Alert Now or Never: Understanding and Predicting Notification Preferences of Smartphone Users

TIANSHI LI, Carnegie Mellon University
JULIA KATHERINE HAINES and MIGUEL FLORES RUIZ DE EGUINO, Google
JASON I. HONG, Carnegie Mellon University
JEFFREY NICHOLS, Google

Notifications are an indispensable feature of mobile devices, but their delivery can interrupt and distract users. Prior work has examined interventions, such as deferring notification delivery to opportune moments, but has not systematically studied how users might prefer an intelligent system to manage their notifications. Hence, we directly probed Android smartphone users' notification preferences via a one-week experience-sampling study ($N = 35$). We found that users prefer mitigating undesired interruptions by suppressing alerts over deferring them and referred to notification content factors more frequently than contextual factors for explaining their preferences. Then we demonstrated the challenges and potentials of leveraging user actions to help predict notification preferences. Specifically, we showed that a model personalized using user actions achieved a performance gain of 39% than a generic model. This improvement is similar to the 42% performance gain using labels solicited from the user while using observable user actions causes no extra disruption.

CCS Concepts: • **Human-centered computing → Empirical studies in HCI**; **Empirical studies in ubiquitous and mobile computing;**

Additional Key Words and Phrases: Notification, interruptibility, intelligent system, smartphone, android, experience-sampling method

ACM Reference format:
Tianshi Li, Julia Katherine Haines, Miguel Flores Ruiz de Eguino, Jason I. Hong, and Jeffrey Nichols. 2022. Alert Now or Never: Understanding and Predicting Notification Preferences of Smartphone Users. *ACM Trans. Comput.-Hum. Interact.* 29, 5, Article 39 (November 2022), 33 pages.
https://doi.org/10.1145/3478868

1 INTRODUCTION

Notifications are an indispensable feature of today's mobile devices, allowing apps to convey relevant updates to users promptly. However, untimely notifications can increase cognitive load, negatively impact productivity, and cause frustration, stress, social pressure, and embarrassment [21, 22, 25, 41, 42, 46]. To mitigate these negative effects, past research has taken two approaches to design intelligent notification systems. One approach is to optimize for certain aspects

Authors' addresses: T. Li and J. I. Hong, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213; emails: {tianshil, jasonh}@cs.cmu.edu; J. K. Haines and M. F. R. de Eguino, Google, 1 Market St, Suite 400, San Francisco, CA 94105; emails: {juliahaines, miguelfrde}@google.com; J. Nichols, Google, 1600 Amphitheatre Pkwy, Mountain View, CA 94043; email: jeff@jeffreynichols.com.

1073-0516/2022/11-ART39
https://doi.org/10.1145/3478868

of user perceptions about notification delivery, such as interruptibility [13, 14, 33, 38, 48] and mental load [21, 35, 36]. A common notification handling technique is to defer the delivery of notifications to an opportune moment [1, 11, 20, 37, 41, 45]. An alternate approach is to emulate users' actions on notifications [19, 30, 40, 51]. Some work in this space attempts to automatically discard notifications that are similar to ones that users have frequently dismissed themselves [2, 30], and other work attempts to delay notification delivery in order to optimize for user responsiveness [3, 33].

Despite their practical value, these approaches are grounded in assumptions about how users want a system to deliver their notifications that have not been formally tested. Prior work has touched on some limitations of these approaches. For example, Horvitz et al. [20] discussed the risk of missing urgent notifications when deferring notifications solely based on users' interruptibility. This example suggests multiple factors affect users' notification preferences and that it will be necessary to integrate these factors together to produce a high-quality system for managing notifications. As another example, Mehrotra et al. [30] manually excluded reminder notifications from being automatically discarded because these notifications do not require any action even if they are considered useful. This example suggests that the rationales of users' actions on notifications can be complicated and having the system simply emulate users may not be an effective option.

To gain a more nuanced understanding of users' notification preferences, we took a user-centered approach and directly probed Android smartphone users' notification preferences via a one-week, mixed-methods study ($N = 35$). The study comprised two tasks that solicited users' subjective feedback on a cross-section of the notifications they received using the **experience-sampling method** (**ESM**) [17]. We also collected traces of user behavior related to phone usage and notification traffic in the background during the study. To study issues from previous research in more depth [20, 30], we conducted mixed-methods analyses to examine two questions:

RQ1. *What factors affect users' notification preferences and how do these factors affect notification preferences?*

RQ2. *What are the reasons that cause users to act upon notifications in a certain way and how do users' actions on their notifications correlate with their notification preferences?*

We also explore the feasibility of directly predicting users' notification delivery preferences based on their behavior. Previous research has highlighted the important role that individual difference plays in users' perceptions about notifications and suggested that it is crucial for intelligent notification systems to be able to automatically personalize to individual preferences [24, 38]. Unfortunately, soliciting users' subjective labels on the fly can incur a considerable cost, so researchers have instead suggested leveraging users' observable actions [34, 46]. Our study, which collected both user labels and behavioral data, provides us with an opportunity to explore the tradeoffs between these two approaches. In the context of our data, we provide insight into an answer for the following question:

RQ3. *How well can a predictor of users' notification delivery preferences be personalized using solely user actions as compared to using user labeled data?*

Our main contributions include

— An analysis identifying factors that affect users' notification delivery preferences, which helps develop an understanding of users' mental models for how an ideal notification manager might work. Specifically, we found that users mentioned content factors significantly more frequently than contextual factors when discussing notification preferences, and preferences vary substantially between different users.

— A systematic examination of the relationship between users' objective behavior and their subjective preferences. Our analysis improves understanding of how and why users interact with notifications, and reveals the intrinsic ambiguities about users' intentions when observing only their actions.
— A demonstration of how personalized models solely using user actions on similar notifications collected in the past can improve the performance of a generic model by 39% as measured by area under the ROC curve. In our dataset, this improvement is close to what can be achieved by a model personalized with labels solicited from the user (42%). Our experimental results suggest a method can dynamically adapt a notification preference prediction model to individual differences after deployment while causing minimal disruption to the user.

2 STUDY METHODOLOGY

We conducted a one-week mixed-methods study to understand how smartphone users might prefer an intelligent system to deliver their notifications and how user behaviors correlate with those preferences. As such, our study both solicited participants' subjective feedback on randomly sampled notifications and collected smartphone usage traces in the background over the entire course of the study, including all incoming notifications, participants' interaction with notifications, and phone usage. The study was operated by an app that participants installed onto their personal phones and data was synchronized from phones to a secured Firebase storage instance.

We first ran a pilot study with 30 participants in January 2019, and then ran a formal study with 40 participants in April 2019 (35 of which contributed complete data). Both studies followed similar study designs and lasted for one week, similar to past work [41, 54]. In this article, we primarily discuss the formal study and analyze data only from that study. We discuss the pilot study only in terms of how its results affected the design of the formal study.

2.1 Sample

We sought a diverse sample of participants during recruitment. In our formal study, 40 participants successfully completed the onboarding process and contributed valid data. Of these, 17 people self-identified as male, 22 as female, and 1 as non-binary gender. The youngest participants were in the 18–24 age group and the oldest participants were in the 65 or older age group. Most subjects were in the 35–54 age group (17 out of 40). The self-reported education level ranged from high school degree to doctoral degree with diverse occupations. More than half of the participants identified as a "heavy user" of their smartphone, defined as using their phone more than 6 hours per day.

The participants of both the pilot study and the formal study were recruited through Google's internal user study recruitment service. We created a screener to identify candidates, and then we selected and contacted the final participants ourselves. We required participants to be located in the US and be at least 18 years of age. We also required participants to have an Android phone as their primary phone, running at least Android Oreo, due to technical constraints. The participants were required to be single language (English) users so that all apps on the phone would mainly target English users and notifications would be written mainly in English. We also excluded users who had more than one phone or used a smartwatch in order to reduce potential interference of receiving notifications on multiple devices.

To encourage participants to stay in our study, which requires constant engagement and collects relatively invasive data, the compensation followed a tiered incentive structure. Participants who answered questionnaires for the study tasks in the first three days received $35 USD. Participants who answered questionnaires in the last four days received an additional $45 USD. There was a $20 USD bonus for participants who completed the entire 7-day study, for a maximum compensation of $100 USD. 47 participants signed up initially, though seven dropped out or did not complete

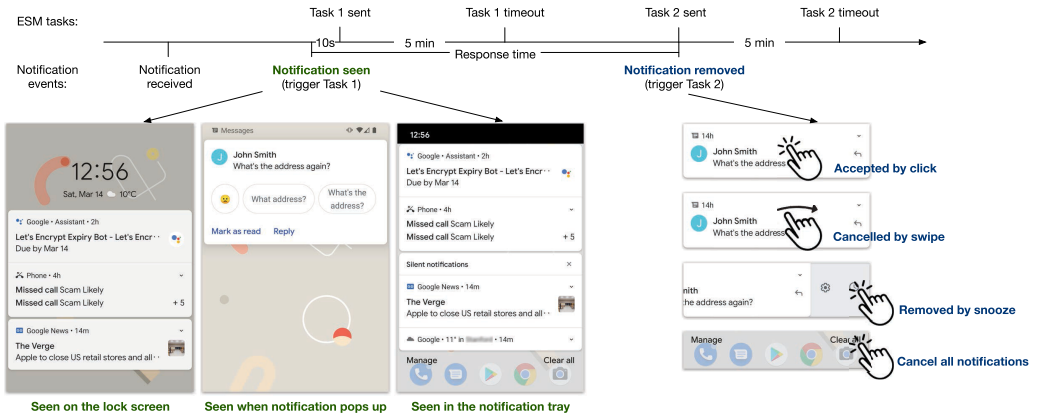


Fig. 1. This figure demonstrates a timeline of user-driven notification events and ESM tasks for one sampled notification. Task 1 asks users about their preferred way to deliver the notification, which is sent 10 seconds after the "notification seen" event is detected. This delay is to reduce the impact of the task on users' behavior after seeing the notification. Task 2 asks users about how promptly they took certain actions to remove the notification and why they took that action. This question is sent immediately after a user-driven notification removal event is detected. Both tasks have an expiration time window of five minutes, to give users flexibility in choosing the exact timing to answer the in-situ query.

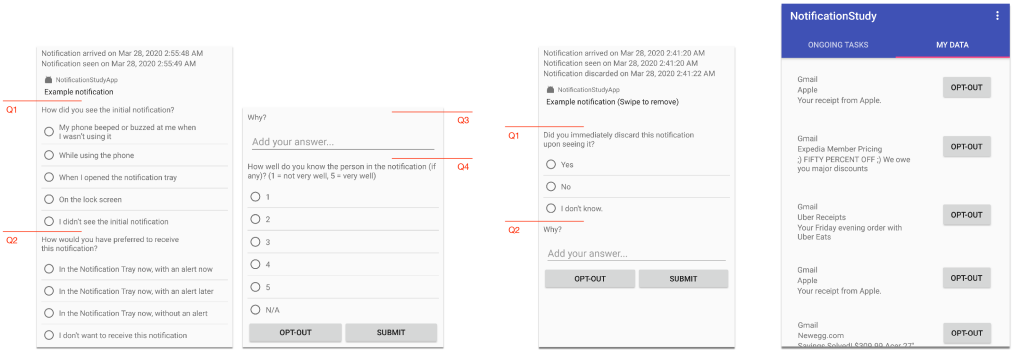
onboarding, and five did not respond to any questionnaires during the study. Phone usage and behavior data were still collected for the five subjects that did not drop out but did not respond to questionnaires. The structure and amount of the incentives for the study were determined by Google's user research compensation policy. This study was approved by Google's privacy working group and complied with all internal corporate policies for the collection and storage of user data.

2.2 Study App

Data collection for the study was conducted remotely via a custom Android app. The study app first guided participants through an onboarding process. This process instructed participants to grant permission to the notification listener service and the accessibility service so that the app could listen to all incoming notification traffic and log phone usage traces. Over the course of the study, participants responded to two types of ESM tasks within the app, which were triggered by notification events. These tasks, denoted Task 1 and Task 2, are described in more detail below (also see Figure 1). Throughout the study, the app also collected all notification content, how users handled notifications and phone usage traces. All collected data was synchronized to a secured Firebase instance that we operated specifically for this study.

2.2.1 ESM Task 1: Notification Delivery Survey. Task 1 was triggered when the notification was *first seen* by the participant. Figure 2(a) shows the screenshots of the survey user interface that was presented to subjects. All questions were required unless the subject chose "I didn't see the initial notification" for the first question. Task 1 was sent for a subset of notifications, the sampling rules for which are detailed in Section 2.2.3.

The most important questions of Task 1 are the second and third: "How would you have preferred to receive this notification?" and "Why?". The former asked the participant to choose one of four options as their preferred way for the system to deliver the notification. The latter requested a free-form explanation of question two. The four options cover different notification delivery



(a) Task 1 questionnaire (b) Task 2 questionnaire (c) Data overview and privacy controls

Fig. 2. Study app UI screenshots. Task 1 asks users about their preferred notification delivery method for the current notification. Task 2 asks users about how promptly and why they took certain actions (e.g., accepted/discarded) on the current notification. In the app, the "MY DATA" section shows all notifications that have been collected during the study. Participants are able to remove all traces of notification by clicking the "OPT-OUT" button on any of the screens.

techniques from the perspective of whether to send an alert and the timing of that alert (also see Figure 2(a))

— In the Notification Tray now, with an alert now
— In the Notification Tray now, with an alert later
— In the Notification Tray now, without an alert
— I don't want to receive this notification.

Because there are many delivery alternatives for an intelligent notification system, our main design consideration was to choose a few representative techniques that have been widely considered and could provide meaningful values to users.

The pilot study used a different version of the second question, which only focused on notification delivery timing and provided three choices for timing: "Now", "Later", or "Never". However, the dataset was highly skewed towards the "Now" option (72.7%), which showed that users either did not want to defer the delivery of notifications in most situations or did not understand the behavior of the "Later" option. In addition, we found in the pilot study that users were more sensitive about whether the notification was silently put in the notification tray than when it was put in the notification tray. Therefore, in the formal study, we redesigned the question to only provide two options for delivery (now or never) and three options for alert (now, later, never). This change helped us achieve a more balanced dataset and gain a better understanding of user preferences.

To detect the time at which a notification is seen, prior research [34] assumed all newly available notifications were seen when the screen was unlocked. Our study followed the same assumption but made a few changes to improve the detection accuracy. We used three heuristics to estimate if a notification had been seen in our study (also demonstrated in Figure 1):

— *Seen on the lock screen:* The first four notifications in the tray were marked as seen whenever the user turned the screen on. We used the Android "screen on" event rather than the "unlock" event to identify when notifications may have been seen on the lock screen, as notifications are visible when a user turns the screen on, even if they do unlock the phone.

— *Seen in the notification tray:* All notifications were marked as seen when the user pulled down the notification tray (except group notifications,[1] which hide the individual notification content they contain).
— *Seen in pop up:* A notification was marked as seen if it popped out from the edge of the screen when the phone was unlocked and in use.

As demonstrated in Figure 1, there was a 10-second delay between a detected notification seen event and the Task 1 notification. This gave users some time to consider the notification before realizing that the notification was sampled so that we might observe more natural behavior. Each Task 1 survey expired 5 minutes after it was sent, at which point it was automatically removed from the notification tray. This ensured users had some flexibility in choosing when to answer study surveys, while still ensuring that the surveys they did respond to would cover recent notifications.

2.2.2 ESM Task 2: Notification Action Survey. Task 2 was triggered when a sampled notification from Task 1 was *removed by the user*, as demonstrated in Figure 1. Figure 2(b) shows the question that was presented to the user afterwards. In Android O and later versions, the notification removal callback function `onNotificationRemoved` in `NotificationListenerService` receives a parameter `reason` which indicates how the notification was dismissed.[2] We analyzed all possible values of the `reason` parameter and derived six notification removal actions. There are four user-driven removal actions: click, cancel (swipe away), cancel all (by clicking the cancel all button in the notification tray), and snooze. There are also two app-driven removal actions: automatically dismiss and overwrite. The overwrite action occurs when the app sends a new notification with the same `id` in the `notify` function call.[3] Only user-driven removal actions can trigger a Task 2 survey.

The task questions first asked whether the subject immediately interacted with the notification upon seeing it, and then asked the subject to explain their choice in a free-form response. Both questions were required unless the subject chose "I don't know" for the first question.

Unlike Task 1, the Task 2 notification was immediately sent to the subject when the interaction with the notification was detected. This is because the task is triggered by an explicit user interaction that indicates that the user has made a decision about the notification, and there is no need to wait for additional reflection before presenting the survey. Similar to Task 1, each Task 2 survey expired after 5 minutes, at which point it was removed from the notification tray.

2.2.3 ESM Task Sampling Rules and Data Collection. Only a small portion of notifications was randomly selected for the ESM tasks to reduce the cost of interruption and fatigue during the study.

We first filtered out notifications that could not be removed by the user[4] and low-priority "bulletin" notifications, which persist in the notification tray for long periods of time. These bulletin notifications are often used to present some ongoing status information (e.g., current weather, current traffic, music playing now). We found certain apps that only create "bulletin" notifications via the pilot study, and we filtered all notifications from these apps during the formal study (see Appendix B for details).

Each subject was asked to complete ESM tasks for no more than seven notifications per day. In the first day of the study, we sampled one notification out of every ten notifications that were seen until the user had responded to seven notifications. On subsequent days, the sampling interval was

[1] https://developer.android.com/training/notify-user/group.
[2] https://developer.android.com/reference/android/service/notification/NotificationListenerService.
[3] https://developer.android.com/reference/android/app/NotificationManager.
[4] https://developer.android.com/reference/android/app/Notification.html#FLAG_ONGOING_EVENT.

recalculated to adapt to each user's phone usage pattern, using the formula $sampling\_interval = max(20, min(5, \frac{3 \cdot 60 \cdot 60 \cdot totalSeenNotificationCount}{currentTimestamp - startTimestamp}))$, in which $currentTimestamp$ and $startTimestamp$ are both unix timestamps. In this formula, $totalSeenNotificationCount$ represents the number of notifications that a particular user has received and seen during the study. The events we used as indicators of a user seeing a notification are summarized in Section 2.2.1.

Notification text content was stored for all notifications that were sampled for an ESM task, in addition to the data that was passively collected for all notifications. No images appearing in notifications were stored.

2.2.4 Passive Data Collection. During the study, the app logged notification data and phone usage data for all notifications passively in the background, which includes both notifications sampled and not sampled for ESM tasks. Passive notification data included the notification text and timestamps of the notification delivery, seen, and removal events. Phone usage data contained the timestamps of phone turned on/unlocked/locked events and app open events, which also contained app package names. We used the Accessibility event `TYPE_WINDOW_STATE_CHANGED` as a proxy for app open events, which is fired when a section of the user interface is visually changed.

Passive data collection allowed us to create a larger unlabeled dataset to complement the smaller, but richer, labeled dataset produced by the ESM tasks.

2.2.5 Participant Privacy Controls. Due to the sensitive and personal nature of notifications, we explicitly designed multiple privacy features to make sure participants had control over their data and could opt-out of the collection of any notification data at any time during the study. First, every ESM task featured an "OPT-OUT" button that would dismiss the survey and remove all information regarding this notification, both locally and remotely on our Firebase instance (see Figure 2(a) and (b)). Second, at any time participants could review all notifications that had been collected in the "MY DATA" section of the study app (see Figure 2). By clicking on the "OPT-OUT" button, the user can delete all data about this notification both locally and remotely.

We made these features clear in both the study instructions and the app's onboarding process. We also provided participants with a technical support e-mail address where they could address any concerns about privacy, but we never received any requests for data deletion nor questions about the privacy features. We did receive other types of technical support questions at this address.

2.2.6 Methods to Keep the Study App Running. Since there is only a background service of the study app running most of the time during the study, we used several methods to prevent the service from being killed by the system and to make sure the ESM tasks were properly sent to participants and the data was properly logged. First, we asked our participants to turn off the system battery saver feature during the onboarding process to prevent the Android system from killing background services for optimizing battery life. Second, our app can catch exceptions that cause the app to crash, and then restart the main activity automatically. Third, the app can continuously send updates of statistics such as the number of notifications received and sampled on the current day to our Firebase real-time database. This allowed us to monitor the app's running status of every participant during the study. We did not notice abnormal status during the study.

2.3 Study Procedure

Our study began on April 2nd, 2019 (Tuesday) and ended on April 9th, 2019 (Monday) for all but three participants. Those three started and ended two days later due to recruiting issues. Before the study, all participants signed a consent form. On the morning of the first day, we sent an e-mail to each participant which contained a link to the study app and study instructions. Once installed, the

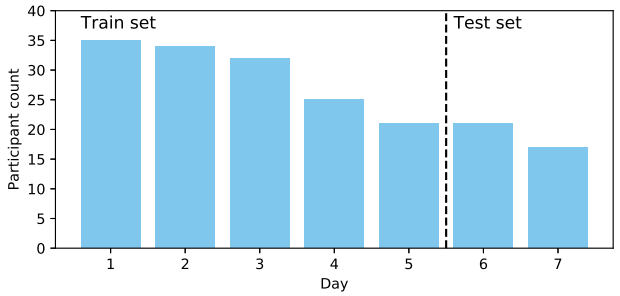


Fig. 3. The number of participants that responded to ESM tasks on each day. The considerable drop off rate suggests it is difficult to run this study with a longer duration and justifies the one-week-long study design.

study app guided users through an onboarding process that requested the necessary permissions to collect data, demonstrated the two ESM tasks, and described the privacy features of the study app.

After seven days, the app automatically stopped collecting data and posted a notification to inform users that the study had ended. Before participants removed the app, they were asked to initiate a manual upload from the app to ensure all data had been uploaded to our remote server.

Over the course of the week-long study, we received phone usage data from 40 participants in total, including how and when users handled notifications and when they opened what apps. Five of these participants only contributed passive data (see Section 2.2.4), while the other 35 participants completed some of the ESM tasks. For our analysis, we removed Task 1 responses that were marked as "not seen" and Task 2 responses with "I don't know" to the first question ("Did you immediately discard the notification upon seeing it?").

2.4 Train/Test Dataset Preparation for Coding (RQ1&2) and Notification Preference Prediction (RQ3)

Before we started the qualitative coding analysis and the experiments about predicting notification preferences, we split the entire dataset into two parts, based on notification delivery time. Notifications delivered in the first five days of the study were treated as the training set, and notifications delivered in the last two days were treated as the test set. The dataset was divided for all analyses, including both the evaluation of the statistical models and the qualitative coding, in order to avoid overfitting to the test set. Dividing the dataset based on time is a common practice in user behavior modeling research, which emulates the real-world situations where past data of the same user or different users is used to understand human behaviors and build predictive models.

Figure 3 shows the number of participants that responded to ESM tasks each day of the study (35 on the first day, 16 on the last day). Due to this dropoff in participation, our training set consists of data from 35 people, while the test set only consists of data from 20 people.

Table 1 presents the number of notifications that were collected (*All*), have a Task 1 response (*Task 1*), have a Task 2 response (*Task 2*), have either a Task 1 or Task 2 response (*Either task* or *All sampled*), and have both Task 1 and Task 2 responses (*Both tasks*), separated into the *Training set* and the *Test set*.

2.5 Qualitative Coding Analysis Process and Result Summary

We performed qualitative thematic analysis [4] on two types of textual data collected in the study: (1) explanations of delivery preference from Task 1, and (2) explanations of notification-related activity from Task 2.

Table 1. Data Collection Overview

Data category	Training set	Test set	Total
All	19,946	4,773	24,719
Task 1	525	102	627
Task 2	376	71	437
Either task (All sampled)	649	134	783
Both tasks	292	46	338

We collected a fairly large dataset regarding the number of notifications logged, while only a small portion was labeled by users, which resonates with the difficulty of collecting subjective labels and the importance of leveraging objective data in prior research [50].

2.5.1 Coding Process. The analysis consisted of two rounds of iterative coding. The first round was carried out on the pilot study data and the second was on the formal study data. In the first round, two researchers collectively conducted open coding on the pilot study data to develop codes for concepts in the text. Then one researcher conducted axial coding to group these codes into high-level categories. In the second round, the researchers collectively labeled the data gathered from the formal study, using the category framework that had been created. The goal of the ESM task analyses was to identify factors that may affect users' notification delivery preferences and user behavior after receiving notifications, so task responses were assigned multiple codes if necessary.

2.5.2 Coding Result Summary. We first derived 40 codes for Task 1 responses and 27 codes for Task 2 responses. We then merged similar codes that appeared less frequently, and finally achieved 32 codes[5] corresponding to 11 factors from 2 high-level categories for Task 1 responses and 13 codes[6] from 4 sub-categories and 2 high-level categories of user action rationales for Task 2 responses. The coding results for these three types of data are presented in Sections 3.2 (notification preference factors) and 4.1 (reasons about user behavior after receiving notifications).

2.5.3 Inter-rater Reliability. To validate our developed codes, another researcher (different from the two researchers who developed the codes) applied the developed coding schemes to a subset of the data. The new coder first discussed the definitions of the codes with one of the two researchers who developed the codes and practiced the two coding tasks using a different random sample of 20 responses for both tasks. Then the new coder independently coded a random sample of 100 responses for each of the two tasks respectively. With the original labels from the first two researchers and the new set of labels from the third researcher for the 100 responses, we calculated the Krippendorff's alpha, which is a standard inter-rater reliability measure for codes that are not mutually exclusive [16]. Krippendorff's alpha was 0.82 for Task 1 labeling results and 0.85 for Task 2 labeling results, which are both above the common standard of reliable labeling results [16].

3 RQ1 RESULTS: UNDERSTANDING USERS' DELIVERY PREFERENCES AND RELATED FACTORS

In this section, we present results exploring our first research question about the factors affecting users' notification preferences. We begin with a notification delivery preference breakdown for all

[5]A code "no meaningful answer" is included for user responses that are too short or do not provide meaningful explanations of their choices. For most factors in Table 2 that contain multiple levels, each level corresponds to one code and the factor itself does not count as a code separately. However, we have two separate codes for the "Notification type" and "People mentioned" factors because the levels listed for the two factors are not exhaustive.

[6]The 13 codes consist of the 12 codes in the "Codes" column of Table 3 and a "no meaningful answer" code for user responses that are too short or do not provide meaningful explanations of their choices.

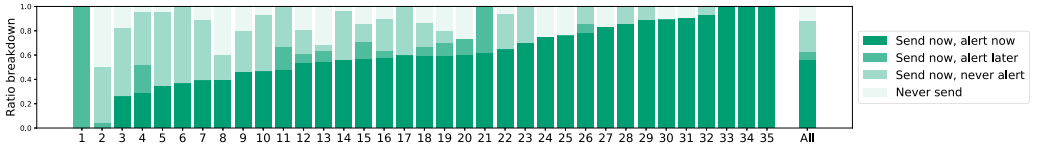


Fig. 4. Per-participant and overall preference distribution (sorted by the ratio of "send now, alert now" in ascending order). This figure suggests there is a strong effect of individual difference in notification delivery preferences.

subjects together and each individual (Section 3.1). Then we dive into an analysis of user responses to Task 1, which probes the underlying reasons for selecting particular preferences (Section 3.2). All analyses in this section are grounded in the Task 1 training set (525 datapoints, see Table 1).

3.1 Notification Delivery Preference Breakdown for all and Individual Subjects

In our sample, 56% notifications were expected to be delivered with an alert immediately, indicating a potential to improve user experience by reducing interruption. The distribution of the other three choices demonstrates users' preferences of methods for reducing interruptions. The ratios of "Send now, never alert" (26%) and "never send" (12%) were higher than "send now, alert later" (6.3%).

We conducted pairwise comparisons of the ratios of the four delivery methods and found significant difference for all the six pairs under McNemar's test (p-values have been adjusted using Bonferroni correction): alert now vs. alert later: $\chi^2(1, N = 525) = 147$, $p < .001$; alert now vs. never alert: $\chi^2(1, N = 525) = 24$, $p < .001$; alert now vs. never send: $\chi^2(1, N = 525) = 98$, $p < .001$; alert later vs. never alert: $\chi^2(1, N = 525) = 199$, $p < .001$; alert later vs. never send: $\chi^2(1, N = 525) = 334$, $p < .001$; never alert vs. never send: $\chi^2(1, N = 525) = 236$, $p < .001$.

Figure 4 shows that our subjects are evenly distributed across the spectrum of the tolerance to interruption caused by notifications. Two subjects (user1 and user2) did not want any of their notifications to interrupt them immediately, while three others (user33, user34, and user35) preferred to receive immediate alerts for every notification. A total of 13 out of 35 participants (user1, 2, 4, 11, 12, 13, 15, 16, 17, 18, 20, 21, 26) chose "send now, alert later" for some of their notifications, while others did not select this option at all. This suggests that individual differences affect people's attitudes towards deferring the alert.

3.2 Factors Affecting Notification Delivery Preferences

Based on the qualitative coding of Task 1 user responses, we generated a list of factors (Table 2) that influenced users' notification delivery preferences. We categorize the factors into two categories: *content* and *context*. Factors related to the notification itself are considered *content* factors. These factors are mostly intrinsic characteristics of notifications, varying from objective factors such as *notification type* to subjective factors such as users' perceived notification *importance*. Factors related to the user's status (e.g., *interruptibility*) or the environment (e.g., *location*) are considered *context* factors. Figure 5 shows the breakdown of delivery preferences for some content and contextual factors with relatively high occurrence.

Although both content characteristics and contextual factors emerged, content-related factors (91.6% of notifications) were mentioned much more frequently than contextual factors (15.3%). The ratio of content-related factors is significantly higher than context-related factors under McNemar's test ($\chi^2(1, N = 464) = 8.7$, $p = .003$).

In the following, we present the definitions, examples and the relationship of each factor to notification delivery preferences (Figure 5).

Table 2. Factors That Affect How and When People Want to Receive Notifications

Categories	Factors	Levels
Content (425)	Importance (212)	Valuable (112) Good to know (47) Irrelevant (53)
	Notification type (196)	Text message (46), Ongoing conversation (30), Facebook (16), Reminder (14), E-mail (13), App update (9), Battery (6), Deal (7), Weather (5), News (6), Two-factor authentication (5), Work-related (3), etc.
	Time-sensitivity (78)	Time-sensitive (17) not time-sensitive (61)
	People mentioned (55)	Friends (19), Families (16), Unimportant persons (8), etc.
	Notification expecting action (31)	
	User-driven notification (29)	
Context (71)	Interruptibility (46)	Busy, improper timing (34) Convenient, proper timing (12)
	Phone/app usage (27)	Using the phone/app (17) Not using the phone/app (10)
	Time (5)	
	Activity (4)	
	Location (3)	

The number of notifications receiving each code are specified in parentheses. Note that each notification may have more than one code.

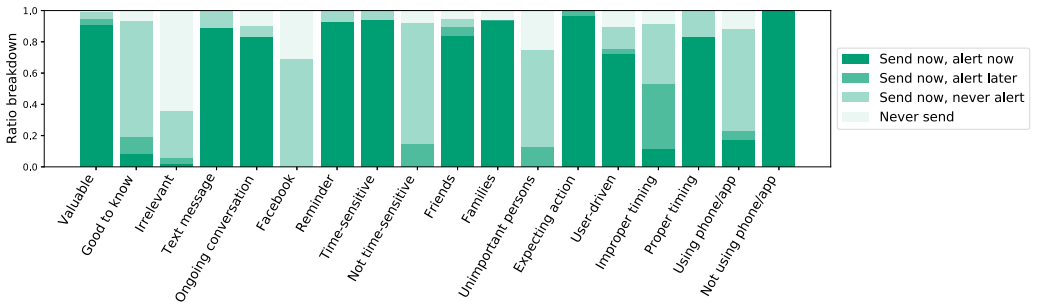


Fig. 5. User delivery preference breakdown for content and contextual factors with relatively high occurrence.

3.2.1 Importance. We noticed that users frequently referred to the perceived importance of a notification as their reason for selecting a certain delivery option. Three levels of perceived importance of notifications emerged from participant responses, which are *"Valuable"* notifications that contain useful and important information (example quotes: "important bank account info", "needed the info"), "Good to know" notifications that contain information that might be interesting but not as important (example quotes: "this was fine", "not very important"), and "Irrelevant" notifications that are totally irrelevant and sometimes even annoying (example quotes: "don't need to know when apps have updated", "because it's a scam").

Figure 5 shows a high percentage of "send now, never alert" in the "Good to know" category and even in the "irrelevant" category, which indicates that users are more inclined to balance the reduction of interruption and the risk of information loss by suppressing the alert and keeping these unimportant notifications for possible future examination.

3.2.2 Notification Type. Some responses attributed their choices to a general like or dislike about a certain group of notifications. We noticed that there were a variety of granularities of notification groups that users would like to manage in the same way. Many of these groups are at

app level. For example, User9 explained the reason for choosing "never send" for a Facebook notification as "Facebook notifications aren't so important. I'll get to them when I look at Facebook." However, we also saw responses that specified preferences at a higher level about characteristics shared across multiple apps (e.g., text message, news, deal) or at a lower level about specific app usage. For example, User18 said that "I prefer not to receive audible notifications for non personally related items coming in to my phone.", which generalizes her dislike about audible notifications to non-personally related notifications that could come from multiple apps. On the contrary, User11 said that "I like to be aware of comments in a current conversation I'm having with someone on social media", which describes a more specific situation that requires a deep understanding of the app usage context to recognize. Overall, our findings suggest that it is important to allow users to manage notifications in ways beyond app-level and channel-level filtering that are available in current systems.

3.2.3 Time-sensitivity. Time-sensitivity is a common reason that emerged from users' responses. We noticed that users often considered all notifications of a particular type to have the same time-sensitivity. For example, time-sensitive notification types include the following:

— text messages ("I want to answer this one as soon as possible so it doesn't expire."—User15)
— reminders ("it's a reminder to eat"—User14)
— sales alerts ("when there are ebay items which i am watching and may want to purchase, they are time sensitive. so i need to know right away"—User18)
— low battery warnings ("so I can charge my phone before it dies"—User31)
— work-related notifications ("work related and time sensitive"—User11).

Some users considered other notification types to not be time-sensitive:

— social media app ("Facebook is NEVER urgent. It can wait until I'm bored enough to check FB."—User9)
— text messages ("Instant messages are rarely as urgent as e-mails or other app notifications. They can wait until the next time I check my phone."—User9)
— e-mails ("e-mails usually aren't urgent so i don't need an alert"—User16)
— app updates ("Update stuff that happens while I sleep. It's good to see in the morning when I get up, but doesn't need an alert.").

Although there seems to be a relationship between notification type and sensitivity, this relationship may vary from person to person, as demonstrated in User9's and User14's opposite attitudes towards text messages.This relationship may also vary from time to time as explained by User5 regarding a LinkedIn notification: "I'm not actively job hunting and do not plan to until I'm closer to graduating. I have started some networking and do let it give me alerts related to that, but they're not urgent. It would probably be a different story if I was actively seeking employment though."

3.2.4 People Mentioned. Users sometimes referred to the person mentioned in the notifications to help explain their notification preferences. Three types of personal relationships emerged in users' responses, which are friends, families, and unknown/unimportant persons. Figure 5 suggests notifications that involve friends and families often need an immediate alert. This is consistent with findings of previous research [34].

3.2.5 User-driven Notification. We define notifications that are directly triggered by user actions as user-driven notifications, and different types of user-driven notifications were associated with different delivery preferences in our users' responses. The first type is reminders set by the user. Users almost always wanted an immediate alert for these notifications because this is what

they are designed for (e.g., "it's an alarm, it's supposed to alert me"—User6). The second type is two-factor authentication requests and users wanted an immediate alert for them as well for similar reasons.

On the contrary, notifications that acknowledge a user action are another type of user-driven notification for which users had varying preferences. Some notifications were considered redundant and should be discarded automatically, such as a "download complete" notification (e.g., "I don't really need an alert for a download, I initiated it and I know it's happening"—User6). However, some notifications were considered useful for keeping users aware of abnormal situations, such as an unauthorized charge or a suspicious log-in attempt, and therefore were preferred to cause an immediate alert (e.g., "I was paying my credit card. This was the e-mail acknowledging payment. Some bank e-mails are advertisements, but if they contain the words "payment posted" or "payment scheduled" I would prefer to hear about those immediately. Also any that suspect fraud!"—User5).

3.2.6 Notification Expecting Action. We classified responses that mentioned that the notifications expect them to take certain action into this category. This action can both be digital (e.g., responding to a message) or physical (e.g., charging the phone). Figure 5 shows that almost all responses that fall in this category are about notifications that required an immediate alert.

3.2.7 Interruptibility. Interruptibility is the most frequently mentioned contextual factor in our user responses. We consider responses that mentioned notifications being sent at a proper/improper timing falling into this category. The responses that indicated proper timings were similar and mostly straightforward, containing keywords such as *"not busy"* and *"convenient"*.

However, we observed that two types of responses about improper timing appear frequently in our sample. The first type is related to situations when it is inconvenient or disruptive for users to check and react to the notification. For example, User12 chose "send now, never alert" for a Facebook messenger chat notification and explained, "so not to wake me up". The second type is related to situations when users are not interested in the content at the moment. For example, User7 chose "send now, never alert" for a sports news notification and said "I'm just not in the mood for fantasy sports right now. I'd rather have a nap." These two types correspond to research about interruptibility prediction based on external contexts [33, 36, 40] such as location and time of day, and internal states such as engagement with the current task [39] and attentiveness [9]. In addition, responses in the "improper timing" category had the highest percentage of choices of "send now, alert later", which suggests deferring the alert is more preferred in this situation.

3.2.8 Phone Usage. Users also frequently mentioned whether they were actively using the phone or the app that issued the notification as a reason for their delivery preferences. Figure 5 shows that users tended to prefer an immediate alert if the notification was sent to them when they were not actively using the phone or the app and dislike an immediate alert if the notification was sent when they were using the phone or the app. This was because they felt an alert was not necessary if they were already using the phone. For example, User9 selected "send now, never alert" for a discord chat notification and explained that "I was looking at the phone, so the sound wasn't needed."

3.2.9 Location, Time, and Activity. When tagging some data points with the factor "Interruptibility", we were focused on expressions in users' responses indicating their perceptions of how busy/available they were, while the factors "Location", "Time", and "Activity" were only used when users mentioned specific locations, times, and activities in their responses. The three factors sometimes co-occurred with Interruptibility , alluding to what caused different levels of interruptibility. For example, User11 implied why it was an improper timing to receive a news notification by *referring to his location*: "will read later, leaving gym—no time". Also, User11

Table 3. Reasons That Explain What Actions People Took to Remove Notifications and How Soon They Took the Action After Seeing it

Categories	Sub-categories	Codes
Direct reasons (199)	Reason about action type (174)	Read and no further action needed (54)
		Want to take action via the notification (91)
		Want to take action but not via the notification (22)
		Clear notifications (11)
	Reason about action delay (57)	Immediate action needed (14)
		Delayed by other tasks (20)
		Intentionally delay the action (23)
Indirect reasons (209)	Notification-related (178)	Useful (41)
		Not needed (121)
		Seen elsewhere (17)
	Availability (37)	Proper timing (4)
		Improper timing (33)

The number of notifications for each factor is specified within the parentheses. Note that each notification may have multiple codes.

implied why it was an improper timing to alert for an e-mail by *referring to a specific time*: "daily work e-mail—will either open Monday or delete".

Overall, these factors were less frequently mentioned than other contextual factors. Among the 10 notifications that were assigned one of the three labels, 4 came from User11, which suggests that individual difference may play an important role here.

4 RQ2 RESULTS: RELATIONSHIPS BETWEEN USER BEHAVIOR AND DELIVERY PREFERENCES

In this section, we examine the relationship between user behavior collected passively during the study, such as subjects' actual actions with their notifications, and subjects' responses to the Task 1 and 2 questionnaires. Our examination is driven by the two sub-questions of RQ2:

— What are the reasons that cause users to act upon notifications in a certain way? (Section 4.1)
— How do users' actions on their notifications correlate with their notification preferences? (Section 4.2).

Analyses in this section are either grounded in the "Both tasks" training set when analyzing the user behavior together with preferences (292 datapoints). or the "Task 2" training set when coding users' explanations for their behavior (376 datapoints). See Table 1 for more information about each data split.

4.1 Qualitative Analysis of the Rationales About How Users Handle Notifications

Using the results of the qualitative coding analysis of users' explanations as collected in Task 2, we generated a list of reasons that caused users to act upon notifications in a certain way (Table 3). Figure 6 plots the breakdown of delivery preferences, removal action types, and self-reported promptness for each of the 11 codes. We will use this to help unravel the relationship between subjective preferences and objective behaviors through grouping the 11 codes into four sub-categories and two categories. Overall, our analysis provides an in-depth understanding of the decision process that users may go through when handling notifications. We learned that although users may take similar actions on two notifications on the surface, the underlying reason and their preferences about how the notification should be delivered can be completely different.

4.1.1 Direct Reasons About Action Types. The sub-category "Reason about action type" includes four codes that help explain why users chose to take different actions on notifications (e.g., click, cancel, cancel all).

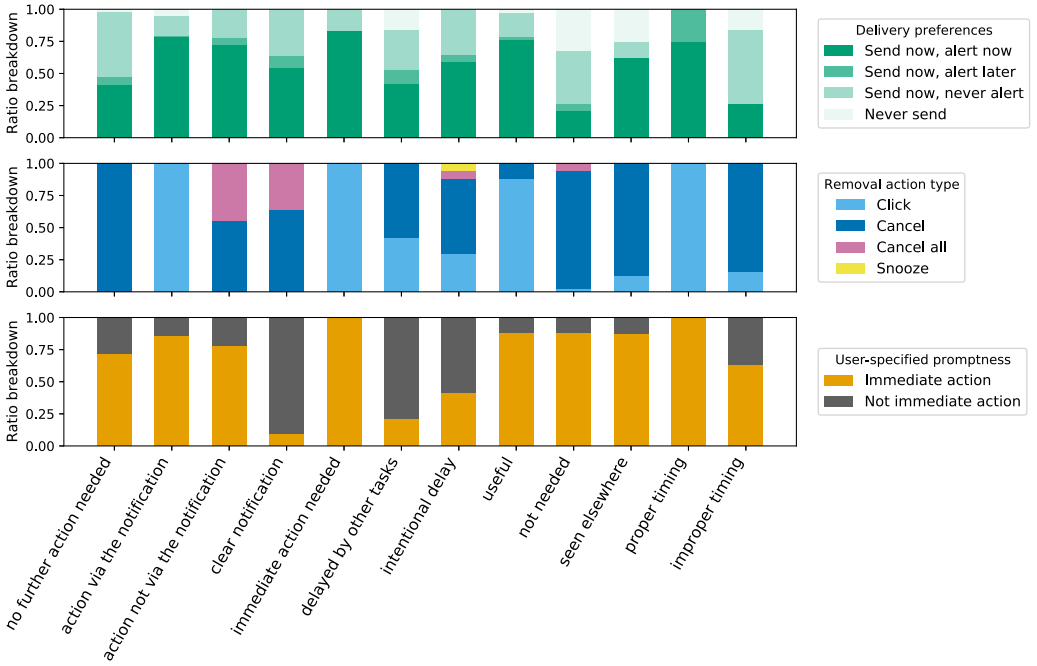


Fig. 6. This figure shows a breakdown of user notification delivery preferences, the user-driven removal action types, and the answers to whether the notification was handled immediately, for different factors affecting how people handle a certain notification. (calculated on the "Both tasks" training set) We can see that the same user behavior can be caused by different intentions with opposite delivery preferences (e.g., "not needed" vs. "action not via the notification").

For the first code ("Read and no further action needed"), we identified three common scenarios which seem to correspond to different preferences.

— Sometimes the notification content contains enough information for the user so they did not need to read it further (e.g., "I read the headline." —User20, about a news notification).
— The notification may notify the user of new information and require no further action (e.g., "I didn't need to do anything with it, it's just a notification of a download"—User6, about a download complete notice).
— The notification might suggest a further action to take, but the user decided not to act. For example, User9 cancelled a notification about a deal about the item on their wish list from Amazon (shopping app), because "I read the notification first, then took a minute to decide if I could make a purchase today or not. It's not really in the budget, so then I dismissed the notification."

The second and the third codes are both related to situations where the user wanted to take further actions in response to the notification. However, the actions were not always directly taken by clicking the notification. As demonstrated in Figure 6, the ratios of the choice of "Send now, alert now" are high (78.3% and 72.2%, respectively) for both situations, while the actions users took were very different. All notifications in the "action via notification" group were removed by clicking while all notifications in the "action not via the notification" group were removed via CANCEL or CANCEL_ALL actions, often during a notification clean up activity. For example, User5 cancelled a notification that reminded him to see photos from last year and explained that "It lets me see

pictures I took a year ago that have been uploaded to Amazon. I like seeing this notice, but I went through the app instead of the notice, so it didn't go away. That's why I manually deleted it."

Similarly, notifications given the last code "Clear notifications" were also removed by CANCEL and CANCEL_ALL actions. The difference in this case is that the CANCEL/CANCEL_ALL actions were not taken immediately when seeing the notification. This is because many notifications in this group also belong to the "Read and no further action needed" group, but instead of cancelling after reading, the user left the notifications in the tray and removed them later, often en masse. For example, User16 did not immediately cancel an Instagram notification about someone accepting his follow request, and said "I just left it there because it required no action." in his free form answer.

4.1.2 Direct Reasons About Action Delay. This sub-category includes three codes describing reasons about how promptly users handled notifications, which are "immediate action needed", "delayed by other tasks", and "users intentionally delayed the action". Here we provide more explanation with statistics and examples.

For the "Immediate action needed" code, Figure 6 shows that 83.3% of notifications were expected to be delivered with an immediate alert, all notifications were removed by click, and all actions were taken immediately after seeing the notification according to users' self-reports.

The "Delayed by other tasks" code refers to situations when users deferred their action because they wanted to finish their current task first. For example, User5 dismissed an e-mail notification after some delay and explained that "(I) was just being lazy. Was doing other things and didn't want to stop right then. Wasn't an important e-mail, so when I dismissed it didn't really matter".

The "Intentionally delay the action" code shows a different reason for delaying action because it is not urgent or they have plans for other activities soon. For example, User18 dismissed an eBay promotional offer notification and later explained "I was working and I knew this was not an issue I needed to deal with immediately." Although it also mentioned the working status before receiving the notification, the key reason for deferring the action seems to be related to the lack of urgency. As compared to the previous code, this code does not necessarily involve the current task and usually means a longer delay. It reflects a situation when the user is first informed about something and then actively integrate it into their schedule for the day. As a result, they may keep the notification as a reminder. For example, "It didn't require prompt attention. I left it in the tray as a reminder to check out the story later"—User7, for a news alert.

4.1.3 Indirect Reasons: Notification-related Characteristics and User Availability. The last two sub-categories happen to match the content and contextual factors discussed in our analysis of first task responses. All but one of the codes ("Seen elsewhere") also appear in the factors that we found affect delivery preferences.

Our analysis found that the "seen elsewhere" code occurs because notifications are often synchronized across multiple devices. For example, User11 mentioned that "I am also using my laptop to have the conversation in addition to my phone" when talking about a Facebook notification. In addition, sometimes there are duplicate notifications about the same information, such as duplicate e-mail notifications from two mail client apps (User4) and duplicate package delivery notices delivered by text message and from an app (User25). Figure 6 shows that subjects still preferred that 62.5% of the notifications in this category be delivered with an immediate alert.

4.2 Quantitative Analysis of the Relationship Between Notification Behaviors and Delivery Preferences

Response time [3, 33, 34, 38] and the action the user took to remove a notification [19, 30, 40, 51] are two common objective metrics that prior notification management algorithms have optimized for or used as a proxy to label the desirability or importance of notifications. In particular, prior

work often associates shorter response time and click-on-notification actions with positive attitudes towards the notification. Therefore, we want to quantitatively study how these two variables correlate with our self-reported notification delivery preferences.

4.2.1 Statistical Test Results. We first conducted a one-way ANOVA test to examine the relationship between response time and delivery preference. Our results showed no significant difference in response time among the four notification preferences ($F(3, 326) = 0.375, p = .771$).

We then conducted a Chi-squared test to examine the relationship between action type and delivery preference. Our results showed that the distribution of action types were significantly different among the four notification preferences ($\chi^2(12, N = 521) = 69.3, p < .001$).

4.2.2 Statistical Modeling Experiments. Next, we conducted statistical modeling experiments using the response time and removal action features to further study the feasibility of using user actions on notifications to label their subjective preferences. Our task aims at predicting whether to deliver the notification with an immediate alert using these features alone. All classifiers were trained on the Task 1 train dataset and evaluated on the Task 1 test dataset.

We first examined the relationship between response time and delivery preference by training a logistic regression classifier with response time as the only feature. The accuracy on the test set is 56.9%, which is equal to the chance accuracy that can be produced by always predicting the most popular option "alert now".

We then examined the relationship between action types and delivery preference. We created a seven-dimension one hot vector as the feature vector, with each dimension representing an action type. The accuracy of a logistic regression classifier trained with these features was 58.8%, which is only slightly better than chance accuracy. Even when we simplify the problem by only focusing on notifications removed by CLICK or CANCEL actions (61.1% of the entire set), which are the two major user-driven notification actions, the accuracy can only be slightly improved to 59.6%.

5 RQ3 RESULTS: PREDICTING DELIVERY PREFERENCES BASED ON USER LABELS/ACTIONS

In this section, we present the results about the last research question "How well can a predictor of users' notification delivery preferences be personalized using solely user actions as compared to using user labeled data?". Since the "send now, alert now" option was selected a lot more frequently than the other three options in our sample, we merged the three options into one group and defined a new binary prediction task. Notifications in the first class are those labeled "send now, alert now" in our dataset (relabeled "alert now" for this section, 56% of the test set), and the second class is made up of all other notifications (labeled "not alert now" in this section, 44% of the test set).

Our experiments included three models: a generic model (baseline) and two personalized models based on user labels of delivery preferences for past notifications and user actions on past notifications respectively. Figure 7 demonstrates the intuition behind how the three models make predictions about users' notification delivery preferences about an incoming notification. It also highlights the different requirements of data labels and the different results of the three models. Our results showed that by aggregating past notifications of similar content, a model of notification delivery preferences personalized on user actions could achieve similar performance as a model personalized on user-labeled preferences (i.e., ground truths). Both personalized models achieved a substantial improvement compared to a generic model built on other users' labels.

5.1 Baseline: Generic Classifier Based on User Labels (Model I)

Our baseline is a generic model that can be used in this scenario: the notification management system relies on a pre-trained, static classifier to predict users' preferences of a new incoming

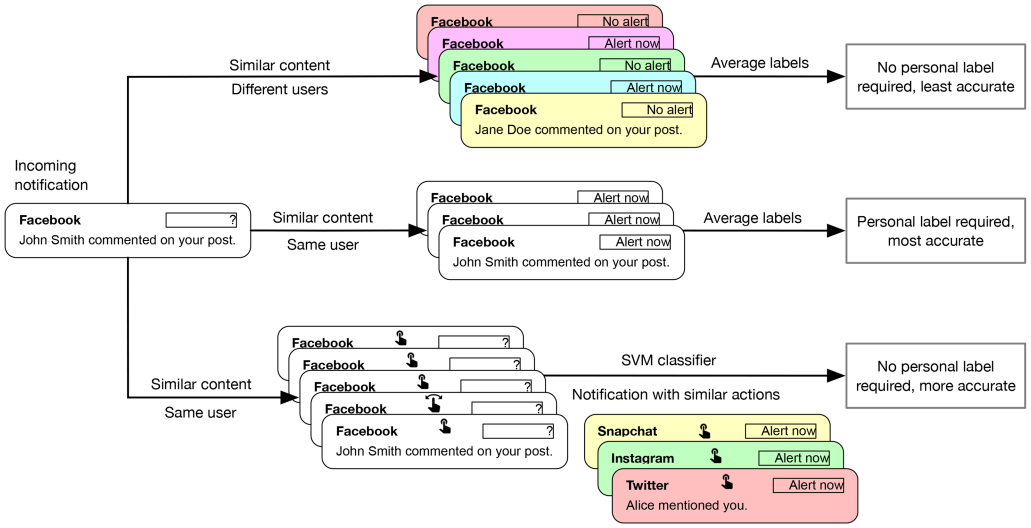


Fig. 7. This figure presents the intuition behind the three classifiers of delivery preferences tested in our experiments, which are a generic classifier (the first row), a classifier personalized on user subjective labels (the second row), and a classifier personalized on user objective actions on notifications (the third row). The rounded rectangles with different colors represent notifications from different users. The smaller rectangle on the top right corner of a notification indicates the user-labeled delivery preference (e.g., Alert now) and the question mark means this notification does not have a label of delivery preference. The different hand icons represent different actions that have been taken to remove past notifications (e.g., click, swipe away). The three leftmost rectangles present the requirements of personalized labels for training the classifier and the performance of the classifier.

notification. The key idea of the predicting algorithm is to find notifications with similar content from the Task 1 training set (525 notifications, from the first five days), and use the ratio of the "alert now" and "not alert now" labels previously specified for these similar notifications to predict the preference for notifications from the Task 1 test set (102 notifications, from the last two days, see Table 1 and Section 2.4 for more detail about how we split the data).

We define similarity as notifications issued by the same app, because of the universal availability of the source app feature in our dataset and the relatively small training set (525 instances). A larger labeled dataset would allow us to relax this definition and also consider other features.

During the classification process, the model uses the preferences of similar notifications labeled by other users in the training set to calculate the preference for a single user's notification in the test set. The algorithm compares the ratio of "Send now, alert now" among all matched notifications with a predefined threshold, and predicts "Send now, alert now" if the ratio is higher than the threshold. We experimented with different thresholds in the evaluation results sub-section.

5.2 User-Label Personalized Model (Model II)

Similarly, the second model also only uses subjective preferences labeled by users. This setting is based on a different fictional use scenario: the system requests the user label their preferred delivery method after receiving a notification, and when a new notification arrives, the model combines other users' data and the labels provided by the same user to predict the preferred delivery method of the new notification.

The classification algorithm works similarly to the algorithm in the baseline model, except that the algorithm uses similar notifications labeled by the same user in the training set when

available, and otherwise falls back to preferences labeled by other users, as in the baseline model.

5.3 User-Behavior Personalized Model (Model III)

In this model, we explore personalized prediction of notification preferences when users' subjective labels are not available for previous notifications. This is an important problem since soliciting subjective preference labels can impose extra burdens on users and is therefore unrealistic for real-world deployment. Following the analysis in Section 4, we studied predicting delivery preferences based on how users handled similar notifications in the past.

5.3.1 Creating the User Behavior Feature Vector. Similar to the statistical modeling experiments in Section 4.2.2, we take both removal action types and response times into account when constructing user behavior feature vectors. To examine the effect of these two types of behavior metrics, we built three versions of behavior feature vectors, which contains information related to: (1) removal action types (7-dimensional one hot vector where each dimension represents one action), (2) response times (1-dimensional vector which response time converted to a logarithmic scale, due to the long-tail distribution (Figure 10), using the following formula: $Time_{log\_scale} = \log_e(Time_{normalized} + 1)$), and (3) both action types and response times (8-dimensional vector that concatenates the previous two feature vectors). The three versions of feature vectors will correspond to three variants of the user-behavior personalized model, which are all evaluated in the next sub-section.

5.3.2 Delivery Preference Prediction Algorithm Design. Similar to Model I and II, our algorithm consists of two parts: for each notification, it first looks up similar notifications that the same user received in the past and traces the past user behavior after receiving those notifications, and then calculates a score for predicting delivery preferences based on a synthesized feature vector created by averaging feature vectors of these similar notifications. We detail the two parts below.

For the first step, our notification search scope can be expanded to the entire training set, which is 38 times larger than the labeled dataset used for finding similar notifications in Model I and II (see the "All" training set in Table 1). This is because reference notifications need not have been labeled by users. This enables us to use a more fine-grained definition of similarity for notifications. We now define similar notifications as those not only issued from the same app but also sent from the same contact or created with the same template (e.g., "[User ID] posted for the first time in a while. Be the first to add a comment.").

We created a notification content string for each notification, which for most apps was built by concatenating the notification title and the notification body. For messaging apps, which we manually identified, we instead create the content string using only the notification title. This is because the title is often more structured and contains the sender's name, whereas the body is free form text and may interfere with the template detection. We also removed all stop words from notification content strings to avoid false-positive matching.

We then iterate through past notification records of the same user and run a dynamic programming algorithm to find the longest common subsequence (not necessarily continuous, LCS) [18] between the notification content strings of the target notification and each candidate. After iterating through the entire list, we obtain a final set of similar notifications which have the maximum LCS with the target notification.

For the second step, we first calculate the behavior feature vectors of the final set of similar notifications, and then average them to obtain a new feature vector representing aggregated behaviors. We then use an SVM regressor to predict a score based on the new vector. In order to explore the best performance we can achieve without requiring the user to label any data

Table 4. Area Under the Curve (AUC) of the ROC Curves of Main Models Examined in Our Experiments

Exp. ID	Options	AUC
I	Generic model	0.539
II	User-label personalized model	0.766
III	Behavior personalized model (using action type)	0.737
III	Behavior personalized model (using response time info)	0.594
III	Behavior personalized model (using action type + response time)	0.749

beforehand, we trained different SVM regressors for different users, and only used *other* users' data to train the regressor. Specifically, each SVM regressor was trained on the "Task 1" training set excluding the same user's data; and during the training process, each synthesized feature vector was created based on similar notifications from the "All" training set, also excluding the same user's data. Our implementation uses scikit-learn SVR under the default configurations. We want to note that this SVM regressor does not use the user's action on the incoming notification as input, which ensures that our model is fully predictive.

A threshold needs to be specified before the actual classification process, and all notifications that have a score above the threshold will be classified to "Send now, alert now". We experimented with different thresholds in the evaluation results sub-section.

5.4 Evaluation Results

We evaluated the three models on the "Task 1" test set and present the results below. Since our models rely on finding similar notifications from the same package to make predictions, only 79 out of the 102 notifications in the test set was used in the evaluation. We selected the 79 notifications based on the coverage of the generic model, which is a subset of the two other personalized models (92 and 96 notifications for the user-label and user-behavior personalized model respectively).

Because the direct output of all three models is a score, we thoroughly enumerated different threshold values for each model to convert the score to a prediction. This allows us to study the tradeoff between the true prediction rate of alert now instances (recall) and the true prediction of other instances (specificity) by plotting **Receiver Operating Characteristic** (**ROC**) curves and calculating the **Area Under the Curve** (**AUC**) for each algorithm. This is among the most commonly used performance measurements for classification problems that require predetermined thresholds. AUC varies from 0 to 1. The AUC of an uninformative classifier is 0.5. For AUC above 0.5, a higher AUC indicates the model is better at differentiating between classes [5].

Table 4 shows the AUC of different models examined in our experiments. We can see that the personalized models have much better predictive powers than the generic model. Figure 8 plots the ROC curves. Note that we only plot the user-behavior personalized model built with the removal action type and the response time, which achieves the best performance among the three behavior-based personalized models we tested. The best performance of the two personalized models are very close, which improve the performance of the generic model by 42% and 39%, respectively.

6 DISCUSSION

Our work offers a new perspective on the study of notifications. Instead of focusing on particular factors that might affect users' perceptions of notifications, we took a user-centered approach to understand how and when a system should deliver notifications. Below, we summarize the lessons we learned from our study regarding users' notification delivery preferences, discuss how they may

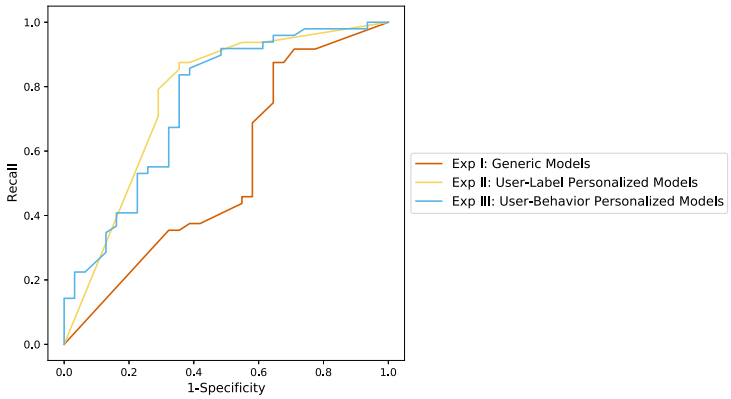


Fig. 8. The ROC curves of our generic and personalized prediction algorithms. Higher the AUC, better the model is at differentiating preferences. Our results are $AUC_{generic} = 0.539$, $AUC_{user\_label} = 0.766$, and $AUC_{user\_behavior} = 0.749$, which indicate that both personalized models substantially outperformed the generic model, and the performances of the two personalized models are similar. Note that we only plot the ROC curve for the user-behavior personalized model with the best performance here, which only uses the removal action type information to create behavior feature vectors.

affect the design of an intelligent notification system, and identify several limitations of our study and their potential implications on the results.

6.1 Lessons Learned About Users' Mental Models of an Ideal Notification Manager

With factors that naturally emerged from the free-form explanations of notification delivery preferences, we are able to gain a better understanding of what factors may be more important to users in determining how to deliver a notification (RQ1). In the following, we discuss some lessons learned about designing an intelligent notification system that fits users' expectations.

Users seem to be fairly risk-averse when making delivery decisions for their notifications. People seem to err on the side of being notified (56% preferred an immediate alert, 26% preferred suppressing the alert, while only 6.3% preferred deferring the alert), seemingly due to the fear of missing out on important or useful information. For example, some subjects were fine receiving promotional notifications if there was a small chance that they could benefit. This echoes the dilemma of the positive and negative effects of notifications discussed in prior work [23, 41, 43]. Specifically, prior research in e-mail spam filtering has a similar assumption that the consequences of information loss or delay depends on the nature of the message [7]. Furthermore, the preference of suppressing *alerts* over deferring *alerts* may be related to previous observations that users often consciously leave some or all notifications in the notification tray until a later time [52] and users snooze notifications [53]. We speculate that users want to gain more control over their notifications and therefore are more inclined to manually defer reading or taking actions on notifications rather than let the system do so. Note that our study was focused on the timing of the *alert* rather than the *delivery*.

The frequent reference to notification content in survey explanations suggests that notification content plays an important on users' notification preferences. This result echoes findings of prior work [11, 33], while we revealed more nuances about how content and contextual factors affect users' mental models of how an ideal intelligent notification system might work. As it seems to be more natural for subjects to reason about their preferences in terms of content features, we suggest notification systems should provide more support to help users manage notifications based

on notification content, such as grouping notifications with similar characteristics across apps to allow for more convenient control and identifying different notification patterns within an app to allow for more flexible control. It may also be useful for these systems to automatically detect intrinsic characteristics of a notification such as time-sensitivity, user's relationship with people mentioned in it, whether driven by the user and digital/physical actions that can be triggered by it. Since these factors were frequently used as justifications of notification delivery preferences, existing algorithms that mine notification management rules [30] may improve the interpretability of the system's decisions by providing explanations of the generated rules based on these factors.

A focus on content features is further called for because of the availability of contextual signals. Many contextual features mentioned by users in their explanations, such as the knowledge of whether the user has already seen the information in the notification, or the user's current task when it does not involve the phone, are difficult if not impossible for the notification system to obtain. Fortunately, our classification experiments suggest that reasonable results may be possible relying only on the content that is available and not on contextual information that may not be.

Taking a step back, we want to stress that although users mentioned contextual factors less often, this does not mean contextual factors are less important. First, our analysis has shown that users' perceptions of certain notification content may vary in different contexts and from person to person. Second, we chose a relatively brief expiration threshold (5 minutes) for our ESM tasks, which may bias users toward responding to ESM surveys when they are available and less affected by context. Third, given that we were using open-ended questions, the less frequent reference to contextual factors could be because our subjects were not fully aware of what we sought to obtain from the question. Hence, we want to caution that the actual influence of contextual factors may be higher than what was measured in our study.

6.2 Lessons Learned About the Relationship Between Subjective Preferences and Observable Behaviors

In Section 4 (RQ2), our qualitative analysis showed that similar actions can be driven by different reasons associated with different delivery preferences. Then we quantitatively showed that the response time and action type were both weak predictors of delivery preference for the same notification. These results offer takeaways for using observable user actions in designing intelligent notification systems.

First, the relationship between user actions and user preferences seems to be moderated by the notification content. That is, it may be helpful to take into account notification types when using user actions to infer delivery preferences. We have seen Mehrotra et al. [30] specifically discussed reminder notifications that do not require follow-up actions. Our systematic analysis of users' rationales behind their actions allowed us to expand on this special case and identify more related scenarios. For example, some notifications contained enough information so users did not need to click on it to read further, such as a news notification or a notification containing a verification code. Some notifications were a confirmation or a notice about an expected action, such as a download complete notice or a credit card payment confirmation.

Another important takeaway is that users may not use notification systems as they were designed, which suggests the intentions behind users' direct interaction with notifications may be clarified when more digital traces are collected and analyzed. We use three examples to demonstrate this point. First, the code "Want to take action but not via the notification" showed that users sometimes chose to directly open the app to complete actions instigated by a notification. Second, users appropriated notification systems for a much wider range of applications than initially intended such as using the notification tray as a type of to do list, which is related to the code "Intentionally delay the action". Third, the delivery of notification in a multi-device condition

has been discussed in some work [8, 31] and we further demonstrated that users still preferred two-thirds of notifications coded as "Seen elsewhere" to be delivered with an immediate alert.

Then in Section 5 (RQ3), we demonstrated that although the response time and action type of a notification do not seem to be effective features for inferring delivery preferences, using actions aggregated on similar notifications received in the past can be an effective substitute for user labels to build personalized models with decent performance. Our current models only use two basic types of features: notification type and response time. Given that our qualitative analysis has identified other aspects of user behavior that may correlate with their delivery preferences, such as app usage before or after receiving and dismissing the notification, we expect incorporating these measures may further improve the prediction accuracy. In addition, as people seemed to use different strategies to handle different types of notifications, NLP features could be used to automatically identify high-level categories of a notification [15] and we can feed this information to the model to make more targeted prediction. Due to the small size of our sample, these sophisticated model designs can only be left for future work to evaluate.

6.3 Design Implications: An Alternative Design of Intelligent Notification Systems

Given the above findings, we argue algorithms predicting when and how to deliver notifications may need to be incorporated into a more nuanced, holistic, mixed-initiative notification management approach to ensure accuracy and user control. For example, it may be a promising approach to provide a flexible interface for users to configure notification preferences based on the app and the notification content, coupled with automatic notification preference tuning.

We therefore propose an alternate design for an intelligent notification system. Unlike other methods that predict the opportune time to send notifications based mainly on contextual factors [1, 11, 20, 37, 41, 45] and prevent missing information via a bounded deferral mechanism [20], our approach primarily focuses on notification content. Users could set the urgency of notifications using hand-crafted keyword-/template-based filters, or adjusting a filtering-aggressiveness threshold to choose filters automatically created by the system. The threshold-adjusting idea could be supported by our template identification process and the context-agnostic preference prediction models introduced in Section 5, and we create a preliminary user interface design to help illustrate this concept in Figure 9.

Current mainstream smartphone operating systems already provide notification management features, such as specifying the importance of notifications per app and per notification channel and snoozing notifications.[7] However, our study results suggest that they still fall short for adoption, as seen for example in the extremely low utilization rate of SNOOZE. This further illustrates the importance of automatic notification preference tuning. Our initial exploration in preference prediction has demonstrated that we could achieve substantial improvement upon the generic model in multiple aspects if a few preference labels on previous notifications are available or some user behavior data is available. We believe there is a large space for future work to study better data collection methods and better prediction algorithm design.

6.4 Limitations and Future Work

As with any study, our work has limitations that must be considered when evaluating its results.

One limitation of including a question about a proposed system behavior, in our case delaying the time of the alert, is that users may not necessarily realize the potential benefits of a feature they have not experienced. Thus, we want to note that this result should not be interpreted as proving that opportune moment detection is not valuable. Instead, we argue that the deferment option may

[7] https://developer.android.com/about/versions/oreo/android-8.0.

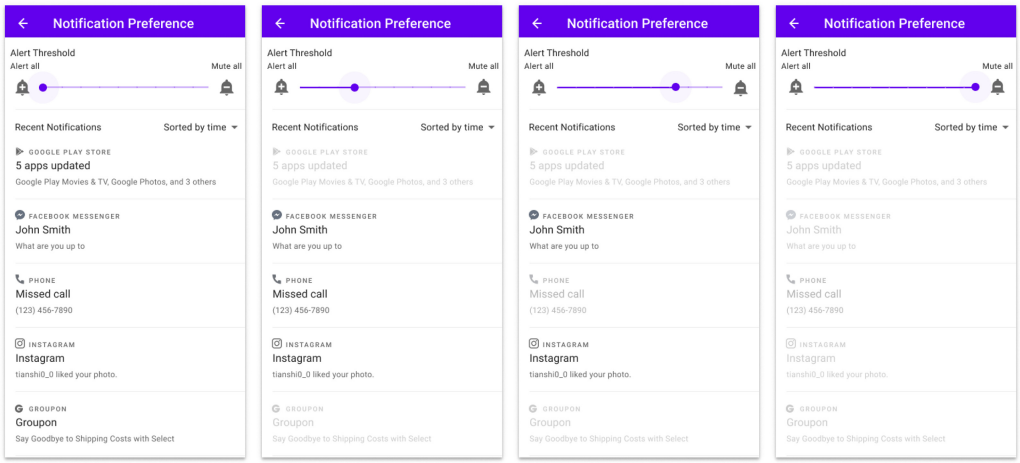


Fig. 9. We envision the content-based classifiers in Section 5 can be used to facilitate notification preference settings, and propose one possible design. In our design, users can use a slider to choose a threshold that reflects their preferred filtering strength and view their previously received notifications as examples to demonstrate what notifications will cause alerts and what notifications will be sent silently (greyed out). This allows users to gain a basic understanding of how the intelligent system works, and have some control over it. This design is mainly informed by the finding that users associate the preference of a notification more with its content, and their perceived urgency of notifications with similar content seem to be consistent and stable.

be more acceptable to users when the concerns of the risk of information loss are addressed, e.g., making sure important notifications will always be delivered on time.

As noted by previous research [46], the ESM has an inherent bias: being able to answer the survey already implies that the subject is likely to be more tolerant of an interruption. Data collected in this way may underestimate the frequency of the moment when people find a notification to be disruptive. Although we granted a 5-minute time window for answering the surveys, this bias may still exist. As a result, we might expect a larger portion of notifications to be deemed inappropriate because of the disruption caused by the study, and thus delivery preferences to favor no alerts or even complete suppression of the notification.

The duration of our study is relatively short (1 week), although the same length of study can be found in prior work with similar study design (logging + daily diary surveys [41], logging + ESM surveys [54]). The week-long study design results in more notifications received on weekdays than on weekends in the training set, which may affect the qualitative and quantitative analysis in Sections 3 and 4. However, the drop-off rates in Figure 3 seem to justify our choice of study duration, which suggests that drop off may have reached 100% by day 11 of a longer study.

We were limited in the number of surveys that we believed users would respond to, especially because our surveys required free-text answers and typing long passages on a phone can be tiring. However, the fact that we were able to obtain statistically significant difference for the main results suggests the sample size is sufficient for our goals.

In this study, we did not analyze the modalities used for sending the alert and could not take into account how users configured modalities to manage interruption [9] and its implications on users' delivery preferences. As previous research showed that different people prefer different modalities for notifications with varying priority and from different people [6, 29], we would like to further investigate factors that affect users' preferences of different modalities and the corresponding prediction task in the future.

7 RELATED WORK

Notification systems have been studied in HCI and Ubiquitous Computing in the context of desktop computers, smartphones, and wearable devices. A myriad of empirical studies, user preference models, and system designs have been proposed to address the perceived negative impact of the disruption caused by notifications. Here, we summarize major related work in three directions

(1) improving notifications based on subjective user feedback,
(2) improving notifications based on objective user behavior, and
(3) studying design guidelines for intelligent notification systems.

7.1 Improving Notifications Based on Subjective User Feedback

Some early work borrowed the concept of the "breakpoint" from classic psychology literature and proposed a "defer-to-breakpoint" policy to manage notifications [1]. The main idea is to exploit the natural transition between two units of the primary tasks (i.e., the breakpoint) to deliver the notification. Breakpoint detection has been studied for both physical activities and interactions with corresponding digital devices. Research has found that this mechanism has positive effects on reducing mental load and task resumption lag [22, 35–37].

Another line of work has studied interruptibility, asking users to evaluate whether a particular moment is good for an external interruption and aimed at scheduling notifications at an opportune moment. Past research that falls into this category usually follows a traditional machine learning research methodology, where ESMs [17] are first used to collect users' feedback on incoming notifications and then predictive models are built of users' interruptibility using the dataset. The predictive model can be used by an intelligent system to automatically defer notifications to more appropriate moments [1, 11, 20, 37, 41, 45].

Both the defer-to-breakpoint policy and opportune moment detection models heavily focus on understanding the user's context when receiving the notification, reflected by the use of contextual features such as location, time, physical activity, and phone usage when building the classifier [10, 13, 14, 38, 44, 48, 55]. Conversely, most of this previous work did not consider the content of the incoming notification when making predictions. The effect of notification content on users' perception of the interruption was usually studied by sending artificial notifications from the study app [38] or only studying a certain type of notification (e.g., phone call [49] or news [11, 37]), which leads to a gap between these interruptibility models and working solutions of notification systems.

Recent work has begun to systematically examine the notifications received on today's mobile systems and how content affects users' perception of the interruption. Fischer et al. [11] controlled topics of news pushed to users during a study via notifications, and found that the entertainment value, relevance, and actionability of the content seem to have more effect than delivery timing on users' impressions of notifications. Mehrotra et al. [34] analyzed a large-scale notification dataset collected in the wild. Their results showed that the perceived disruption can be influenced by the sender-recipient relationship. Iqbal et al. [23] found that people were willing to tolerate mobile interruptions in order not to miss important updates, which resonates with our study. As more work has pointed out the important role that notification content plays in users' subjective feelings about the interruption caused by a certain notification, researchers started expanding the meaning of interruptibility and integrating content features into predictive models [8, 32, 33].

Although notification content is considered more frequently by intelligent notification management systems, its use has also exposed more issues. For example, inferences based on context and content may not always be consistent. When contextual-based interruptibility is low but notification importance is high, what should the system do? Some work suggests that notification content may be more influential than the context [11], while other research observed the opposite [26]

that context overshadowed the sender-recipient relationship in predicting the attentiveness, responsiveness, interruptibility, and opportuneness of the notification.

In our work, we framed our query to users from a different perspective, as a chance to adjust how the system delivered the notification (see Section 3.2). Unlike prior research that fixed dependent variables before running the study [11, 26, 34], we asked users to explain their choices in a free form response. In this way, we attempted to eliminate the potential bias of priming users with certain factors, so we could learn about the dominant factors that naturally emerged in their responses. Our results showed that content factors (e.g., importance, notification type, time-sensitivity) were mentioned much more frequently than contextual factors (e.g., interruptibility, location, time, phone usage), which suggests that smartphone users expect more system-level control on their notifications based on *what* the notification is, as compared to *when* it is delivered to them.

7.2 Improving Notifications Based on Objective User Behavior

A critical challenge in modeling user preferences is that collecting subjective labels usually incurs a considerable cost to users because these queries are also interruptions. This cost directly leads to the sparsity of the labeled data [34], making it hard for the model to fit users' preferences in a supervised manner. The intrinsic disruptive nature of the queries may also bias the dataset towards moments that are more appropriate to users, where users have time to handle the study queries [46].

Therefore, some work adopted an alternative approach, which used observable user behaviors as the source of feedback. *Engagement* and *responsiveness* are two common metrics in past literature. Engagement is determined by the way the notification is handled, specifically focusing on whether a notification is accepted (click to open) or discarded (swipe to remove). Some work [30, 40, 51] directly used whether the user engages with the notification by accepting it as the prediction target. Others applied reinforcement learning and treated the user action as the reward function [19]. Responsiveness is usually quantified by the response time, the time from the user receiving or seeing a notification to the time they removed the notification. Mehrotra et al. [33] set a hard response time threshold and considered all instances that had a shorter response time than the threshold to have been appropriate and instances with a longer response time to be inappropriate.

In contrast, we find in our work that nuanced user preferences cannot be characterized by response time or the action type alone. Unstructured reasons provided by our participants offer some explanation of this difference. For example, users delayed acting on important notifications because sufficient information was available in the notification preview text. In other cases, they dismissed a notification quickly if it was completely irrelevant, or chose to defer the bulk clean up of irrelevant notifications to a later time. We further learned in the predicting algorithm experiments that although some user actions only have very weak predictive power for a single notification, aggregating them for a group of similar notifications received in the past can generate a much more stable and accurate prediction for a new incoming notification.

7.3 Studying Design Guidelines for Intelligent Notification Systems

Empirical studies and predictive modeling have provided an important foundation for designing intelligent notification systems. Past research has proposed various system designs, which aimed at either reducing the disruption or improving the engagement by deferring the notification to a more appropriate time [1, 2, 11, 20, 37, 41, 45], filter out unimportant notifications [2, 30], or automatically choose the appropriate modulation for the notification alert [12, 28, 46].

However, due to the nuances of individual's different delivery preferences, current physical and mental status, and future plans, it is exceptionally difficult to achieve near perfect accuracy on statistical modeling tasks [32, 50]. Furthermore, a misprediction may cause a large disruption and

users would not accept a system that defers or stops an important notification [34]. This means that a system for notification management is unlikely to be able to rely on modeling alone in order to provide a high-quality experience. As a result, in our study we do not focus our data collection solely on the opportuneness of notification timing but on the larger space of notification management.

In the same spirit, some previous research focused not only on improving accuracy, but also aimed at building explainable intelligent notification systems to give users a better understanding and stronger feelings of trust in the system [27]. PrefMiner [30] is a system that aims at automatically dismissing unimportant notifications. The system mines human-readable rules to manage notifications and requests for user approvals before using the rules. In this way, users can benefit from the automatic management and also stay in the control. In this article, we propose algorithms that can be used to automatically initiate the notification delivery preference configurations for notifications from different apps and notifications using different templates from the same app, which also aims at reducing the manual labor when still keep the systems' actions intelligible and controllable.

Recently, the designs of the commercial systems and proposals from academic papers [2, 8, 47, 48] also show a trend to provide users with more fine-grained ability to control notifications. However, our study demonstrates that some existing features in commercial system (e.g., snooze notification) has an extremely low utilization rate. We consider proper automation may still be an indispensable part for such systems to achieve success.

8 CONCLUSION

In this article, we study how to improve notifications by directly asking users about their preferred delivery method for a notification. We conducted a mixed-methods study, which includes both experience-sampling studies that solicited users' notification delivery preferences and passive data collection that records how people used their phones and handled notifications during the course of the study. We learned that subjects seemed to refer to notification content factors (e.g. importance, notification type and timesensitivity) more frequently when explaining their delivery preferences than contextual factors (e.g., interruptibility, phone usage). Overall, we noticed a strong emphasis of avoiding the risk of losing important information and a significant impact of individual difference in the delivery preferences of users from our sample.

Then we further studied the relationship between user behavior and notification delivery preferences, with the hope that the easily accessible objective behavior data can help us improve the prediction performance for notification delivery preferences of an individual user. We adopted a context-agnostic approach, where we assign fixed preference predictions to the same type of notifications (e.g., notifications from the same app, notifications using the same template). Our results showed that our algorithm could adapt to individual preferences by leveraging either user-specified preferences for notifications received in the past or user behavior observed after receiving notifications in the past. Finally, we discussed design implications of our findings and proposed an alternative design of an intelligent notification system that can be supported by our prediction model.

APPENDICES

A APPENDICES WILL APPEAR IN SUPPLEMENTAL MATERIAL

Please note that the appendices included here will be moved to a separate supplemental material document for publication.

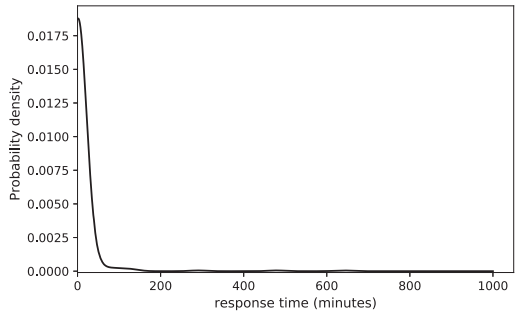


Fig. 10. Response time probability density in the labeled part of the dataset, which had a long-tail distribution. (estimated with Kernel density estimation, $bandwidth = 20$)

Table 5. Raw Count and the Ratio of Different Removal Action Types

Action name	All (train)	All sampled (train)
CLICK	1,854 (9.3%)	147 (23%)
CANCEL	5,326 (27%)	243 (37%)
CANCEL_ALL	389 (1.95%)	28 (4.3%)
SNOOZE	12 (0.060%)	1 (0.15%)
AUTOMATIC	6,676 (34%)	142 (22%)
OVERWRITE	4,897 (25%)	84 (13%)
Not removed	792 (4.0%)	4 (0.62%)

CANCEL seems to be the most commonly used action for smartphone users to handle notifications.

B PACKAGE NAMES USED FOR FILTERING OUT BULLETIN NOTIFICATIONS

— android
— com.lge.systemservice
— com.motorola.vzw.settings.extensions
— com.google.android.googlequicksearchbox
— com.google.android.apps.maps
— com.google.android.music
— com.spotify.music
— au.com.shiftyjelly.pocketcasts

C ADDITIONAL ANALYSIS RESULTS

C.1 User Behavior Overview and the Impact of ESM

Overall, subjects believed they handled the sampled notifications immediately after seeing them in most cases, with only 22.1% notifications marked as not immediately discarded/accepted/snoozed (83 notifications). We also found a consistent trend in the response time, the time between seeing and acting on a notification, which has a long-tail distribution (Figure 10).

Table 5 shows the raw count and the ratio of different actions that caused a notification to be removed. The first four rows ("CLICK", "CANCEL", "CANCEL_ALL", "SNOOZE") represent user-driven actions that remove notifications, and the next two rows ("AUTOMATIC", "OVERWRITE") represent two situations where the app automatically removed the notification before the user took

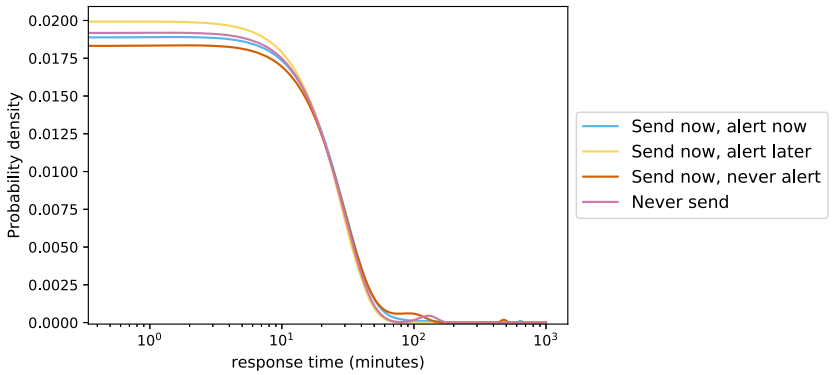


Fig. 11. The distributions of response times only vary slightly across notifications of all four preferences. (estimated with Kernel density estimation, $bandwidth = 20$; response times plotted in logarithmic scale).

any action. The last row ("Not removed") represents a special situation where the notification was never removed during the study.

Below are some notes on the impact of our experience sampling methodology on this data. Overall, the impact of ESM tasks and sampling rules seem to be small, which suggests that the usage data collected in our study could reflect users' natural behavior.

- The median of response time among all notifications and notifications sampled for ESM tasks are 9.6 seconds and 7.8 seconds, respectively. The small difference seems to suggest that the ESM tasks did not substantially alter user behavior. The difference is not statistically significant (p = 0.345) under Mann-Whitney U test.
- The distributions of notification removal actions are similar between the "All (train)" and "All sampled (train)" subset, which also suggests that our ESM tasks did not substantially alter users' behavior. The sampling bias generally seen in ESM studies may account for this difference and the previous one.
- The ratios of notifications that were removed by the app or never removed during the study are consistently higher in the "All (train)" group than in the "All sampled (train)" group. This is likely because a prerequisite of sampling a notification for the ESM tasks is that it was seen by the user, who would then have had the opportunity to act upon it.
- The relative ratio of cancel actions to click actions is higher in the "All (train)" group than the "All sampled (train)" group. This is likely due to group notifications, which were not sampled for ESM tasks and are easy for subjects to remove en masse with a single swipe. See Section 2.2.1 for more details of this sampling choice.

C.1.1 Data Visualization of Response Times and Notification Preferences. Figure 11 is a plot of the distributions of response times across notifications of the four delivery preferences, which is intended to give insight into the relationship between these two variables. All four probability density functions are estimated with kernel density estimation using Gaussian kernels ($bandwidth = 20$).

It is notable that the distribution of response times seem to vary only slightly across all four delivery preferences. This suggests that a response time threshold is unlikely to be a workable method of inferring the delivery preference of a notification.

C.1.2 Data Visualization of Action Types and Notification Preferences. Figure 12 shows the ratios of different preference options for notifications removed due to different actions. We excluded

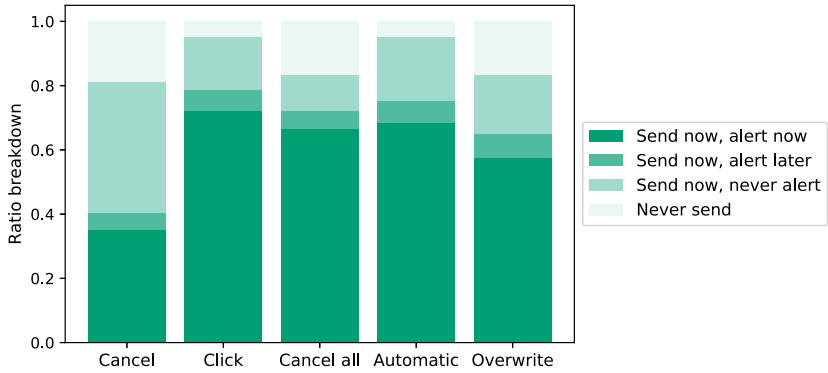


Fig. 12. User preference breakdown for different removal actions. We can see that all five types of removal actions have a considerable portion of notifications preferred an immediate alert, which suggests the removal action type information only has weak predictive power for notification delivery preferences.

the snooze action and "not removed" status, because both had too few labeled data points (1 for "snooze", 4 for "not removed").

From the figure, we can see that there are some difference in the preference distribution among different action types, especially for the cancel action. However, there are still a considerable portion of notifications that were preferred to be delivered with an immediate alert for all five action types, which suggests that only using the removal action type information may not be sufficient to infer users' preferences.

C.1.3 Summary. The data visualization and statistical modeling experiment results both seem to contradict the assumptions in prior research [3, 19, 30, 33, 38, 40, 51]. Based on our data, we find the response time and removal action type cannot be reliably used to determine whether and how to send a notification. Quicker reactions to a notification does not always mean the user feels positively about the notification, and dismissing a notification does not always suggest a negative feeling. The next section will show examples to help explain this seemingly counter-intuitive result.

REFERENCES

[1] Piotr D. Adamczyk and Brian P. Bailey. 2004. If not now, when? The effects of interruption at different moments within task execution. In *Proceedings of the 2004 Conference on Human Factors in Computing Systems*. ACM. DOI: https://doi.org/10.1145/985692.985727

[2] Jonas Auda, Dominik Weber, Alexandra Voit, and Stefan Schneegass. 2018. Understanding user preferences towards rule-based notification deferral. In *Proceedings of the Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM. DOI: https://doi.org/10.1145/3170427.3188688

[3] Daniel Avrahami and Scott E. Hudson. 2006. Responsiveness in instant messaging: Predictive models supporting interpersonal communication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. DOI: https://doi.org/10.1145/1124772.1124881

[4] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. DOI: https://doi.org/10.1191/1478088706qp063oa

[5] Christopher D. Brown and Herbert T. Davis. 2006. Receiver operating characteristics curves and related decision measures: A tutorial. *Chemometrics and Intelligent Laboratory Systems* 80, 1 (2006), 24–38. DOI: https://doi.org/10.1016/j.chemolab.2005.05.004

[6] Yung-Ju Chang, Yi-Ju Chung, and Yi-Hao Shih. 2019. I think it's her: Investigating smartphone users' speculation about phone notifications and its influence on attendance. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM. DOI: https://doi.org/10.1145/3338286.3340125

[7] Gordon V. Cormack. 2008. *Email Spam Filtering: A Systematic Review*. Now Publishers Inc. DOI: https://doi.org/10.1561/9781601981479

[8] Fulvio Corno, Luigi De Russis, and Alberto Monge Roffarello. 2018. AwareNotifications: Multi-device semantic notification handling with user-defined preferences. *Journal of Ambient Intelligence and Smart Environments* 10, 4 (2018), 327–343.

[9] Tilman Dingler and Martin Pielot. 2015. I'll be there for you: Quantifying attentiveness towards mobile messaging. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM. DOI: https://doi.org/10.1145/2785830.2785840

[10] Joel E. Fischer, Chris Greenhalgh, and Steve Benford. 2011. Investigating episodes of mobile phone activity as indicators of opportune moments to deliver notifications. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*. ACM. DOI: https://doi.org/10.1145/2037373.2037402

[11] Joel E. Fischer, Nick Yee, Victoria Bellotti, Nathan Good, Steve Benford, and Chris Greenhalgh. 2010. Effects of content and time of delivery on receptivity to mobile interruptions. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*. ACM. DOI: https://doi.org/10.1145/1851600.1851620

[12] Robert Fisher and Reid Simmons. 2011. Smartphone interruptibility using density-weighted uncertainty sampling with reinforcement learning. In *Proceedings of the 2011 10th International Conference on Machine Learning and Applications and Workshops*. IEEE. DOI: https://doi.org/10.1109/icmla.2011.128

[13] Claudio Forlivesi, Utku Günay Acer, Marc van den Broeck, and Fahim Kawsar. 2018. Mindful interruptions: A lightweight system for managing interruptibility on wearables. In *Proceedings of the 4th ACM Workshop on Wearable Systems and Applications*. ACM. DOI: https://doi.org/10.1145/3211960.3211974

[14] Claudio Forlivesi, Marc van den Broeck, Utku Günay Acer, and Fahim Kawsar. 2018. On-Wearable AI to model human interruptibility. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. ACM. DOI: https://doi.org/10.1145/3210240.3210814

[15] Kieran Fraser and Owen Conlan. 2020. Enticing notification text & the impact on engagement. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*. ACM. DOI: https://doi.org/10.1145/3410530.3414430

[16] Andrew F. Hayes and Klaus Krippendorff. 2007. Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures* 1, 1 (2007), 77–89. DOI: https://doi.org/10.1080/19312450709336664

[17] Joel M. Hektner, Jennifer A. Schmidt, and Mihaly Csikszentmihalyi. 2007. *Experience Sampling Method: Measuring the Quality of Everyday Life*. Sage.

[18] D. S. Hirschberg. 1975. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM* 18, 6 (1975), 341–343. DOI: https://doi.org/10.1145/360825.360861

[19] Bo-Jhang Ho, Bharathan Balaji, Mehmet Koseoglu, and Mani Srivastava. 2018. Nurture: Notifying users at the right time using reinforcement learning. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. ACM. DOI: https://doi.org/10.1145/3267305.3274107

[20] Eric Horvitz, Johnson Apacible, and Muru Subramani. 2005. Balancing awareness and interruption: Investigation of notification deferral policies. In *Proceedings of the International Conference on User Modeling*. Springer, Berlin, 433–437. DOI: https://doi.org/10.1007/11527886_59

[21] Shamsi T. Iqbal, Piotr D. Adamczyk, Xianjun Sam Zheng, and Brian P. Bailey. 2005. Towards an index of opportunity: Understanding changes in mental workload during task execution. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. DOI: https://doi.org/10.1145/1054972.1055016

[22] Shamsi T. Iqbal and Brian P. Bailey. 2006. Leveraging characteristics of task structure to predict the cost of interruption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. DOI: https://doi.org/10.1145/1124772.1124882

[23] Shamsi T. Iqbal and Eric Horvitz. 2010. Notifications and awareness: A field study of alert usage and preferences. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*. ACM. DOI: https://doi.org/10.1145/1718918.1718926

[24] Nicky Kern and Bernt Schiele. 2006. Towards personalized mobile interruptibility estimation. In *Proceedings of the International Symposium on Location- and Context-Awareness*. Springer, Berlin, 134–150. DOI: https://doi.org/10.1007/11752967_10

[25] Kostadin Kushlev, Jason Proulx, and Elizabeth W. Dunn. 2016. "Silence your phones": Smartphone notifications increase inattention and hyperactivity symptoms. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM. DOI: https://doi.org/10.1145/2858036.2858359

[26] Hao-Ping Lee, Kuan-Yin Chen, Chih-Heng Lin, Chia-Yu Chen, Yu-Lin Chung, Yung-Ju Chang, and Chien-Ru Sun. 2019. Does who matter?: Studying the impact of relationship characteristics on receptivity to mobile IM messages. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM. DOI: https://doi.org/10.1145/3290605.3300756

[27] Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. 2009. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. DOI: https://doi.org/10.1145/1518701.1519023

[28] Hugo Lopez-Tovar, Andreas Charalambous, and John Dowell. 2015. Managing smartphone interruptions through adaptive modes and modulation of notifications. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*. ACM. DOI: https://doi.org/10.1145/2678025.2701390

[29] Afra Mashhadi, Akhil Mathur, and Fahim Kawsar. 2014. The myth of subtle notifications. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM. DOI: https://doi.org/10.1145/2638728.2638759

[30] Abhinav Mehrotra, Robert Hendley, and Mirco Musolesi. 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*. ACM. DOI: https://doi.org/10.1145/2971648.2971747

[31] Abhinav Mehrotra, Robert Hendley, and Mirco Musolesi. 2019. NotifyMeHere: Intelligent notification delivery in multi-device environments. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*. ACM. DOI: https://doi.org/10.1145/3295750.3298932

[32] Abhinav Mehrotra and Mirco Musolesi. 2017. Intelligent notification systems: A survey of the state of the art and research challenges. arXiv:1711.10171. Retrieved from https://arxiv.org/abs/1711.10171.

[33] Abhinav Mehrotra, Mirco Musolesi, Robert Hendley, and Veljko Pejovic. 2015. Designing content-driven intelligent notification mechanisms for mobile applications. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. DOI: https://doi.org/10.1145/2750858.2807544

[34] Abhinav Mehrotra, Veljko Pejovic, Jo Vermeulen, Robert Hendley, and Mirco Musolesi. 2016. My phone and me: Understanding people's receptivity to mobile notifications. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM. DOI: https://doi.org/10.1145/2858036.2858566

[35] Tadashi Okoshi, Julian Ramos, Hiroki Nozaki, Jin Nakazawa, Anind K. Dey, and Hideyuki Tokuda. 2015. Attelia: Reducing user's cognitive load due to interruptive notifications on smart phones. In *Proceedings of the 2015 IEEE International Conference on Pervasive Computing and Communications*. IEEE. DOI: https://doi.org/10.1109/percom.2015.7146515

[36] Tadashi Okoshi, Julian Ramos, Hiroki Nozaki, Jin Nakazawa, Anind K. Dey, and Hideyuki Tokuda. 2015. Reducing users' perceived mental effort due to interruptive notifications in multi-device mobile environments. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. DOI: https://doi.org/10.1145/2750858.2807517

[37] Tadashi Okoshi, Kota Tsubouchi, Masaya Taji, Takanori Ichikawa, and Hideyuki Tokuda. 2017. Attention and engagement-awareness in the wild: A large-scale study with adaptive notifications. In *Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications*. IEEE. DOI: https://doi.org/10.1109/percom.2017.7917856

[38] Veljko Pejovic and Mirco Musolesi. 2014. InterruptMe: Designing intelligent prompting mechanisms for pervasive applications. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. DOI: https://doi.org/10.1145/2632048.2632062

[39] Veljko Pejovic, Mirco Musolesi, and Abhinav Mehrotra. 2015. Investigating the role of task engagement in mobile interruptibility. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. ACM. DOI: https://doi.org/10.1145/2786567.2794336

[40] Martin Pielot, Bruno Cardoso, Kleomenis Katevas, Joan Serrà, Aleksandar Matic, and Nuria Oliver. 2017. Beyond interruptibility: Predicting opportune moments to engage mobile phone users. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–25. DOI: https://doi.org/10.1145/3130956

[41] Martin Pielot, Karen Church, and Rodrigo de Oliveira. 2014. An in-situ study of mobile phone notifications. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services*. ACM. DOI: https://doi.org/10.1145/2628363.2628364

[42] Martin Pielot and Luz Rello. 2015. The do not disturb challenge: A day without notifications. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM. DOI: https://doi.org/10.1145/2702613.2732704

[43] Martin Pielot and Luz Rello. 2017. Productive, anxious, lonely: 24 hours without push notifications. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM. DOI: https://doi.org/10.1145/3098279.3098526

[44] Benjamin Poppinga, Wilko Heuten, and Susanne Boll. 2014. Sensor-Based identification of opportune moments for triggering notifications. *IEEE Pervasive Computing* 13, 1 (2014), 22–29. DOI: https://doi.org/10.1109/mprv.2014.15

[45] Swadhin Pradhan, Lili Qiu, Abhinav Parate, and Kyu-Han Kim. 2017. Understanding and managing notifications. In *Proceedings of the IEEE INFOCOM 2017—IEEE Conference on Computer Communications*. IEEE. DOI: https://doi.org/10.1109/infocom.2017.8057231

[46] Stephanie Rosenthal, Anind K. Dey, and Manuela Veloso. 2011. Using decision-theoretic experience sampling to build personalized mobile phone interruption models. In *Proceedings of the Lecture Notes in Computer Science*. Springer, Berlin, 170–187. DOI: https://doi.org/10.1007/978-3-642-21726-5_11

[47] Luigi De Russis and Alberto Monge Roffarello. 2017. On the benefit of adding user preferences to notification delivery. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM. DOI: https://doi.org/10.1145/3027063.3053160

[48] Iqbal H. Sarker, Muhammad Ashad Kabir, Alan Colman, and Jun Han. 2017. Designing architecture of a rule-based system for managing phone call interruptions. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM. DOI: https://doi.org/10.1145/3123024.3124562

[49] Jeremiah Smith, Anna Lavygina, Jiefei Ma, Alessandra Russo, and Naranker Dulay. 2014. Learning to recognise disruptive smartphone notifications. In *Proceedings of the 16th International Conference on Human Computer Interaction with Mobile Devices and Services*. ACM. DOI: https://doi.org/10.1145/2628363.2628404

[50] Liam D. Turner, Stuart M. Allen, and Roger M. Whitaker. 2015. Interruptibility prediction for ubiquitous systems: Conventions and new directions from a growing field. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. DOI: https://doi.org/10.1145/2750858.2807514

[51] Liam D. Turner, Stuart M. Allen, and Roger M. Whitaker. 2017. Reachable but not receptive: Enhancing smartphone interruptibility prediction by modelling the extent of user engagement with notifications. *Pervasive and Mobile Computing* 40 (2017), 480–494. DOI: https://doi.org/10.1016/j.pmcj.2017.01.011

[52] Liam D. Turner, Stuart M. Allen, and Roger M. Whitaker. 2019. The influence of concurrent mobile notifications on individual responses. *International Journal of Human-Computer Studies* 132 (2019), 70–80. DOI: https://doi.org/10.1016/j.ijhcs.2019.07.011

[53] Dominik Weber, Alexandra Voit, Jonas Auda, Stefan Schneegass, and Niels Henze. 2018. Snooze!: Investigating the user-defined deferral of mobile notifications. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM. DOI: https://doi.org/10.1145/3229434.3229436

[54] Dominik Weber, Alexandra Voit, Philipp Kratzer, and Niels Henze. 2016. In-situ investigation of notifications in multi-device environments. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. DOI: https://doi.org/10.1145/2971648.2971732

[55] Fengpeng Yuan, Xianyi Gao, and Janne Lindqvist. 2017. How busy are you?: Predicting the interruptibility intensity of mobile users. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM. DOI: https://doi.org/10.1145/3025453.3025946

Received 2 April 2020; revised 18 July 2021; accepted 30 July 2021