PokeME: Applying Context-Driven Notifications to Increase Worker Engagement in Mobile Crowd-sourcing

Thivya Kandappu
Singapore Management University
thivyak@smu.edu.sg

Abhinav Mehrotra
University College London, UK
a.mehrotra@ucl.ac.uk

Archan Misra
Singapore Management University
archanm@smu.edu.sg

Mirco Musolesi*
University College London, UK
m.musolesi@ucl.ac.uk

Shih-Fen Cheng
Singapore Management University
sfcheng@smu.edu.sg

Lakmal Meegahapola
EPFL, Switzerland
lakmal.meegahapola@epfl.ch

ABSTRACT

In mobile crowd-sourcing systems, simply relying on people to opportunistically select and perform tasks typically leads to drawbacks such as low task acceptance/completion rates and undesirable spatial skews. In this paper, we utilize data from *TASKer*, a campus-based mobile crowd-sourcing platform, to empirically study and discover whether and how various context-aware notification strategies can help overcome such drawbacks. We first study worker interactions, in the absence of any notifications, to discover some spatio-temporal properties of task acceptance and completion. Based on these insights, we then experimentally demonstrate the effectiveness of two novel, non-personal, context-driven notification strategies, comparing the outcomes to two different baselines (no-notification and random-notification). Finally, using the data from the random-notification mechanism, we derive a classification model, incorporating several novel contextual features, that can predict a worker's responsiveness to notifications with high accuracy. Our work extends the crowd-sourcing literature by emphasizing the power of smart notifications for greater worker engagement.

CCS CONCEPTS

• Information systems → Crowdsourcing; **• Human-centered computing → Empirical studies in ubiquitous and mobile computing**; **User models**.

KEYWORDS

intervention techniques; notifications; mobile crowd-sourcing

* The author is also affiliated with The Alan Turing Institute, UK and University of Bologna, Italy.

CHIIR '20, March 14–18, 2020, Vancouver, BC, Canada

ACM ISBN 978-1-4503-6892-6/20/03…$15.00
https://doi.org/10.1145/3343413.3377965

ACM Reference Format:
Thivya Kandappu, Abhinav Mehrotra, Archan Misra, Mirco Musolesi*, Shih-Fen Cheng, and Lakmal Meegahapola. 2020. PokeME: Applying Context-Driven Notifications to Increase Worker Engagement in Mobile Crowd-sourcing. In *2020 Conference on Human Information Interaction and Retrieval (CHIIR '20), March 14–18, 2020, Vancouver, BC, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3343413.3377965

1 INTRODUCTION

This paper seeks to harness the power of *proactive notifications* for a novel class of mobile crowd-sourcing applications–those that seek engage residents of urban spaces for various smart campus/city services. Under this paradigm, residents voluntarily perform a variety of location-specific "tasks", that provide feedback about campus or city resources or that enable community-centric services. Examples of such platforms for citizen-centric engagement include *Citizen*[1], *CitySourced*[2] and *OneService*[3]). Fostering greater participation & increasing task acceptance/completion rates remain a formidable challenge for such platforms. A variety of different strategies have been proposed to increase such participation–for example, (a) delivering proactive recommendations of detour-minimizing tasks [11], (b) provisioning public displays to support situated feedback [9], (c) providing enhanced intrinsic motivation through explicit feedback [13] and (d) providing rewards to incentivize task execution at less popular locations [11, 29].

We look at developing notification (task reminder) strategies that suitably influence the *spatiotemporal profile* of tasks performed, and increase task acceptance and completion rates. Past research on user attention in ubiquitous applications has focused on both quantifying interruptibility [8, 24] and establishing how different contextual factors (e.g., time of day, activity [25] and group membership [17]) affect individual response to in-App notifications. We seek to discover how such contextual factors affect the efficacy of notifications for mobile crowd-sourcing scenarios, which possess two distinguishing characteristics: (a) Unlike tasks in the online world (e.g., emails or IMs), the user's response involves a *physical* action (namely, visiting the corresponding location to execute the task). (b) Second, the crowd-sourcing platform is primarily interested in *aggregated* outcomes –e.g., did notifications increase the overall task completion rate at low-participation locations?—and less concerned about individual-specific responses.

[1] https://www.citizen.com/
[2] https://www.citysourced.com/
[3] https://www.oneservice.sg/service/index/realtime

Our goal is to develop context-aware task reminder strategies that perform opportune interventions on users, to achieve desired changes in task acceptance/completion behavior. To achieve this goal, we utilize empirical data from a longitudinal study, involving 481 users of *TASKer*[4], a mobile crowd-sourcing platform operationally deployed on a university campus. We first analyze the spatio-temporal properties of worker interaction with *TASKer*, in the absence of any notifications, to identify some *novel* contextual factors (e.g., the residence duration of a worker at a given location) that affect aggregate task acceptance & completion rates. We then perform another study, involving 432 users over a total period of 6 weeks, using *PokeMe*, a *Smart Notification* component added to *TASKer*, to demonstrate the overall effectiveness of, and important differences between, 3 different *person-independent, context-aware* notification strategies. Finally, using empirical data on notification responses, we build a *person-specific* model that can accurately predict an individual worker's context-dependent response to notifications.

Research Questions and Contributions: We experimentally obtain insights to the following questions:

Are there any observable trends/patterns in task acceptance in the absence of notifications? Using historical data from the *TASKer* App, we investigate how (space,time) context affects task acceptance in the absence of explicit notifications. Specifically, in addition to the (expected) results that more popular (heavily visited) campus locations see greater task acceptance rates, we observe the novel result that the task acceptance and execution rates are higher for *staypoint* campus locations (e.g., cafeteria or classrooms, where people stay longer) as opposed to locations that are *transitory* (e.g., travel passageways) or that involve *small-group* interactions.

Do notification reminders affect task acceptance and performance rates? We show that different low-intensity notifications all significantly nudge worker-task outcomes: (a) increasing task acceptance rate to 22% (from just 2.44% in the absence of task reminders), and (b) achieving a dramatic three-fold reduction in task *dereliction rates* (the likelihood of eventually not completing an accepted task).

How do different context-based notification strategies affect the micro-dynamics of task acceptance & performance? We show key differences between 3 distinct location-based reminder strategies: *random*, *low-popularity* (where reminders are sent to users visiting low acceptance-rate locations) and *start-staypoint* (where reminders are sent whenever a student is observed in a classroom, where she is likely to reside for a while). In particular, *low-popularity* reminders effectively increase task acceptance rates in such locations by 12%, hence, improve the spatial fairness. *Start-staypoint* strategies cause users to accept tasks near (within the same building) as the current staypoint, whereas *random* notifications cause users to accept tasks while in transit (well before reaching the task location).

Can contextual factors help build an accurate personalized model of notification effectiveness?: Using the *random* notification dataset, we uncover several novel contextual factors that *significantly* impact the likelihood of accepting tasks in response to notifications-e.g.,(a) overall temporal location popularity, (b) per-user temporal location popularity, (c) closeness to task location and (d) stay-time duration. More importantly, the features help to accurately predict a user's receptivity to a task notification (AUC=0.86).

Broadly, our research helps establish the importance of incorporating context-aware notification strategies to enhance user engagement with mobile crowd-sourcing applications

[4]Anonymized to ensure double-blind reviewing

2 SPATIOTEMPORAL TASK ACCEPTANCE PATTERNS

We start our investigation by first asking: *"Are there observable trends/dynamics between the crowd-workers in the way they accept and perform tasks?"*. The overall goal is to understand the underlying role of *space* and *time* on task acceptance behavior.

To investigate this issue, we first examine 3 aggregate measures: (a) the popularity of locations in terms of the daily task completion rate, (b) whether preference for tasks varies between "stay locations" (where workers are stationary over longer periods) and transient locations and (c) time periods when a worker is *likely* to be more interruptible. Insights from these studies can then be used to identify opportune moments for interruption via task reminders, and thereby help shape overall worker behavior and task outcomes. To make the subsequent discussion clearer, we define two terms: (a) *Task Acceptance Location:* this is the location of a crowd-worker at the time that she accepts an available task, and (b) *Task Location:* this is the location associated with the specific task–i.e., the location which the worker has to visit to execute the task.

2.1 Our Dataset

Throughout this paper, we make use of 2 different datasets, both from an urban university campus: (1) mobility traces of users obtained through a Wi-Fi-based indoor localization platform, and (2) user behaviors and task-related data obtained from an university-wide mobile crowd-sourcing platform.

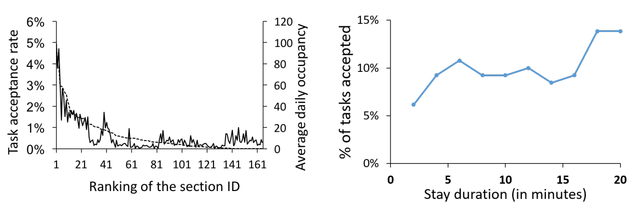
Wi-Fi based indoor localization data: The dataset comprises indoor location data, collected from over 10,000 students connected to the campus Wi-Fi, derived using a Wi-Fi fingerprinting based localization algorithm. The location traces have room-level accuracy (error of 6-8 meters) and a refresh rate of 5-6 seconds.

***TASKer* data:** *TASKer* is an experimental mobile crowd-sourcing platform deployed throughout the university. *TASKer* utilizes student workers to crowd-source reports on various facilities and resources on campus (e.g., the cleanliness of restrooms or the wait times in cafeterias). Over a period of 3.5 years, it involved a participant base of more than 1400 users, who have completed more than 150,000 tasks. In this work, we use a longitudinal subset of *TASKer* data, comprising more than 9,000 tasks performed by 488 students over a span of 6 weeks (from 23 February till 6 April 2017). For each user, we capture the following categories of information: (a) user profile, (b) App usage (e.g., App browsing time, frequency of the App usage and session duration) and (c) task-related transactions (e.g., location of the task, time at which the task got accepted and performed, etc.).

2.2 Task Acceptance Trends on an Urban Campus

2.2.1 Acceptance Popularity of Locations. We first study the impact of footfall on overall task acceptance rates. More specifically, for each task location, we first find, from historical mobility traces, the number of user visits (aggregated across all users) to the location within the task validity period. We then compute the nominal task acceptance rate for each location, by dividing the number of accepted task instances by the total number of posted tasks in that location, and compute the task acceptance rate, at a location l, is obtained by normalizing this nominal task acceptance rate by the total number of user visits–i.e., $TaskAcceptanceRate(l) = \frac{\text{Nominal Task Acceptance Rate (l)}}{\text{Total User Visits(l)}}$. Fig. 1a plots this task acceptance rate in descending order, with the X-axis denotes the task acceptance rate-based rankings of all campus locations – we also plot the averaged total user visits of the locations during the tasks validity period, in secondary Y-axis.) We see that the acceptance rate is highly skewed (Jain's fairness index=0.3). Moreover, the task acceptance rate does not always conform to the total number of user visits–the Spearman Rank Correlation coefficient is 0.46 with a two-tailed p=0.002. Collectively, these observations suggest that workers favor tasks at certain locations, and that these are not always the most-visited locations. We conjecture that certain locations may not be conducive to for App-related interactions, as these places either permit less distractions (e.g., group study rooms, library) or may be purely transit passages (e.g., the underground concourse and corridors)

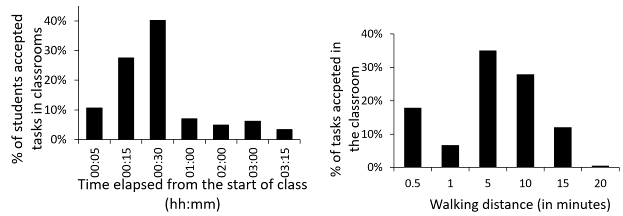
The observed skewness in such task acceptance rate naturally begets the question: what strategies can help increase task execution (by improving acceptance) at less popular locations (e.g., past work has looked at differential pricing, with greater rewards for less popular locations–e.g., [12])? We shall explore whether novel *notification* strategies can help tackle this issue.



(a) Aggregate visit count **(b) Length of stay episodes**

Figure 1: (a) Task Acceptance Rate & Aggregate Visit Count across locations, and (b) Length of stay episodes and task acceptance

2.2.2 Stay Locations and Residency Time. We next study whether the task acceptance dynamics is moderated by a worker's *mobility vs. stationary* behavior. More specifically, to analyze the possible impact of residency time on task acceptance, we first transform each user's mobility trace, using the previously described Wi-Fi indoor location data, into *trajectories*, that consist of 'stay episodes', punctuated by 'transient intervals'. Formally, an episode is represented by a tuple $< u, l, d, t_s, t_e >$, where u is the ID of the student, l is the room-level location ID, d is the day of the week, t_s is the starting time and t_e is the ending time of the episode. Episodes longer than 5 minutes (i.e., instances where the user resided at one location for longer than 5 minutes) are designated as 'stay episode', with the remaining periods of time being labeled as 'transient intervals'. For greater accuracy, we eliminated devices (or users) that are extremely stationary (such as laptops) by filtering out stay episodes longer than 4 hours, especially as the maximum class duration is 3 hours and 15 minutes. From the empirical WiFi data, which can track a student's location only when she is on campus *and* connected to WiFi, we found users are trackable only for 65% of the 9am-6pm daily window–i.e., approx. 378 mins. Moreover, the total time spent by users at *on-campus stay locations* is around 55% (207 mins), with the remaining 10.1% spent in transient intervals.



(a) Interruptible Window **(b) Distance Distribution**

Figure 2: (a) Percentage of tasks accepted while the user attending class as a function of time elapsed since the beginning of the class, and (b) distribution of the distance between task and user location (i.e., classroom) measured in minutes

We find that, on average, 73.8% of the tasks are accepted while users are on a *stay episode* (which represents only 54.9% of the typical workday). This indicates that user propensity to accept tasks is higher at such "stay locations". Moreover, Fig. 1b depicts the percentage of tasks accepted (across all users) as a function of the total length of the corresponding stay episode. We see that the task acceptance rate is positively correlated with the length of the stay duration–indeed, a separate regression confirms this (correlation coefficient=0.72 with p-value < 0.002). The analysis suggests that, at least in a campus, users tend to browse for and accept tasks at places such as classrooms, cafeterias and libraries, where they are stationary for longer periods of time.

2.2.3 Interruptible Time Window. Given the observed worker preference for accepting tasks when one is stationary for longer intervals, the next question is: *within such stationary episodes (when a worker is likely to be otherwise occupied), are there time periods during which users seem more interruptible?* We know that wrongly timed interruptions on mobile Apps create a negative perception among users, and can harm such App adoption and usage. To discover possible opportune moments for such interruptions, we focus specifically on the "in-class" stay episodes, as these are usually long, predictable and constitute an appreciable part of students' daily campus routines: on average, a student spends 12 – 17.5 hours per week (25-40% of their on-campus stay) attending classes.

To address this question, for each accepted task, we first extract a user's current location (from historical mobility traces) as a tuple $\langle u, l, p, t \rangle$, where u is the user ID, l is the current location ID of the user, p is the accepted task location, and t is time stamp. We then match the the academic timetable (which specifies the (location, start time, duration) of each class) with the user's spatio-temporal movement pattern to determine whether she is attending a lecture.

Fig. 2a illustrates the percentage of tasks accepted (i.e., over all the tasks that whose task acceptance location is a classroom) as the function of the elapsed class time (X-axis). Note that more than 75% of in-classroom task acceptances occur within the first 30 minutes. Intuitively, we conjecture that users remain more distracted (and possibly less engaged) during the initial period of a class, and progressively get more involved in their academic work as the session progresses. In addition, Fig. 2b plots the distribution of the walking distance between the task location and the task acceptance location (i.e., the classroom). We can see that, for 60% of such tasks, the task location is in the same building (walking distance ≤ 5mins) as the classroom, indicating in-classroom users behave opportunistically, selecting tasks that are near their current location.

Key Takeaway: The studies on *TASKer* data clearly reveal that (a) workers tend to disproportionately accept tasks when they are stationary, and that accepted tasks are often very near such staypoints, and (b) at staypoints, workers tend to accept tasks earlyThese observations lead to the research question that we investigate next: *"can smart task notification strategies be used to modify or leverage on these behavioral patterns?"*.

3 SYSTEM OVERVIEW

Given the task acceptance dynamics investigated in the previous section, we designed and deployed *PokeMe*, a context-driven notification engine that reminds workers of tasks in the base *TASKer* crowd-sourcing platform. Figure 3 illustrates the overall functional architecture of *TASKer*, which includes a key new component: PokeMe *Engine*, that determines *when*, *where* and *to whom* such task reminder notifications are sent.

3.1 Implementation Details

The implemented *TASKer* system consists of three components: (a) a web interface for task creation, (b) a server and a database for storing tasks and responses, and (c) a client mobile application for crowd-workers. The client application (available on both Android and iOS) shows workers various available tasks (e.g., report the cleanliness of the toilet in level 2 of building *A*), and allows them to select and execute such location-specific tasks.

For operational reasons, *TASKer* generates tasks across 3 distinct *time windows*: 9am-12noon, 12noon-3pm and 3pm-6pm, with the *TASKer* App only showing tasks valid within the current time window. Moreover, each task is associated with a variable *execution interval* (T_s, T_e) (where T_s and T_e denote the start and end time instants of the interval), such that the task can only be performed within this execution interval.

3.2 PokeME Task Reminder Engine

The *PokeMe* notification engine is used to implement a variety of context-driven reminder notification strategies. Based on past results [30] that demonstrate the harmful effects of excessive notifications, *PokeMe* is configured to ensure that such reminders are infrequent–users never receive more than 2 such notifications in a single day (the 9am-6pm window), with a maximum of only 1

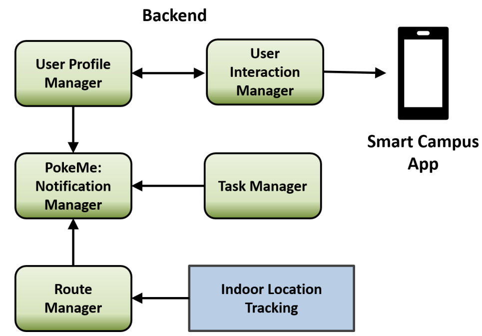


Figure 3: *TASKer* framework - architecture

notification within a single *time window*. For our research, we configured *PokeMe* to target workers with several distinct notification strategies[5]:

Strategy S0 (No notification): This is the default mode and it serves to establish the baseline behavior, against which other notification strategies are compared. (In the user study, this strategy is applied during weeks 1 & 2.)

Strategy S1 (Random): In this case, notifications are fired at *random* time instants during the day, reminding users of available tasks. For a chosen time window, *PokeMe* will randomly pick a time instant and fire a notification saying "You have new tasks nearby to accept and perform!". However, if the user has exhausted his task quota, then this notification will not be generated.

Strategy S2 (Low Popularity): This strategy is meant to counter the observed low rates of task acceptance at certain locations (Section § 2). In this case, the *PokeMe* engine maintains a list of locations whose $TaskAcceptanceRate$ is significantly lower than that of other locations. Subsequently, if *PokeMe* detects a user at such a location, it fires a notification, alerting the user to the availability of tasks. Such reminders are served on a first-come-first-serve basis, while adhering to the twice-a-day and once-a-time window limits on notifications.

Strategy S3 (Start-Staypoint): These notifications are fired when the *PokeMe* engine detects a user in a classroom during a scheduled class (corroborated via the SMU timetable). This strategy is based on the previously observed patterns of preferential acceptance of tasks *during the initial part of a stay period.* More specifically, the notification is generated at a random time instant, uniformly distributed over the first 15 minutes, when attending a class.

Note that upon clicking on the notification in *S1 – S3*, the user will be redirected to the *TASKer* App and shown all the available tasks in the same building.

3.3 Experiment Study Details

To study the effectiveness of such smart notification strategies, we utilized a deployment of *TASKer*, integrated with the additional *PokeMe* engine, over a 6 week trial period, October 2 - November 20, 2017 (with tasks being available only on working weekdays). During this period, a total of 432 students opted to participate in the study (which had been previously approved by the university's Institutional Review Board (IRB)). During this period, the *TASKer*

[5]Compared to prior work on notifications for online tasks, our notifications are novel as they utilize spatio-temporal properties of **both users and tasks**.

platform provided a total of 37,600 distinct tasks, with a worker permitted to execute at most 10 tasks in any 3-hour *time window.* To ensure a more uniform task completion rate, each user is allowed to perform at most 5 tasks in first and second hour segments of the time window, with any residual tasks (up to 10) being allowed in the third hourly segment.

Table. 1 shows the total number of *PokeMe* workers involved – both registered and active workers and the number of responses received, on weekly basis. To establish a *baseline*, notifications were disabled in the first two weeks of the study–i.e., strategy *S0* was deployed across all users. For the subsequent four weeks, the user pool was divided into 4 equal-sized groups (a *between-subjects* evaluation methodology), with a specific group being subject to one of the four treatment strategies *S0-S3.*

More specifically, we tackle the following research questions:

- Are there tangible benefits from task reminders, even if they are fired at possibly inopportune moments?–i.e., do such notifications truly increase task acceptance rates, and/or reduce the risk of *task dereliction*–i.e., the likelihood that users will not complete accepted tasks ?
- Are there observable changes in worker behavior when workers are subject to such task reminder notifications? In particular, do the intervention strategies result in changes to (a) *when* and *where* users usually accept or execute such physical-world tasks, and (b) the time taken to accept and perform tasks?

Table 1: Summary of user details.

Week	No.Registered users	No.Active users	Responses received
1 (Baseline)	350	246	8971
2 (Baseline)	362	125	3506
3 (PokeME reminders)	382	188	5888
4 (PokeME reminders)	408	185	7472
5 (PokeME reminders)	421	177	6583
6 (PokeME reminders)	432	155	5738

4 EVALUATION

In this section, we empirically evaluate the efficacy of *PokeMe*, which decides when, where and whom to remind about available tasks by measuring the following:
(1) Reaction modality in perceiving a notification: A task reminder can alert the user by means of flashing (the screen lit and shows the message content as a push notification), sound and vibration. In this study, we study different reaction modalities such as (a) whether the user clicked/dismissed a notification, and/or (b) whether he reacted to the notification message by accepting tasks shortly after notification receipt. *(2) Change in overall task acceptance and dereliction rate:* We see whether task reminders improve the task acceptance and/or reduce the dereliction rate, specially at locations that seem least popular? *(3) Change in task acceptance behavior:* We study aspects such as (a) reaction time – time taken to accept and complete a task since the receipt of a notification, and (b) whether notifications alter the patterns/locations of task acceptance.

4.1 Reaction Modality in Perceiving Notification

To study the different reaction modalities, we conducted an additional post-trial survey, receiving responses from 206 of the 432 users to the following question: *How do you normally check or read your notifications in* TASKer *App?*

We presented different reaction modalities – (a) click on the notifications and then accept tasks from the window to which the notification redirects, (b) read the content from the notification bar and then manually open the App manually to look for tasks, and (c) ignore the notifications–i.e., neither click on it nor accept tasks. In contrast to our belief, there's a significantly large group of users (more than 40% of the respondents) who react by manually opening the App to accept tasks after reading the notification bar. This suggests that measuring 'notification clicks' is a misleading measure of user reaction. Accordingly, we look at the task *acceptance history* and measure a notification's effectiveness by the tasks accepted within a certain threshold (15 minutes) of the notification's delivery time (tabulated in Table. 2. We see that, across all 3 strategies, the number of such task acceptances was comparable to the total number of notifications, indicating the effectiveness of such notifications.

4.2 Overall Task Acceptance Rate

Prior studies in mobile interruptibility [25] have shown, albeit for online applications, that judicious use of context-aware notifications can improve user response time and notification acceptance. From Table 2, we see a similar behavior for physical crowd-sourcing tasks: *task reminders did improve the acceptance rate by 22%* (with a three-fold reduction in task dereliction rate) – even for the random test group. Somewhat surprisingly, the overall acceptance rate is similar across all 3 strategies. This begets the question: "are these task reminders successful because they (almost magically) were delivered at opportune, interruptible moments?". Because we do not have access to ground-truth of additional physical context (such as the user's activity or emotional state), we rely on the following question posed during the post-study survey: *How occupied you were when you received the notifications?.* The survey responses show that majority of the respondents (more than 70% of them comprising of 23%, 25% and 22% from strategies S1, S2 and S3, respectively) felt that the notifications were received while they are already preoccupied or busy, hinting that the notifications did *not* reach the users during opportune moments. This qualitative-finding contradicts with our original intuition – i.e., the task reminders were successful as they were delivered during the most opportune moments. This is likely due to the fact that the reminders we sent caught user attentions (regardless of their context) and subsequently led to task acceptance.

4.3 Acceptance Popularity

Our "low popularity" reminder strategy is specifically designed to improve the low task acceptance popularity at certain historically under-performing locations. This strategy did have the desired effect, as evidenced by Fig 4, which plots the weekly task acceptance at the 50-least popular locations, before and after the introduction of notifications. The *X*-axis is presented in ascending order of task

Table 2: Overall statistics of task reminder strategies

Strategy	Number of users	Notifications sent	Notifications clicked	Tasks accepted in 15 minutes	Tasks accepted in 30 minutes	Daily task acceptance rate	Overall task failure rate
No notification	82	–	–	–	–	2.44%	17.42%
Random	115	726	123	531	616	23.76%	6.84%
Low Popularity	114	808	117	510	605	24.01%	6.97%
Start Staypoint	121	729	129	432	476	19.94%	7.10%

acceptance rate after notifications. We see an overall average improvement of over 12% in task acceptance across these 50 least popular locations, suggesting that *notifications can indeed be a means of overcoming well-known spatial skews in mobile crowd-sourcing.*

Figure 4: Task acceptance rate of low popularity locations

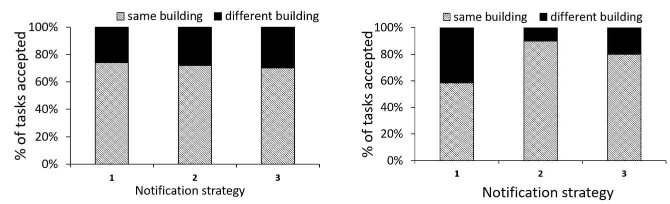
4.4 Behavioral Insights

We now study the worker responses to *PokeMe* task reminders in greater detail, to especially understand why task reminders work and how user behavior differs across the three notification groups (S1–S3). For all studies conducted in this section, we use weeks 1-2 (without notifications) as the baseline and weeks 4-6 as the treatment outcomes[6].

4.4.1 Notifications and user's search range. We first observe differences in the spatial range within which users look for tasks, in response to different notification strategies. For simplicity, we consider a binary range: whether the user's task acceptance and the task location are both in the same building or not. As illustrated in Fig. 5a, during the no-notification baseline period, majority of tasks (around 82%) have the task acceptance and task locations within the same building. This confirms the the tendency of users to pick only nearby tasks in the absence of any stimulus.

This pattern changes significantly in both directions after we introduce notifications. As illustrated in Fig. 5b, for users receiving random notifications (the S1 group), 42% of all tasks are now accepted while users are in different buildings. No changes were observed in group S0. This suggests that random reminders have enabled users to favorably consider tasks that are not located in their current building of residence, but might be compatible with their future schedules. In contrast, for the S2 and S3 reminder groups, the proportion of same-building task acceptance actually

[6]Week 3 is excluded, as it coincides with the university's mid-term break



(a) Before notification

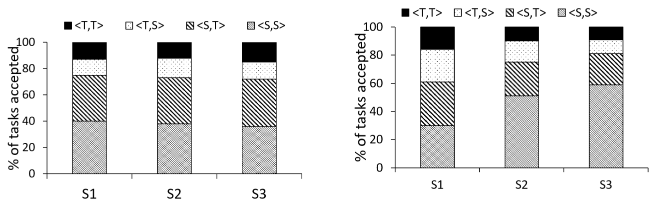
(b) After notification

Figure 5: Range of the accepted tasks when notifications were (a) absent vs. (b) present

increases (compared to the no-notification baseline)–by 18% and 10% respectively. While the *PokeMe* App is designed to initially show same-building tasks for strategies S2 and S3 if users click on the notification, note that the majority of users actually manually open the App in response to notifications (in which case they can see tasks across all buildings). Clearly, such users still overwhelmingly prefer same-building tasks even when they see the complete task listing.

We hypothesize that this preference may be due to the longer stay times (relative to the 3-hour task performance window) associated with notification receipt by both S2 and S3 users. In particular, S2 users often receive notifications while they engage in group study sessions (as group study rooms are often low-popularity locations), while S3 users receive notifications while attending classes. In either case, such users continue to remain at their current location for a significant time after notification receipt. More specifically, analysis of residency times show that 64% of the S2 users continue to engage in group study sessions for at least 35 minutes beyond the time instant of notification receipt, while S3 users are observed to remain in the class for 3 additional hours after the notification time.

4.4.2 Mobility patterns and task acceptance. In this section, we look at the impact of notification strategies on *where* users accept tasks. For each task accepted by a user, we label both the locations of task acceptance and task performance as either a *stay location* or a *transient location*. The determination of the stay and transient locations are user-specific. Our analyses are illustrated in Fig. 6, where we use the tuple $< A, B >$ to denote the mobility pattern: A and B denote task acceptance and performance location, respectively – e.g., $< S, T >$ represents that tasks are accepted at stay locations and performed at transient locations. From Fig. 6a, we can see that users are *accepting* tasks mostly at their stay locations (including both $< S, S >$ and $< S, T >$), amounting to close to 75% of all tasks accepted for all test groups. The users also prefer task locations that belong to one of their stay locations (including both $< S, S >$



(a) Before notification **(b) After notification**

Figure 6: Mobility patterns and notifications

and $< T, S >$), amounting to close to 60% of all tasks *performed* for all test groups.

These patterns change significantly after we introduce notifications. From Fig. 6b, we can see that different notification strategies produce very different responses in task acceptance patterns. For random notification group (S1), users are now more likely to accept tasks while on the move (the combined ratio of $< S, S >$ and $< S, T >$ is now only around 57%, down from around 75%), and choose tasks that are at their transit locations (the combined ratio of $< T, S >$ and $< T, T >$ is now around 43%). However for S2 and S3 users, notifications actually reinforce their tendency to accept and perform tasks at stay locations. For task acceptance at stay locations, the ratios are now 75% and 79% respectively, for S2 and S3. For task performance at stay locations, the ratios are now above 70% for both S2 and S3 groups.

The observations highlight the different influences of randomized vs. context-dependent locations in *TASKer* : (a) S1 promotes greater task completion of tasks in transitory areas, (b) S2 helps to increase the *spatial fairness* of task execution (by causing workers to choose tasks from less popular task locations), whereas (c) S3 help to increase the *overall task execution rate* (by increasing task acceptance), but not materially enhance fairness.

5 CONTEXTUAL FACTORS AND TASK REMINDERS

Studies in the previous section show that our 3 different notification strategies (S1-S3) elicit a user response around 60-70% of the time. Given the resulting notification response data, we now investigate (*post-hoc*) whether it is possible to identify additional *novel contextual factors* that help build a more accurate model of notification responsiveness. Building such a predictive model will allow us to deliver even more effective notifications for future mobile crowd-sourcing applications.

For this specific study, we considered only the "random" group (S1) data, as such randomized treatment allows us to observe worker response across diverse context states Prior work (e.g., [25]) indicates that interruptibility for *online* mobile applications is influenced by various features pertaining to the characteristics of the notification, mobile application and physical-context of the user (e.g., activity, engagement and emotions). To include the specific characteristics of mobile crowd-sourcing, we adopt a set of features from 3 broader classes: (a) *Notification Related (N)*–e.g., the user chooses to react to a notification and accept tasks, because the task location is highly popular; (b) *Task Related (T)*–e.g., the user does not accept any notifications for tasks that have higher execution complexity; and (c) *User Related (U)*–e.g., a user accepts a notification and tasks because she frequently visits the task location. Table 3 lists the complete set of features that we considered as *possible* determinants of notification responsiveness.

Notification arrival time (N): This metric captures the time at which the notification is delivered to the user, expressed as the minutes elapsed since start-of-the-day (9am in our case).

Overall temporal location popularity (N): This per-location, hourly metric (a 24-element vector) aims to quantify the popularity of locations. To derive it, we use the number of hourly visitations to a particular location and normalize the counts over all locations in the same hourly period.

Semantics score (N): This metric expresses the semantic diversity of a location, as a function of the spaces in its neighborhood. This score is computed by dividing the number of locations with similar semantics by the number of different semantics observed within a 3-hop radius of the location. Note that we adopted a simple labeling mechanism where each location is assigned a label (e.g., classrooms, restaurant) based on the location's primary function.

Closeness of task location: This task-intrinsic feature measures the distance between task and user location (i.e., the likely detour overheads), and hence measures user convenience.

Task complexity: In *TASKer*, the reporting tasks have 4 different response types or modes: multiple choice, binary (yes/no), free-text and photo tasks. The complexity of each type is measured by the ratio between the time spent on tasks of that type and the total time spent over all types of tasks in a day. This daily proportion is then averaged over the number of days in baseline period (i.e., weeks 1 and 2).

No. of tasks available: This measures the number of tasks available, at the instant when a notification arrives, at the user location.

User's remaining task quota: In *TASKer*, each user is allowed to complete 10 tasks in a 3-hour time window. This metric keeps track of the remaining task quota that a user still has at the moment of receiving a notification.

Per-user temporal location popularity: This metric defines an hourly temporal profile of a user. Each location i can be represented as a 24-element vector, with each element represents the proportion of the hourly visitations at location i, as compared to total visitations of the user (across all locations) during the same hourly period.

Stay-time duration: Motivated by our earlier finding in §2.2.2, we quantify the stay-time duration as follows: given a series of stay episodes of a user j in a given day d, we compute the ratio r^d_{ji} between the length of stay episode s^d_{ji} at location i and total length of all the stay episodes combined ($\sum_{\forall i} s^d_{ji}$).

Given these features, we utilize a logistic regression model to predict (as a binary outcome variable) a user's receptivity towards a task reminder. The classifier was trained with from the first 2 out of the last 3 weeks (in which notifications were present), while the last week's data was used as the test set. Table 3 tabulates the coefficients of the logistic regression. For brevity, the table lists only the features that turned out to be significant. To measure the accuracy, we compute AUC scores of the combined classifier.The overall AUC score turns out to be 0.86, indicating that carefully chosen contextual features, even at aggregate level, help to accurately predict a user's receptivity towards a task notification.

Table 3: Summary of the features.

Class	Features	Description
Notification	Notification arrival time	Time at which notification is received by the user
	Overall temporal location popularity	Normalized hourly user visitations of notification location
	Semantics score	A diversity metric that measures the similarity of the location as compared to its neighborhood
Task	Closeness of task location	Distance between the task and notification location
	Task complexity	Complexity of the task
	No. of tasks available	Number of tasks available at user's current location
User	User's task quota	Remaining daily task quota as he receives the notification
	Per-user temporal location popularity	Hourly visitation of a specific user
	Stay-time duration	Proportion of residency time of a location over total stay time

Table 4: Coefficients of the logistic regression.

Feature	Coefficient	Std. Err	95% CI	user 1	user 2	user 3	user 4	user 5
Closeness to task location	-0.38^{***}	0.11	(-0.59, -0.16)	-0.2^{**}	-0.12	-0.24	-0.15	-0.09^{*}
Task complexity	-0.2^{**}	0.07	(-0.34, -0.06)	-0.01	-0.003	-0.03^{*}	-0.1	-0.09
Overall temporal location popularity	0.32^{***}	0.16	(0.006, 0.63)	0.14^{*}	0.21^{***}	0.01	0.12^{**}	0.09
Per-user temporal location popularity	0.222^{**}	0.10	(0.027, 0.418)	0.16	0.19	0.18^{**}	0.05	0.07^{**}
No. of tasks available	0.03^{*}	0.01	(-0.05, -0.01)	0.24^{*}	0.28^{**}	0.12	0.15^{**}	0.22
Semantics score	-0.043^{*}	0.22	(-0.47, 0.38)	-0.0005	-0.019	0.022	0.001	-0.01^{**}
Stay-time duration	0.5^{***}	0.14	(0.23, 0.77)	0.09	0.28^{***}	0.04^{**}	0.82	0.1

*p<0.1; **p<0.05; ***p<0.01

We make the following key observations: (1) We see that, if all the other features are kept constant, we will see a 37% increase in the odds of accepting a task following a notification for a one unit increase in overall temporal location popularity. (2) Similarly, we see 24.8%, 4.2% and 64.8% increase in the odds of accepting a task following a notification for a one unit increase in per-user temporal location popularity, semantics score and stay-time duration, respectively.

These insights (albeit from a campus-based study) reveal several novel contextual factors that can be incorporated into impactful notification strategies for mobile crowd-sourcing applications. For example, such notifications are particularly effective (in terms of increasing user task acceptance) when sent to users at staypoint locations, or for tasks located at places frequently visited by a worker. Of course, a notification strategy may well need to balance such response maximization with other objectives (e.g., S2, which attempts to reduce the spatial skew of completed tasks.).

To further understand the influence of individual features, we run logistic regression for each features separately.

Personalized Notification Strategies: The study reported in §5 considers an aggregate model to identify the features that influence a user's receptivity to reminder notifications. After witnessing the strong influence of user-related features (e.g., user location popularity), it is legitimate to ask whether a notification strategy should be personalized–i.e., defined on per-user basis. For example, user A may preferentially accept tasks in the morning while commuting to work, while user B may choose to perform tasks while on his way back home in the evening. These user-dynamics will be often neutralized in aggregate user modeling.

To investigate the potential of per-user preference modeling, we conducted an additional, albeit preliminary, analysis: we utilize the features reported in §5 and use logistic regression to build per-user context model. For brevity, we present the coefficients (in Table 4) for 5 randomly chosen users (out of the top-20 users in the random group) who frequently reacted to notifications. We can see that a handful of features consistently influence the users' receptivity to reminders; moreover, the average AUC score (across these 5 users) = 0.81, indicating that such a "person-specific context-driven notification strategy" may indeed further improve worker responsiveness.

6 DISCUSSION

Our work identifies several key new contextual factors that impact worker responsiveness to notifications, and demonstrates that such context-aware notifications can significantly increase worker engagement. There are, however, many other facets to explore.

Interplay of Intervention Strategies: We showed that a *low-popularity* notification strategy (strategy *S*2) helps to significantly reduce the spatial skew in task performance. Past work has proposed using differential task rewards [12], with tasks in less popular locations being allocated proportionally higher rewards, to tackle such skews. Our work now demonstrates that notifications can offer a less-expensive alternative to address such skews: in particular, in *PokeMe*, we achieved a task completion fairness (Jain's index) of 0.24 with an average per-task payout of $0.25, in contrast to corresponding values of 0.36 fairness and $0.82 per-task payout under a non-uniform pricing strategy. It will be interesting to develop a technique that combines such differential rewards and notifications to optimally impact the overall task completion rate.

Extrapolating to City-scale Crowd-sourcing: The campus-based studies with *PokeMe* arguably have unique characteristics, such as young demographics and work/study-driven periodic patterns of interaction & movement. It is interesting to speculate if our observations on notification impact will apply to more general *city-scale* settings. A definitive answer requires large-scale experimentation, which has significant practical challenges. However, initial data from our deployment of a city-scale trial called *Smart City* [7] suggest that several of our observations may be applicable more broadly. Fig. 7 depict the diurnal change in average task acceptance rate, observed over a trial period of 9 months. Each data-point represents the task acceptance rate (in y-axis) within each corresponding hour-segment (in x-axis) – the "peaks" we observed are marked in red boxes. We find that more tasks were accepted during three time windows: 8am-9am, 11am-1pm, and 5pm-7pm, which seem to align well with the hypothesis users tend to accept tasks while (or shortly after/before) they commute to/from work or during lunch-hours. In addition, via continuous tracking of worker city-wide trajectories, we observe that 75.2% of tasks are accepted by commuters at stay locations – note that stay locations comprised, on average, 72% of the total working day. This corroborates the preferential acceptance of tasks in *PokeMe* at campus-specific stay locations.

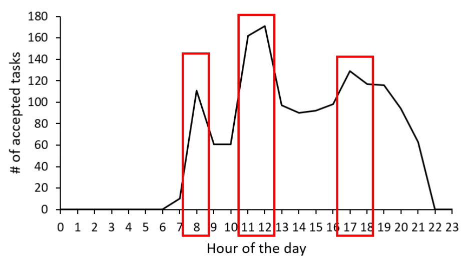


Figure 7: Hourly task acceptance rate across a day

7 RELATED WORK

In this section, we review related work that (a) focuses on modeling mobile interruptibility and (b) develops strategies to improve task completion in a physical crowd-tasking environment, and (c) attempts to improve user engagement for *online* crowd-sourcing.

7.1 Model User Interruptibility

A significant body of research has recently explore various aspects of users' interaction with mobile notifications [1, 3, 4, 7, 22, 26, 27, 30]. In particular, prior works show that users act upon the receipt of notifications promptly [30] based on the importance of the applications that triggered them [26]. Similarly, the authors in [15] and [22] point out that users are willing to tolerate some disruption in return for receiving notifications with valuable information. Through a series of works [20, 21, 25] and in [10], the authors show that the attentiveness of users can be determined by contextual factors In [28], authors infer boredom by using people's phone usage data, and exploit it for triggering proactive recommendations. In a more recent study [24], the authors identify the breakpoints between 2 physical activities by using mobile and wearable devices, and suggest that such breakpoints represent opportune, interruptible moments. However, this entire body of work investigates the use of notifications with *online* applications, and not on how such notifications impact physical-world crowd-sourcing tasks.

[7]Trials underway with over 1800 residents in a large Asian city; details withheld for anonymity.

7.2 Task Compliance in Mobile Crowdsourcing

Mobile crowd-sourcing platforms often face challenges in ensuring greater task completion and sustainable worker engagement. Prior works [16, 23] have explored the challenges in mobile crowdsourcing. Various approaches have been proposed to tackle these issues. Authors in [19] and [32] suggested providing recommended action items to workers. Authors in [11] showed that proactively providing trajectory-aware recommendations improved user engagement and task completion rates, while Kim et al. [18] employ proximity-based notifications to alert the workers. The system reported in [13] used explicit feedback as a means to improve user motivation. Authors in [12] and [29] proposed differential rewards or lotteries to incentivize task acceptance.

7.3 User Engagement in Online Crowdsourcing

Similar to the challenges faced in the mobile world, the online crowd-sourcing counterpart has also focused on improving sustained participation – several studies have focused on factors that influence volunteerism and retention [31]. Prior works [2, 5, 6, 14] have explored various approaches – in [5], Baruch et al. tackled the issue of lack of volunteerism by conducting user opinion surveys from online crowd-workers of Tomnod platform, while [14] studies various *intrinsic* and *extrinsic* motivators that would possibly induce users to participate in crowd-tasks [14]. Similarly, social gamification strategies coupled with various contextual factors have been proposed in [2] to boost the user engagement in Enterprise crowdsourcing platforms, while context-aware nudging techniques have been proposed in [6] to improve the task commitment. Our work focuses instead on *physical world* crowd-sourcing tasks and identifies novel features that influence worker responsiveness to notifications for such tasks.

8 CONCLUSIONS AND FUTURE WORK

Overall, our work demonstrates the important role that contextualized notification can play even in mobile crowdsorucing applications (where users have to physically move to execute tasks) and the possible tradeoffs (between overall task execution rates and fairness) that different notification mechanisms generate. More importantly, we identified several novel contextual features, spanning three distinct features classes and showed that these features can help predict (albeit via a *post-hoc* analysis) user responsiveness to notifications with reasonably high accuracy (AUC = 0.86).

In future work, we shall extend the study of context-aware notification mechanisms to city-scale crowdsourcing applications, as well as study the role of additional contextual activities (e.g., group interaction effects or physical activity) in shaping user interaction with such notifications.

9 ACKNOWLEDGMENT

This material is supported by the National Research Foundation, Prime Ministers Office, Singapore under its International Research Centres in Singapore Funding Initiative.

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 *CHI'04*. ACM, 271–278.

[2] G. Afentoulidis, Z. SzlÃąvik, J. Yang, and A. Bozzon. 2018. Social Gamification in Enterprise Crowdsourcing. In *International Conference on Web Science (WebSci)*.

[3] Brian P Bailey and Joseph A Konstan. 2006. On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in Human Behavior* 22, 4 (2006), 685–708.

[4] Brian P Bailey, Joseph A Konstan, and John V Carlis. 2001. The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In *INTERACT'01*. IEEE, 593–601.

[5] A. Baruch, A. May, and D. Yu. 2016. The motivations, enablers and barriers for voluntary participation in an online crowdsourcing platform. *Computers in Human Behavior* (2016).

[6] S. Bashirieh, S. Mesbah, J. Redi, S. SzlÃąvik, and R. Jan Sips. 2017. Nudge your Workforce. A Study on the Effects of Task Notification Strategies in Enterprise Mobile Crowdsourcing. In *International Conference on User Modelling, Adaption and Personalisation (UMAP)*.

[7] Edward Cutrell, Mary Czerwinski, and Eric Horvitz. 2001. Notification, disruption, and memory: Effects of messaging interruptions on memory and performance. In *Interact'01*. IOS Press, 263–269.

[8] Mary Czerwinski, Edward Cutrell, and Eric Horvitz. 2000. Instant Messaging: Effects of Relevance and Timing. In *HUMAN-COMPUTER INTERACTION*. 71–76.

[9] Fiorella De Cindio. 2009. Moments and modes for triggering civic participation at the urban level. In *Handbook of research on urban informatics: The practice and promise of the real-time city*. IGI Global, 97–113.

[10] 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 *MobileHCI'10*. ACM, 103–112.

[11] Anonymized for Double Blind Review. [n.d.].

[12] Anonymized for Double Blind Review. [n.d.].

[13] Jorge Goncalves, Simo Hosio, Jakob Rogstadius, Evangelos Karapanos, and Vassilis Kostakos. 2015. Motivating Participation and Improving Quality of Contribution in Ubiquitous Crowdsourcing. *Comput. Netw.* 90, C (Oct. 2015).

[14] M. Hosain. 2012. Users' Motivation to Participate in Online Crowdsourcing Platforms . In *International Conference on Innovation, Management and Technology Research (ICIMTR)*.

[15] Shamsi T Iqbal and Eric Horvitz. 2010. Notifications and awareness: a field study of alert usage and preferences. In *CSCW'10*. ACM, 27–30.

[16] Brent Hecht Jacob Thebault-Spieker, Loren Terveen. 2015. Avoiding the South Side and the Suburbs: The Geography of Mobile Crowdsourcing Markets *(CSCW '15)*. 265–275.

[17] Kasthuri Jayarajah, Youngki Lee, Archan Misra, and Rajesh Krishna Balan. 2015. Need accurate user behaviour? Pay attention to groups!. In *2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*.

[18] Y. Kim, E. Harburg, S. Azria, A. Shaw, E. Gerber, D. Gergle, and H. Zhang. 2016. Studying the Effects of Task Notification Policies on Participation and Outcomes in On-the-go Crowdsourcing. In *2016 Association for the Advancement of Artificial Intelligence (AAAI)*.

[19] Nicolas Kokkalis, Thomas Köhn, Johannes Huebner, Moontae Lee, Florian Schulze, and Scott R Klemmer. 2013. TaskGenies: Automatically Providing Action Plans Helps People Complete Tasks. *ACM Transactions on Computer-Human Interaction* 20, 5 (2013), 27.

[20] Abhinav Mehrotra, Robert Hendley, and Mirco Musolesi. 2016. PrefMiner: Mining User's Preferences for Intelligent Mobile Notification Management. In *Proceedings of UbiComp'16*. ACM, Heidelberg, Germany, 1223–1234.

[21] Abhinav Mehrotra, Mirco Musolesi, Robert Hendley, and Veljko Pejovic. 2015. Designing Content-driven Intelligent Notification Mechanisms for Mobile Applications. In *UbiComp'15*. ACM, 813–824.

[22] Abhinav Mehrotra, Veljko Pejovic, Jo Vermeulen, Robert Hendley, and Mirco Musolesi. 2016. My Phone and Me: Understanding People's Receptivity to Mobile Notifications. In *CHI'16*. ACM, 1021–1032.

[23] M. Mushthag and D. Ganesan. 2012. The role of superagents in mobile crowdsourcing. *Workshops at AAAI conference on Human Computation and Crowdsourcing* (2012).

[24] 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 *2017 IEEE International Conference on Pervasive Computing and Communications, PerCom 2017, Kona, Big Island, HI, USA, March 13-17, 2017*. 100–110.

[25] Veljko Pejovic and Mirco Musolesi. 2014. InterruptMe: designing intelligent prompting mechanisms for pervasive applications. In *UbiComp'14*. ACM, 897–908.

[26] Martin Pielot, Karen Church, and Rodrigo de Oliveira. 2014. An in-situ study of mobile phone notifications. In *MobileHCI'14*. ACM, 233–242.

[27] Martin Pielot, Rodrigo de Oliveira, Haewoon Kwak, and Nuria Oliver. 2014. Didn't you see my message?: predicting attentiveness to mobile instant messages. In *CHI'14*. ACM, 3319–3328.

[28] Martin Pielot, Tilman Dingler, Jose San Pedro, and Nuria Oliver. 2015. When attention is not scarce-detecting boredom from mobile phone usage. In *UbiComp'15*. ACM, 825–836.

[29] John P. Rula, Vishnu Navda, Fabián E. Bustamante, Ranjita Bhagwan, and Saikat Guha. 2014. No "One-size Fits All": Towards a Principled Approach for Incentives in Mobile Crowdsourcing. In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications (HotMobile '14)*.

[30] Alireza Sahami Shirazi, Niels Henze, Tilman Dingler, Martin Pielot, Dominik Weber, and Albrecht Schmidt. 2014. Large-scale assessment of mobile notifications. In *CHI'14*. ACM, 3055–3064.

[31] S. A. Sheppard, J. Turner, J. Thebault-Spieker, H. Zhu, and L. Terveen. 2017. Never Too Old, Cold or Dry to Watch the Sky: A Survival Analysis of Citizen Science Volunteerism. *ACM Human Computer Interaction* (2017).

[32] Kartik Talamadupula, Subbarao Kambhampati, Yuheng Hu, Tuan Anh Nguyen, and Hankz Hankui Zhuo. 2013. Herding the crowd: Automated planning for crowdsourced planning. In *1st AAAI Conference on Human Computation and Crowdsourcing*. 70–71.