calculated for the behavioral clash at that time. Finally, algorithm 2 returns the behavioral clash score.

### 5.3 Data accumulation

Data accumulation is the last step in finding the preeminent log of latest data, For data accumulation, the weekly data depending upon identical behavioral patterns detected by the behavioral clash is accumulated. Rather than probability to determine the behavioral similarity, behavioral clash score between two adjoining weeks is used because similar contextual information is not expected every week. In case the behavioral clash score between two weeks is 0 (%), it shows that the user behavioral pattern in both weeks is almost the same (Sarker et al. 2017). We start to accumulate from the most current week $Wk_n$ to preceding week $[Wk_{(n-1)}, Wk_{(n-2)}, ....]$ until we get the prominent variation in the behavioral clash score among two adjoining weeks. We then set the threshold for the current behavioral patterns. By examining the comprehensive behavior in all the datasets, the prominent change is met once it surpasses the average variations outcome. $X_{\mathrm{total}}$ is the total behavioral clash score and $No_{\mathrm{weeks}}$ is a total no of weeks in the dataset, subsequently from Eq. 2, the average of a score can be defined as:

$$\mathrm{AverageScore} = \frac{X_{\mathrm{total}}}{\mathrm{No}_{\mathrm{weeks}}} \times 100 \tag{2}$$

By accumulating the most latest four weeks' data (from $Wk_{(n-3)}$ up to $Wk_n$), Fig. 2 shows the current behavioral pattern of an individual smartphone user. It exhibits that week $Wk_n$ is the utmost latest week whereas $Wk_{(n-3)}$ is a limit for latest user behavior patterns, it is a behavioral pattern that depends upon the relevant contexts before the week $Wk_{(n-3)}$ (from $Wk_1$ up to $Wk_{(n-3)}$) are noted as a previous behavior. The user behavior patterns after $Wk_{n-3}$ before the utmost latest week $Wk_n$(starting $Wk_{(n-3)}$ to $Wk_n$) are called as latest behavior of a user. In case we notice no variation starting from $Wk_1$ log to the end of the log $Wk_n$ then entire log behavioral patterns are considered as the latest patterns instead of promptly identifying the count of intervals initially, the proposed algorithm interactively determines the preeminent period of the latest log from an individual's smartphone notifications dataset. That's why the weeks count and time limits for the latest log will be inconsistent for every user depending upon how the user behavior changes in separate contexts with the passage of time.

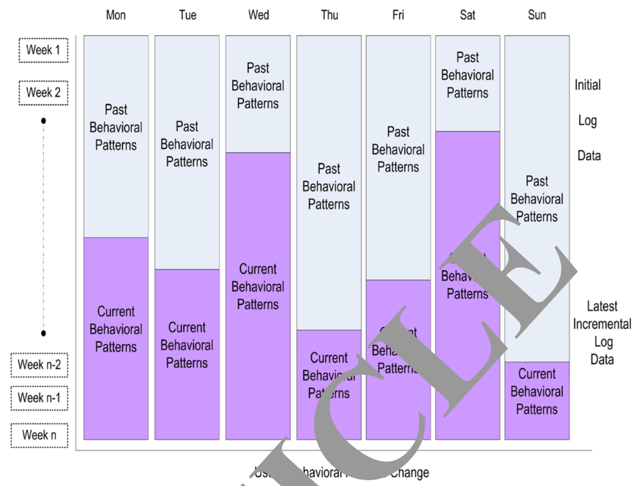



**Fig. 2** Data accumulation example: Data Accumulation technique of alike behavioral patterns for the identification of log data accommodating updating behavioral patterns

### 5.4 Machine learning-based latest rules generation

After the determination of the of the latest data log $DS_{latest}$ rules will be generated using that log. For the rules production we will use our previous machine learning rule-based method names behavioral adversarial traversal tree (Khan et al. 2022) on the latest log. We are using that approach because that approach generates more generalized rules as compared to other approaches. For rules generation this approach first produces a tree based on the context precedence. When the tree is generated, the rules are devised via traversing form the root node to the node which make decisions. The nodes are devised based on the presidence of contexts. To identify the precedence of context, we first calculate the information gain (Srinivasan et al. 2014), a statistical quality that computes the entropy and then checks that at what degree the specified context distinguishes the practicing datasets into desired behavior class present in the data log shown in eq 3.

$$M(N) = -\sum_{x \in X} p(x) \log 2 p(x) \tag{3}$$

Here in Eq 3, N is the current data whose entropy is being calculated, X is denoted as set of classes in N. The rules produced from the following approach are easily understandable by humans. ($\mathrm{relevant}_{\mathrm{contexts}} \rightarrow \mathrm{human}_{\mathrm{behavior}}$). The rules produced in this approach met a certain level of confidence with less number of contexts. When more number of contexts are considered the proposed approach also able to meet that

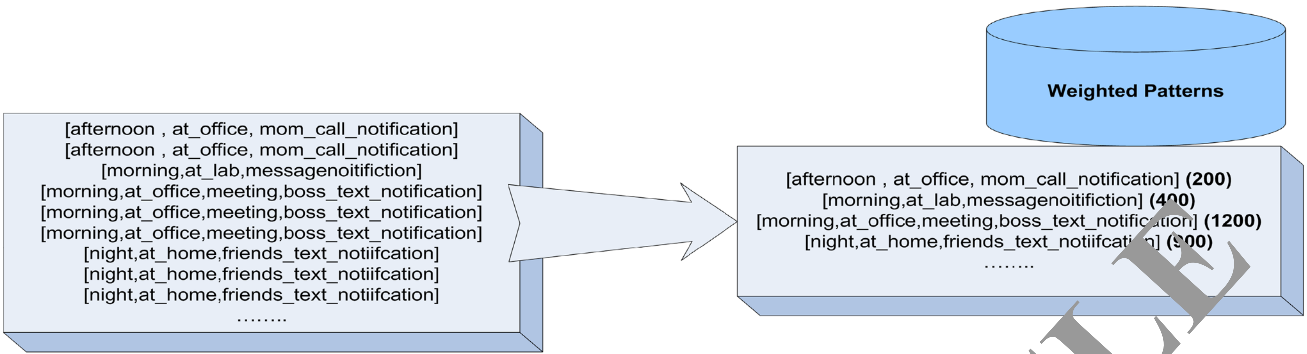



**Fig. 3** Weighted patterns compression

expectation. In the proposed method the rules will be generated by considering the dynamic latest log of data $DS_{latest}$. We merge the latest rule-set with the initial ruleset $R_{initial}$. For the generation of the initial ruleset we will use the same approach as we have discussed above (Khan et al. 2022). In the process of merging we will remove the expired rules from $R_{initial}$ which can't expresses the latest behavior of a user. Same from ruleset $R_{latest}$ our method will also delete the rules which are present in initial ruleset $R_{initial}$. Thus, as a result the complete updated set of latest rules will be produced by merging the initial and the latest ruleset e.g., $R_{updated} = \text{merger}(R_{initial}, R_{latest})$. Those updated rules will be used later in the pattern compression module for the generation of the individual co-occurring patterns.

### 5.5 Weighted contextual patterns compression

Each latest rule is composed of contextual patterns that happen simultaneously at a given timestamp e.g., Time, Location, apptype.In the course of the pattern extraction procedure, we append the derived items like location context, app-type context, relation context, and time slot context shown in Fig. 3. The main reason is to produce more comprehensive rules where possible because the events always occur in proper contexts and are different for different relations as discussed previously. The preset sampling period introduced for creating the items is 25 s. During extraction of patterns rules, it is noted that due to the repeated nature of contextual data, the pattern extraction leads to duplicate extractions, e.g., if the time for sample is 25 s and the set of contexts Lab, Lifestyle_app, friend→click in two months period occurs for 120 times then there will be 220 such contexts. So, rather then repeating the patterns, we compress the patterns by associating the weight with every pattern, signifying the number of periods it iterates. The weighted patterns will be later used for the process of generating the co-occurring patterns.

### 5.6 Co-occurring patterns generation

After the pattern compression, the proposed method produces frequent co-occurring patterns of all dimensions, representing which recent co-occurring context activities happen simultaneously. During the pattern generation process a certain confidence threshold is introduced. A context can said to be frequent if and only if it happens as equal to the confidence threshold stated. For example, frequent contexts are { Work, Entertainment_app, Netflix, Sat[slot1]}. From those contexts, possible patterns are generated as long as it surpasses the predefined confidence level. Another case of an association rule from the previous example can be like {Work, Entertainment_app, Netflix, Sat[slot]} → {dismiss}; the following rule can be able to predict if a user is more lenient towards dismissing a notification when provided the latest context. Formally for pattern generation, let Con= $\{Con_1, Con_2, Con_3, ...., Con_n\}$ are the set of patterns contexts, and domain Dm=$\{d_1, d_2, d_3, ...., d_n\}$ are the designated fields. DS is the set of records from the notification's log, where each record of notification has a specific event associated with it depending upon user behavior (i.e., deferred, reject, accept).

### 5.7 User behavioral prediction

The behavioral predictor helps us to understand the future that in different circumstances how the user will interact with different smartphone notifications. The rule set generated from a co-occurring pattern generation module can be utilized for predicting the individual's future behavior shown in algorithm 3. For better prediction proposed approach is be more helpful because it utilizes the latest individual behaviors with the removal of repeated co-occurring patterns. Our prediction engine takes contexts like (Afternoon, Lab) and the resultant context which ought to be predicted, e.g., click as an input. While taking it as an input we look for the association rules produced by PredictionMiner whose ascendants would be a subset of the input contexts and subsequent is equivalent to the destinated context class. An example of the matching rule is: {afternoon,gmail}→click with confidence 0.8. The list of predictions are returned based on multiple contexts which are found in matching association rules. If we found one more rule e.g., {Afternoon,calling_app} →{click}with a confidence of 0.9, we return both the calling app and Gmail as a candidate for sending notifications. Predicted contexts ranking is dependent upon the declining order of confidence score for every distinct prediction. During the prediction process among all the rules which are matched those rules will be considered whose confidence value is calculated as the highest e.g., for rule.g. for rule{ afternoon, calling_ app} confidence value would be 0.9 and for another rule {afternoon,calling_app,mom_call}with a confidence of 0.9. if two separate predictions confidence values become equal, those rules are preferred, which holds more number of contexts.

---

***Algorithm 3:*** *Behavioural Contexts Prediction Algorithm*

---

***Input*** Context: $C = C_1, C_2, C_3, \dots C_n$ Each instance $C_i$ holds he contexts values with its behavior B and introduced confidence threshold $C_{th}$

<u>**Procedure Start**</u>

```
Step 1  Context_list=clMethod PDT(B,cl)
   Step 2  If all values of contexts cl with corresponding
           |  behavior B present in the list then return the
           |  result with 100% confidence value
           end
   Step 3  if context_list is empty then Return [illegible] as [illegible]ty
           |  with its context value
           end
   Step 4  foreach Context_list cl∈R_updated do
   Step 5  |  triverseRules(R_updated)
   Step 6  |  if  cl==R_updated
   Step 7  |  |  confidence++
   Step 8  |  |  if confidence==C_th
   Step 9  |  |  |  then
   Step 10 |  |  |  ruleset (R_updated, {context_list, C_i}, B)
   Step 11 |  |  |  else
   Step 12 |  |  |  delete R_updated from list
           |  |  end
   Step 13 |  |  triverseRules(R_updated)
           |  end
           end
Step 14    Output (Predicted Contexts)
```

---

## 6 Experimental evaluation and results

In this section, the performance comparison of the PredictionMiner approach has been done for the validation of the proposed method. Various experiments have been done on individual mobile-phone notification data logs. In the experimental section, we will answer the following questions for the authentication of the proposed approach.

**Q1:** While producing the rules, is there any difference between the PredictionMiner latest set of rules and the initial set of rules that originated from the entire smartphone notification dataset?

**Q2:** What are some common sample patterns and how can they be used?

**Q3:** Is the prediction Miner Approach personalized? Using the behavioral clash score, how can we find individuals' pre-eminent periods of recent notification logs?

**Q4:** How effective is our PredictionMiner approach, and what are the precision-recall results of our predictions?

To answer the above questions, different kinds of experiments are performed on real-life notification datasets of different smartphone consumers. The notification datasets used in the following experimentation are collected by individual smartphone users in the NotifyMiner project (Khan et al. 2022). NotifyMiner dataset collectively holds the constraints like device id, user id, app name, app type, seen time, arrival time, user's GPS location, user interaction, user condition, and network usage. Those raw datasets have been used in our experimentation section to validate our approach PredictionMiner. In order to evaluate our approach, we have experimented with eight notification log datasets. The notification dataset contains 40,000 notifications, denoted as USR01, USR02,...,USR08.

### 6.1 Rules discovery effect

To answer question no 1 in the first experiment, we have checked whether the proposed approach has any effect on the discovery of the latest rules or not. For this, we have shown the rules production comparison between eight datasets USR01, USR02...USR08 for a specific 75% confidence threshold is shown in Fig. 4. For comparison purposes, we have used the base model that holds the whole dataset which means that if there is a little change in human behavior the older rules will nullify the latest rules and our proposed approach, which considers the vigorous latest log for a pre-eminent period. By analyzing the results, it is seen that rules produced from our method are more in count as compared to the base model. The main reason behind it is that, in the base model, the former adverse rules would invalidate the latest rules if there is a change in human behavior. Whereas, in the proposed approach, we determine the latest behavioral rules based on the user-changing patterns in the dataset and pro-

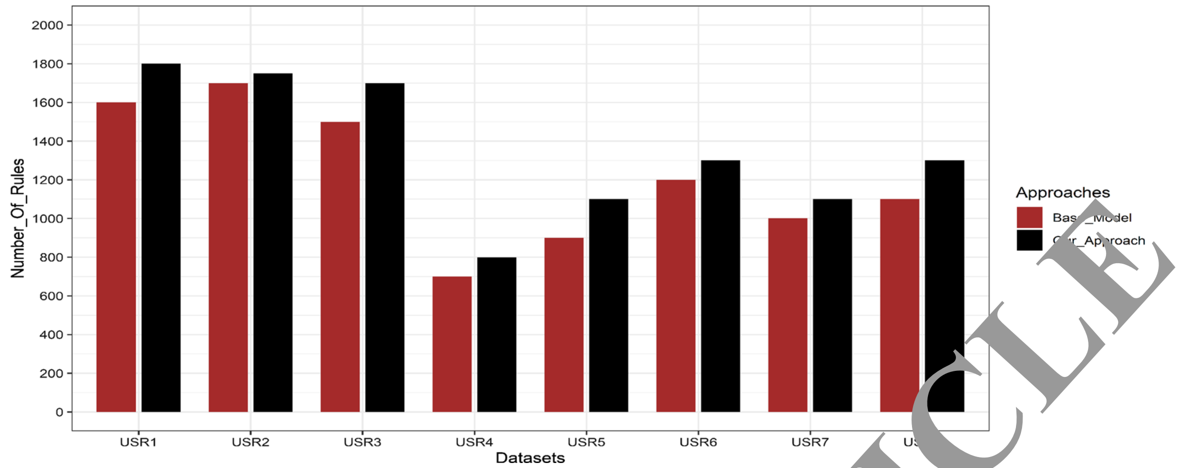



**Fig. 4** Impact on the count of rules produced from the proposed method with the basic rules discovered from the base model

duce the entire set of current upgraded behavioral rules. So, the rule count rises because newly discovered rules depend on the user's latest behavioral patterns.

## 6.2 Behavioral clash score effect on discovering individual latest log

To identify the latest individual log in the following experiment, we will display the impact of the behavioral clash score. This experiment will answer our question 3. Table 1 shows the behavioral clash score for a sample user for every adjoining week. $W_x$ denotes the latest week in the dataset. Upon observing Table tab1, it is seen that the individual behavior doesn't remain the same over the time period. For the adjoining weeks in the start (Week $[W_x]$ to week $[W_{(x-1)}]$) the behavioral clash score is the same as zero because the behavior is matched during those weeks in the same contexts. But if we observe some adjoining weeks like (week $[W_{(x-5)}]$ to week $[W_{(x-6)}]$) the behavioral clash score is more than zero. It's because the user behavior is not matching under the same contexts. For the individual dataset (USR5), the behavioral clash score is not always zero. Using Eq. 2, we calculate the average behavioral clash score as (1.95%) for an individual using a real threshold value instead of assuming the random threshold value. In Table tab1, it is identified that the user behavioral patterns are the same from starting $W_x$ to week $W_{x-2}$ ($\geq 1.95\%$). For the particular user, a prominent change has been seen between weeks $[W_{(x-5)}]$ to week $[W_{(x-6)}]$ we can say that the last 7 weeks of the users is the latest log period which can represent the latest updated behavior of the user. In Fig. 6, it is shown that the clash score differs between different users based on how consistent their behavior is. Figure 5 shows that the user's preeminent log period also differs from each other based on their different behavioral patterns. So, we figured out that using a static period of the log is not the right option to model the individual's smartphone usage behaviors. The log period should be personalized based on user characteristics. For prediction purposes, if the log period is not correctly identified, the prediction of user behaviors cannot be correct.

**Table 1** Sample user dataset (USR05) behavioral Clash Score Count

| Adjoining weeks | Behavioral clash score (%) |
|---|---|
| Week $[W_x]$ to Week $[W_{(x-1)}]$ | 0 |
| Week $[W_{(x-1)}]$ to Week $[W_{(x-2)}]$ | 0 |
| Week $[W_{(x-3)}]$ to Week $[W_{(x-4)}]$ | 0.23 |
| Week $[W_{(x-4)}]$ to Week $[W_{(x-5)}]$ | 1.45 |
| Week $[W_{(x-5)}]$ to Week $[W_{(x-6)}]$ | 2.98 |
| Week $[W_{(x-6)}]$ to Week $[W_{(x-7)}]$ | 0.52 |
| Week $[W_{(x-7)}]$ to Week $[W_{(x-8)}]$ | 3.33 |
| Week $[W_{(x-8)}]$ to Week $[W_{(x-9)}]$ | 4.78 |
| Week $[W_{(x-9)}]$ to Week $[W_{(x-10)}]$ | 1.10 |
| Week $[W_{(x-10)}]$ to Week $[W_{(x-11)}]$ | 2.22 |
| Week $[W_{(x-11)}]$ to Week $[W_{(x-12)}]$ | 4.96 |

## 6.3 Co-occurring pattern generation

To answer question 2 in this experiment, we have visualized some common co-occurring patterns from recent notification logs of the sample users. We have visualized the smaller portion of association rules produced through Prediction-Miner. The sample user USR07 patterns are visualized in Fig. 7. Figure 7a shows the base patterns mined through the co-occurring pattern generation module, and in Fig. 7b, we have shown the detailed patterns of notification clicks. Each matrix row shown in Fig. 7 represents the association rule. The rules subsequent is represented along the row names,

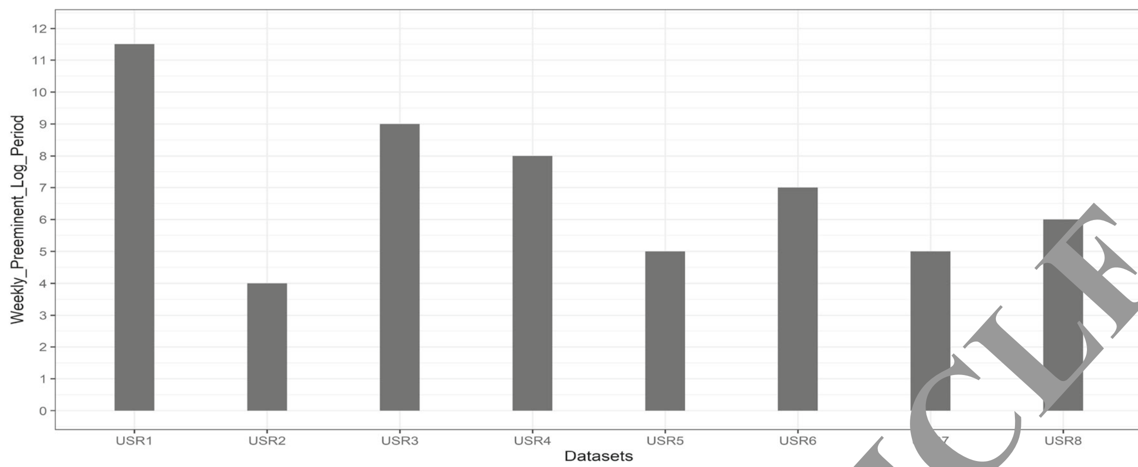



**Fig. 5** Effect on Individual's Preeminent Log varies between individuals

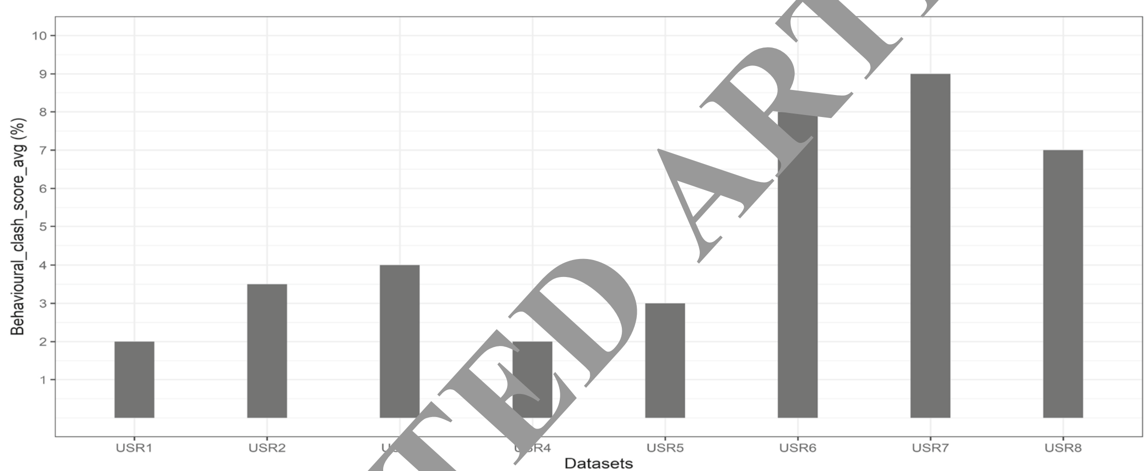
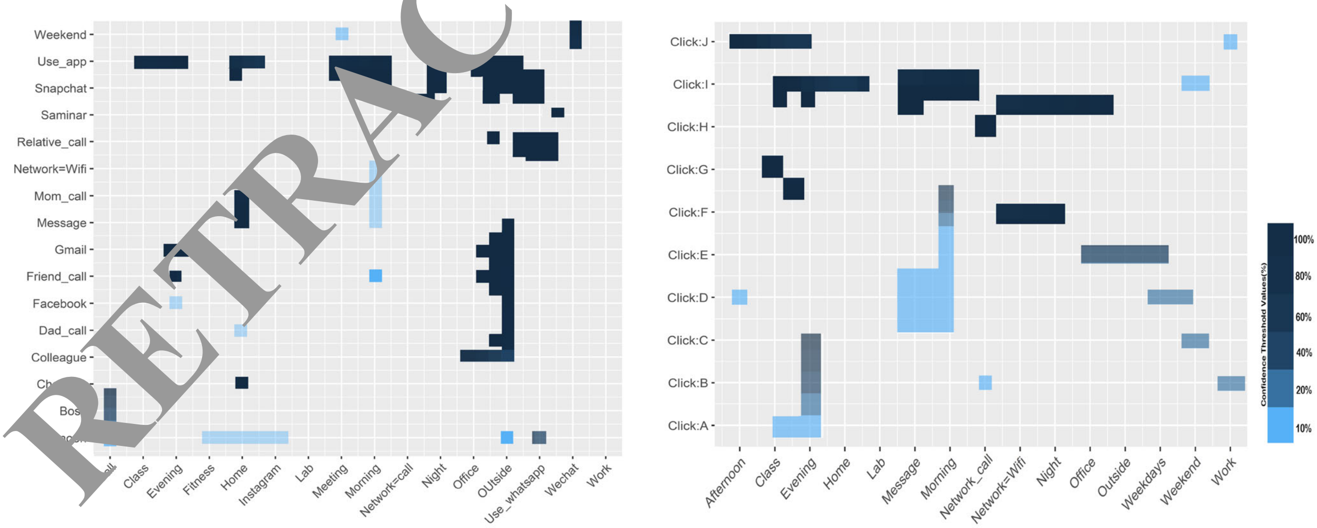



**Fig. 6** Effect of behavioral clash score on individuals




(a) Overall Co-Occurring Patterns (b) Click Notifications Patterns (c) Key

**Fig. 7** Co-occurring patterns of sample user USR07 and the antecedent of the rules is mentioned along the column names. For every row, the cells can be colored if the association rule ascendant is specified in the column and the row specifies the subsequent. The cells' color depends upon the confidence value denoted as a percentage presented in Fig. 7c. Figure 7a presents the base patterns for prolonged durational contexts for a sample user; it shows the base patterns for prolonged durational contexts for a sample user, such as calling patterns at different locations. For the following patterns generation, 1% threshold level of support is used. The first rule indicates the sample rule that a user uses app when the location is class; also, the user is more likely to use apps in the evening time in cell ID C2. While we move along to cell ID C2, we see that a user is also more prone to use the app when he is in a meeting or in the morning time. Similarly, if we move along row 16, we can see that when the location is outside user became more attentive towards different kind of apps. Figure 7b shows the click patterns for a sample user. We have selected the click pattern because, right now, we are more interested in the occasions at which the user is more likely to click the notifications, as it can help us to predict the user's future behaviors precisely. The click patterns could be used to preload the specific notifications when the ideal occasion is met. We see that the user-click ratio increases towards different apps when the time is morning or evening. Similarly, a user mostly clicks on the calls if they are from B or H relations. In the next part, we will show how the mined co-occurring patterns can help predict user future notification interactions. of social apps, i.e., Instagram, Fitness, etc. Those patterns can be utilized in various applications, like (1) pre-loading specific app types based on user temporal locations and (2) pre-loading specific app notifications when the user is more prone to attend them.

Figure 7b shows the click patterns for a sample user. We have selected the click pattern because, right now, we are more interested in the occasions at which the user is more likely to click the notifications, as it can help us to predict the user's future behaviors precisely. The click patterns could be used to preload the specific notifications when the ideal occasion is met. We see that the user-click ratio increases towards different apps when the time is morning or evening. Similarly, a user mostly clicks on the calls if they are from B or H relations. In the next part, we will show how the mined co-occurring patterns can help predict user future notification interactions.

## 6.4 Predictions precision recall trade off

To answer question 4 in the following experiment, we will check the precision-recall tradeoff of our predictions. For it, we have selected a use case in which we check for an app and its click prediction. We employed quality evaluation metrics from the classification literature to assess the accuracy of click prediction.

- Precision: The portion of time the user clicks on the specified notifications to complete his task.
- Recall: The proportion of the time the notifications are shown to the user.

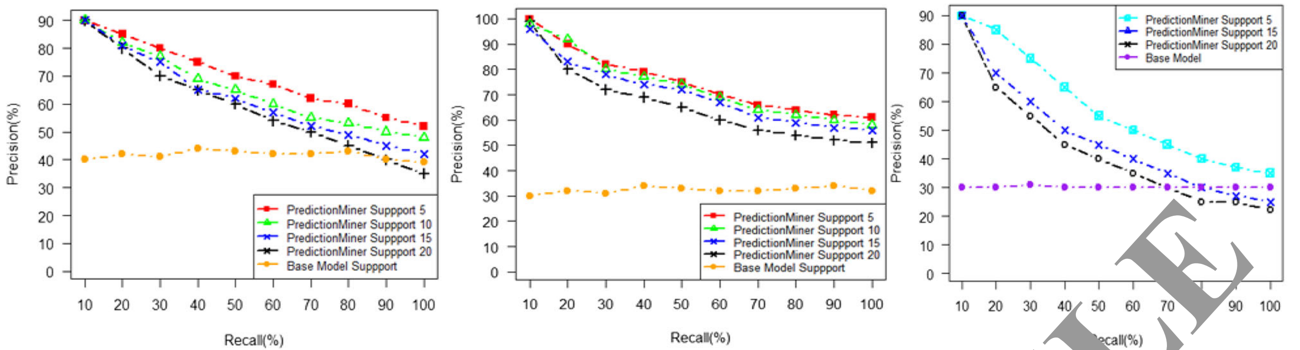
Contrarily, the lower precision results in higher recall results. In this experiment, we have introduced the higher confidence rank threshold to represent whether the user has clicked or not clicked the notification. When the confidence threshold is low, it produces an increased recall, and if a higher confidence threshold is high, the precision will be higher.

To evaluate the following tradeoff, we again run our PredictionMiner method on the longitudinal trace of recorded data with notification click record via increasing the confidence threshold value. In Fig. 8a and b, we have shown the click predictions by introducing different supports. In Fig. 8c we have shown the dismiss predictions by introducing different supports. Fig. 8a–c shows that the prediction based on the latest co-occurring patterns outperformed most base models depending upon the area matric below the recall prediction curve. While making predictions, it is noted that generally, more than 55% time, we accomplish 89–110% betterment in precision compared with base model predictors in predicting the upcoming app is clicked or dismissed. The reason is we have used the dynamic latest behavioral co-occurring patterns of the users, while the base model doesn't consider the recentness of the user behavioral patterns. The most significant challenge in the rule mining methods is the selection of suitable support for devising patterns. Figure 8a and b shows the impact of the utter support on the PredictionsMiner click predictions. As the support decreases from 20–5, we have noticed a 4–5% improvement in predictions. Thus, it could be potentially useful to mine patterns that could occur only 5 times to improve prediction accuracy.

## 6.5 Efficiency analysis and comparison

To answer question 5, in this section we are going to analyze our approach's efficiency. We will analyze the efficiency of our proposed approach by measuring the error rate (%) and prediction coverage (%) and by comparing it with state of the art model for delivering context-specific notifications. Figures 9 and 10 present the outcome for both approaches with a specific confidence threshold (80%).We have shown the prediction results of individual users by utilizing their datasets. From Figs. 9 and 10, we have found out that our approach consistently shows better results compared with the base model for predicting smartphone interaction behavior.

As we have discussed before, the main reason is in the base model; the rules are not the latest according to user's current




(a) Effect of support on click predictions utilizing user's dataset USR01

(b) Effect of support on click predictions utilizing user's dataset USR03

(c) Effect of support on dismiss notifications prediction utilizing user's dataset USR01

**Fig. 8** Precision-recall tradeoff for click and dismiss notification predictions using co-occurring patterns

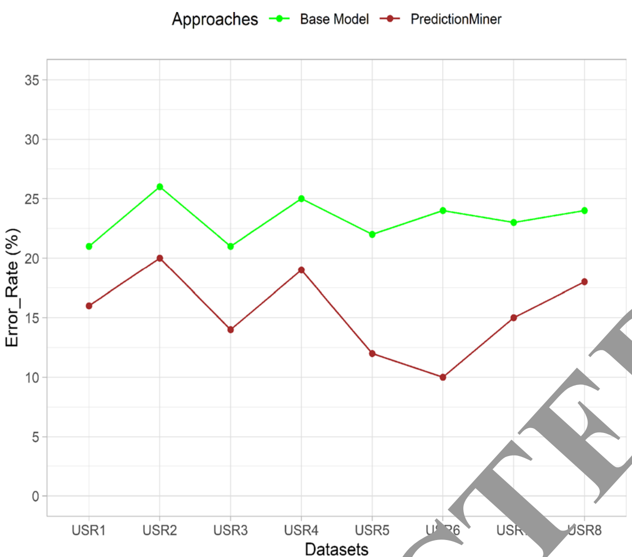



**Fig. 9** Proposed method and base model efficiency comparison with regard to error rate (%)

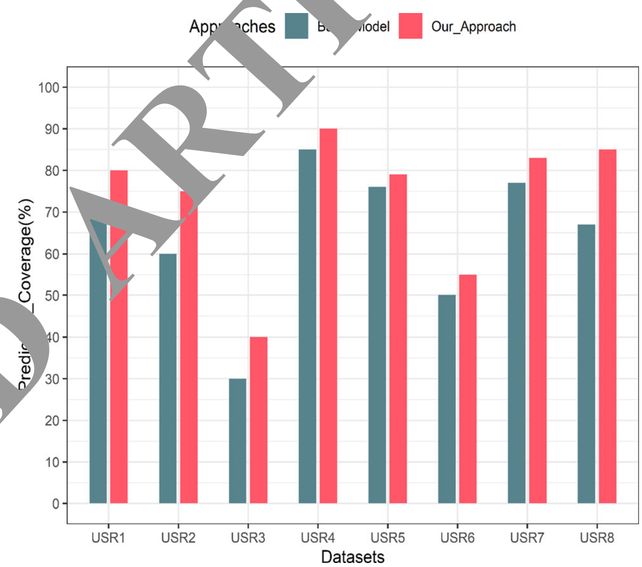



**Fig. 10** Proposed method and base model efficiency comparison with regard to prediction coverage (%)

behavior. We have noticed that while making predictions, the error rate is quite high, our proposed approach solves that problem by generating the rules that are dependent upon the user's latest behavioral patterns. It makes our method more productive and reliable by expanding prediction coverage and reducing the errors.

## 7 Discussion

Generally, our proposed machine learning rules-based model is totally personalized and exhibits the individual's current behaviors by utilizing their latest smartphone notifications log. From literature review we have analyzed that this approach is one of its kind that considers the dynamic data log for the purpose of mining the up-to-date individual behavioral rules. By comparing with previous approaches that uses the fixed period of log data(discussed in "related work and problem statement" section) it is seen that the proposed rule-based approach is more effective for predicting the individual user's smartphone interaction patterns. The proposed approach not only reduced the error-rate but also increases the prediction coverage as shown in Figs. 9 and 10.

Identification of the behavioral change of individuals is another key contribution of our work. The preeminent period of time for data log collection is dynamically devised based on the individuals behavioral change. The preeminent time period is different for different individuals based on their behavioral change. If overall the user behavior is not changed then all the log data will be considered as the recent log period, which makes this approach quite different compared to old approaches which considered only latest log of two to five months and consider it as a recent behavioral log of individuals.

Another finding of this study is the detection of the outdated behavioral rules which do not depict the latest behavioral patterns of individuals. As discussed the user behaviors towards smartphones does not remain static and they change over time making the user behavioral rules outdated and not interesting for the users. If those rules have not been detected and removed they makes the ruleset unnecessarily large and decrease its effectiveness. Our approach keeps on updating the behavioral rules. It removes the expired rules which do not depict the individuals latest behavioral patterns that, make proposed approach very much effective shown in Figs. 5 and 6.

Another key finding of this study is to remove the redundancy among the latest behavioral rules. In the rule extraction process, some redundant rules are extracted, which makes the rules extraction process pointless and creates reliability problems. It also makes decision-based problems more complex. The weighted pattern compression module associates the weight to each rules and helps us in discovery of the redundant rules and remove them on the basis of the threshold.

The recent co-occurring patterns generation module helps us to get more insight of understanding the user interaction with a smartphone in different contexts. The current co-occurring patterns helps to find the contexts which occasionally occur for the specified user. Those co-occurring patterns later helps in making the future behavioral predictions of individual users that during the following circumstances how user will interact with their smartphones. This recent co-occurring patterns generation makes the prediction process more accurate and improve the result as shown in the Figs. 7a and b and 8a and b, which is another finding of this study.

As the center of attention of this work is to produce the updated behavioral rules of individuals so, rather than the incremental data we have processed the entire dataset. The reason behind it is that the behavioral change threshold will not be reached if we use the incremental data of small period. In addition to smartphones, the proposed up-to-date rules-based approach can also be used for stock exchange updated trends-based prediction, current trend-based job market analysis, updated trend-based medical health care systems and many relevant areas where human recent interests and fondness is entangled.

## 8 Conclusion

In this research, we have proposed a novel PredictionMiner approach for mining the latest frequently co-occurring patterns. Later, we have utilized those patterns for mining user's future behavioral predictions. For this purpose, we studied the different user behavioral aspects: (1) Volatility in user behaviors.(2) Determined the preeminent period of the latest notification logs data. (3) Identified and removed expired rules which cannot represent the current user's behavioral activities. (4) Discovered the new latest user behavioral rules and using these rules mining the frequently co-occurring patterns. (5) By using the extracted co-occurring patterns later mining of user's behavioral predictions. The rule set we concluded not just contains the notable rules of specific smartphone users from all the notification data logs, but also includes their current behavioral patterns. The latest behavioral patterns exhibit the latest behavior of the user, that allowed us to make exact user behavioral predictions. However, we have used the individual notifications log data to demonstrate our method, the following approach can also be applied to diverse application domains of natural life.

In our future work, we plan to discover different research directions for improving the PredictionMiner. We will explore the co-occurring patterns of events over larger time period intervals. We may plan to execute our context prediction method comparison with other classifier-based methods, which are being widely used and have the ability to trade off precision and recall. Accessing the exact timing of user smartphone interactions and delivering customized notifications following approach can also play a vital role.

**Funding** The authors have not disclosed any funding.

**Data availability** The datasets generated and analyzed during the current study are not publicly available due to privacy Reasons and Ethical Concerns (Data includes personal information's, notifications and contacts) but are available from the corresponding author on reasonable request.

## Declarations

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

**Ethical approval** The material is the author's own original work, which has not been published elsewhere. The author takes full responsibility of the contents and data analyzed in the paper.

## References

Adamczyk Piotr D, Bailey BP (2004) If not now, when? the effects of interruption at different moments within task execution. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp 271–278

Agrawal R, Srikant R et al (1994) Fast algorithms for mining association rules. Proc 20th Int Conf Very Large Data Bases VLDB 1215:487–499

Amornchewin R, Kreesuradej W (2009) Mining dynamic databases using probability-based incremental association rule discovery algorithm. J Univers Comput Sci 15(12):2409–2428

Azuaje F (2006) Witten ih, frank e: Data mining: Practical machine learning tools and techniques 2nd edition

Belkhir A, Abdellatif M, Tighilt R, Moha N, Guéhéneuc Y-G, Beaudry É (2019) An observational study on the state of rest api uses in android mobile applications. In: 2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft), pp 66–75

Boehm-Davis DA, Remington R (2009) Reducing the disruptive effects of interruption: a cognitive framework for analysing the costs and benefits of intervention strategies. Accid Anal Prev 41(5):1124–1129

Bordino I, Donato D, Poblete B (2012) Extracting interesting association rules from toolbar data. In: Proceedings of the 21st ACM international conference on information and knowledge management, pp 2543–2546

Chen G, Li Z (2021) A new method combining pattern prediction and preference prediction for next basket recommendation. Entropy 23(11):1430

Freitas AA (2000) Understanding the crucial differences between classification and discovery of association rules: a position paper. ACM SIGKDD Explor Newsl 2(1):65–69

Halvey Martin, Keane Mark T, Smyth Barry (2006) Time based patterns in mobile-internet surfing. Proceedings of the SIGCHI conference on Human Factors in computing systems, pages 31–34

Han J, Kamber M (2011) Pei. data mining concepts and techniques. MK

Hao T, Xing G, Zhou G (2013) isleep: Unobtrusive sleep quality monitoring using smartphones. In: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, pp 1–14

Ibrahim A, Clinch S, Harper S (2022) Extracting behavioural features from smartphone notifications. Behav Inf Technol 1–19

Jiang X, Mao B, Guan J, Huang X (2020) Android malware detection using fine-grained features. Sci Program 31:1–13

Khan MF, Lu L, Toseef M, Musyafa A, Amin A (2022) Notifyminer: rule based user behavioral machine learning approach for context wise personalized notification services. J Ambient Intell Humanized Comput 1–17

Lee H, Park J, Lee U (2022) A systematic survey on android api usage for data-driven analytics with smartphones. ACM Comput Surv 55(5):1–38

Li Y, Zhang Z-H, Chen W-B, Min F (2017) Tdup: an approach to incremental mining of frequent itemsets with three-way-decision pattern updating. Int J Mach Learn Cybern 8(2):441–453

Liao Z-X, Pan Y-C, Peng W-C, Lei P-R (2013) On mining mobile apps usage behavior for predicting apps usage in smartphones . In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management, pp 609–618

Mafrur R, Nugraha I, Choi D (2015) Modeling and discovering human behavior from smartphone sensing life-log data for identification purpose. HCIS 5(1):1–18

Mehrotra A, Hendley R, Musolesi M (2016) Prefminer: mining user's preferences for intelligent mobile notification management. In: Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing, pp 1223–1234

Monk CA, Gregory Trafton J, Boehm-Davis DA (2008) The effect of interruption duration and demand on resuming suspended goals. J Exp Psychol Appl 14(4):299

Pielot M (2014) Large-scale evaluation of call-availability prediction. In: Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing, pp 933–937

Rawassizadeh R, Tomitsch M, Wac K, Min Tjoa A (2013) Ubiqlog: a generic mobile phone-based life-log framework. Pers Ubiquit Comput 17(4):621–637

Sarker IH, Colman A, Kabir MA, Han J (2016) Behavior-oriented time segmentation for mining individualized rules of mobile phone users. In: 2016 IEEE international conference on data science and advanced analytics (DSAA), pp 488–497

Sarker IH, Colman A, Kabir MA, Han J (2016) Phone call log as a context source to modeling individual user behavior. In: Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing: adjunct, pp 630–634

Sarker IH (2018) Mobile data science: towards understanding data-driven intelligent mobile applications. arXiv preprint arXiv:1811.02491

Sarker IH, Kabir MA, Colman A, Han J (2017) Identifying recent behavioral data length in mobile phone log. In: Proceedings of the 14th EAI international conference on mobile and ubiquitous systems: computing, networking and services, pp 545–546

Sarker IH, Salim FD (2018) Mining user behavioral rules from smartphone data through association analysis. In: Pacific-Asia conference on knowledge discovery and data mining, pp 450–461

Sarker IH, Kayes ASM (2020) Abc-ruleminer: User behavioral rule-based machine learning method for context-aware intelligent services. J Netw Comput Appl 168:102762

Sarker IH, Colman A, Kabir MA, Han J (2018) Individualized time-series segmentation for mining mobile phone user behavior. Comput J 61(3):349–368

Srinivasan V, Moghaddam S, Mukherji A, Rachuri KK, Xu C, Tapia EM (2014) Mobileminer: mining your frequent patterns on your phone. In: Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing, pp 389–400

Thusaranon P, Kreesuradej W (2015) A probability-based incremental association rule discovery algorithm for record insertion and deletion. Artif Life Robot 20(2):115–123

Ünvan YA (2021) Market basket analysis with association rules. Commun Stat Theory Methods 50(7):1615–1628

Visuri A, van Berkel N, Okoshi T, Goncalves J, Kostakos V (2019) Understanding smartphone notifications' user interactions and content importance. Int J Hum Comput Stud 128:72–85

Walsh S, Fraser K, Conlan O (2022) Classification and impact of call-to-actions in push-notifications. In: International conference on advances in mobile computing and multimedia intelligence, pp 3–17

Zhang Z, Li Y, Chen W, Min F (2014) A three-way decision approach to incremental frequent itemsets mining. J Inf Comput Sci 11(10):3399–3410

Zhiang W, Li C, Cao J, Ge Y (2020) On scalability of association-rule-based recommendation: a unified distributed-computing framework. ACM Trans Web (TWEB) 14(3):1–21

Zhu H, Chen E, Xiong H, Kuifei Y, Cao H, Tian J (2014) Mining mobile user preferences for personalized context-aware recommendation. ACM Trans Intell Syst Technol (TIST) 5(4):1–27

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.