Open access • Book Chapter • DOI:10.1007/978-3-319-24195-1_6

Push or Delay? Decomposing Smartphone Notification Response Behaviour

— Source link

Liam D. Turner, Stuart M. Allen, Roger M. Whitaker

Institutions: Cardiff University

Published on: 08 Sep 2015

Topics: Context awareness

Push or Delay? Decomposing Smartphone Notification Response Behaviour

Liam D Turner, Stuart M Allen, and Roger M Whitaker

{TurnerL9,AllenSM,WhitakerRM}@cardiff.ac.uk
Cardiff School of Computer Science & Informatics, Cardiff University, Cardiff, UK

Abstract. Smartphone notifications are often delivered without considering user interruptibility, potentially causing frustration for the recipient. Therefore research in this area has concerned finding contexts where interruptions are better received. The typical convention for monitoring interruption behaviour assumes binary actions, where a response is either completed or not at all. However, in reality a user may partially respond to an interruption, such as reacting to an audible alert or exploring which application caused it. Consequently we present a multi-step model of interruptibility that allows assessment of both partial and complete notification responses. Through a 6-month in-the-wild case study of 11,346 to-do list reminders from 93 users, we find support for reducing false-negative classification of interruptibility. Additionally, we find that different response behaviour is correlated with different contexts and that these behaviours are predictable with similar accuracy to complete responses.

Keywords: Interruptibility, smartphone notifications, interruptions, context awareness, implicit sampling, mobile

1 Introduction

Over the last decade the rise of the smartphone has had a profound effect on society, providing an ever-present opportunity for information retrieval and delivery. The *app* culture has extended the diversity of interruptions from phone calls and SMS messages to include *notifications* - snippets of information from diverse services, intended to inform or prompt reaction. Inappropriately timed interruptions are a fundamental issue, being at best an annoyance and at worst a dangerous distraction. Techniques are needed to enable services to determine and exploit interruptible moments, in order to deliver the right information at the right time.

Intelligent systems capable of predicting the success of individual notifications are highly desirable, yet this is dependent on the nature of the service [14]. For example, an appropriate time to prompt the user for a health related intervention is unlikely to be the same as one to notify them of a social network update. Similarly, the former may require them to undertake a specific and timely action (e.g. report their progress *in situ*), while the later is simply delivering information, and could safely be ignored. Currently, the delivery of notifications is largely at the interrupter's discretion, leaving the interruptee to reactively assess the appropriateness, or manually manage blanket-rules. The smartphone's ubiquity brings further opportunities, particularly with the evolving habitual role interruptions are having in our daily lives.

A fundamental issue in building intelligent interruption systems is to identify contexts where the benefit of interrupting outweighs the perceived cost on the user. Previous work has typically involved relying on the user to explicitly provide feedback after each interruption (e.g., [15, 18]), thereby classifying their interruptibility at that time. However, this creates a rigid *black-box* between delivery and feedback, leaving an outstanding issue of what to do when complete responses are not made. In these cases important information is being lost, for example, it may be that the user wasn't interrupted, or that they weren't physically interruptible to the extent that they changed focus, or that they were but didn't want to provide feedback. Identifying the attentiveness of users [15] through partial responses forms the focus of this study.

Previous interruptibility studies have identified that the abstract convention of an interruption and a response is a staged process of decision making and information exchange [12]. However, developing this in the context of smartphones has received little attention. Additionally, previous work has shown that smartphone interactions vary based on the level of focus available [17]. Therefore we hypothesise that, in comparison to a black-box approach, decomposing response behaviour to notifications will enable more cases of interruptibility to be observed and ultimately avoid misclassification. We present a model that decomposes response behaviour from notification delivery through to notification consumption. By developing and deploying a bespoke to-do list reminder app on Android smartphones, we find empirical evidence supporting this approach. In particular we are able to reduce misclassifications by separating out truly unsuccessful interruptions from partial responses where some degree of interruptibility was shown.

2 Current Conventions

The general convention for studying interruptibility is to issue interruptions under different contexts and see if a response is given. For smartphone notifications, this has the benefit of issuing interruptions through mechanisms that the user already adopts. This has typically involved explicitly interrupting the user to ask how interruptible they are, an arguably redundant practice if the user responds. Other studies attempt to implicitly operate through useful applications such as a reminder function in a Mood Diary [18]. However across these studies, each interruption attempt is typically represented in a similar way - as a feature representation of the current context and a label of the user's interruptibility given that context.

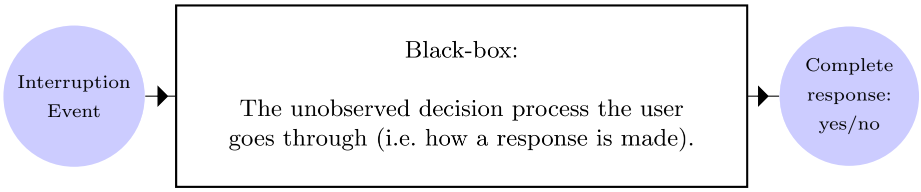

Interruption Event
Black-box:
The unobserved decision process the user goes through (i.e. how a response is made).
Complete response: yes/no

Fig. 1. The current underlying convention for determining interruptibility.

2.1 Representing the current context

The influential factors of interruptibility have been widely debated, particularly between: user characteristics (e.g., cognitive load), interruption properties (e.g., notification content) and the local environment (e.g., location or activity) [4, 11]. Additionally, the definition of what it means to be interruptible is also fragmented, with some studies focusing on: the physiological ability to switch focus (e.g., [2, 10]); the affect the interruption would have on the current task (e.g., [14, 7, 6]) or the user's sentiment towards the interruption (e.g., [15]).

The data sources sampled from also vary greatly in the literature, with peripheral hardware (e.g., wearable accelerometers [5]) historically used. The introduction of the smartphone has enabled many of these sensors to be contained within a single device that isn't alien to the user. Although smartphone sensors have some issues with accuracy and consistency [8], similar issues affect user annotation and bespoke equipment is impractical for large-scale, "in-the-wild", and longitudinal studies. Software APIs such as those tracking UI events [14, 1] have also been used, however these are often platform dependent or limited to moments when the user is performing a task on the device.

2.2 Labelling response behaviour

After initiating an interruption, studies across the literature typically judge success by either observing whether a response is made (e.g., [16]) or by requesting a self-assessment of its appropriateness (e.g., [15, 3, 19]). In either case, this resembles a *black-box* system that either succeeds or fails. However in reality, a user can exhibit a degree of interruptibility without completing the response. For example, if a response is started but abandoned, it may be the case that all notifications are unsuitable at this time, or only those from that application (e.g., all emails) or just the particular content in question (e.g., emails from the particular sender). If the user doesn't provide feedback or provides incomplete feedback, the *black-box* approach can't distinguish between these cases. For intelligent systems, this could lead to misclassifications of partial responses as null-responses, where no observable attempt to respond is made.

Retrieving a label Determining a quantified and unbiased measure of interruptibility remains an ongoing challenge. A common method for capturing

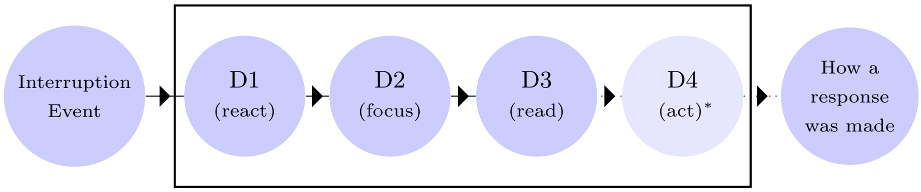


Fig. 2. Decomposing the decision process for Android notifications. *D4 may not apply to all notifications.

this has been through self-reports, also known as Experience Sampling Methods (ESM) (e.g., [15, 22]). On the positive side, due to the smartphone's ubiquity, this enables the collection of user opinion in situ, however it has several drawbacks for interruptibility studies. Firstly, it introduces an additional task that the user has to be sufficiently interruptible for and willing to complete. Secondly, it assumes that a user can accurately and consistently quantify their interruptibility, making it prone to errors across users and contexts [13]. Thirdly, it is also subject to potential behavioural bias from the consistent reminder that behaviour is being monitored [13].

An alternative is to use software events to implicitly capture indicators of interruptibility. For example, Smith et al [21] infer interruptibility by noting whether incoming phone calls are answered. This measures what the user does rather than what the user thinks, providing consistency over subjective self-reporting, but it is limited to externally observed behaviour.

3 Modelling a response as a decision process

To observe within the *black-box*, we decompose notification response behaviour into a sequence of atomic (possibly subconscious) decisions that a user makes. This enables us to examine the extent to which the response is pursued and avoid assumptions about the information the user knew. It is important to note that we are not aiming to extend or change the existing response process - we are trying to observe a process that already occurs. While this concept has previously been explored for other systems [12], we are unaware of its application explicitly to smartphone notifications. Additionally the variability of notifications [14] and operating system conventions presents a non-trivial task of creating a model robust to these inconsistencies. Within the scope of this study we focus on Android, due to it's market share and open APIs to observe decision behaviour.

From a prediction standpoint, being able to predict at least a partial response is useful for some applications. An example would be an application that delivers repetitive identical reminders. The user may not want to complete the process each time as the staged information delivery wouldn't change. Instead, just knowing that they acknowledged the interruption may be suitable.

3.1 Decomposing Android notifications

For Android notifications, user interface conventions and available APIs dictate a sequence of up to 4 atomic decisions (visualised in Figure 2) to be observed. A decision occurs at each point new information about the notification is provided to the user. After being presented with this information (e.g., the application icon), the user makes the decision to either continue (and be presented with the next piece of information) or terminate the response at that point (e.g., turn the screen back off and resume their previous activity). Inactivity with the device represents a failed attempt to interrupt the user, i.e. null-responses. The steps in our model are listed below, with the included examples assuming that the device is not-in-use when the notification is delivered:

D1 The process begins as the device attempts to gain the user's attention through sound, vibration or visual cues. Depending on the state of the device, this may go unnoticed (e.g. if the device is in a bag). However if the user is interrupted, they decide to either *react* and switch focus towards responding to the notification or ignore it.

D2 After choosing to react (e.g. turning the screen on), an icon graphic can be seen which indicates the notifying application (e.g. an email has arrived). Given this information, the user then decides to either *focus* their attention towards a content summary (e.g. the email subject) by accessing a notification drawer, or exit and return to their previous activity.

D3 On seeing the content summary, the user makes the decision whether or not to *read* a fuller message by consuming the notification and entering the relevant application (e.g. accessing the email client).

D4 Finally, if relevant to the notification, the user decides whether or not to *act* on the content, (e.g. send documents in reply to the email).

3.2 Model Generality

The stages visualised in Figure 2 represent the maximum observable decisions that a user goes through, however, this could vary due to notification inconsistency. Some decision steps can be obscured by properties such as a recognisable tone, such as the distinction between D1 and D2. Similarly, this may occur if the notification summary contains the complete content (e.g., a repetitive reminder) rather than dynamic meta-data (e.g., email sender). Additionally, when the device is not-in-use, the unlock process provides distinct points to observe decisions being made (Table 1). However, if the device is already unlocked (i.e., in-use), D1 and D2 cannot be easily distinguished due to limitations in observable UI events (e.g. accessing the notification drawer). However, the ability to observe some degree of partial response behaviour still offers improvement over relying on completed responses.

4 Case study: timely Android notifications

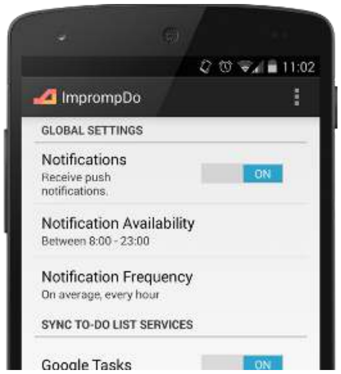
To determine the extent our decision model captures complete and partial responses to notifications, we collected data “in-the-wild”, using a bespoke Android application. From this we explore how many additional cases of interruptibility are captured in comparison to a *black-box* approach and identify correlations between the context before an interruption and response behaviour. Finally we assess the extent individual response behaviour can be predicted through machine learning algorithms.

For this case study we focus on notifications that require timely delivery to be effective and provide the user with information in stages, consistent with the model in Figure 2. Whilst this doesn’t represent all possible variations of notifications, it is representative of a subset where interruptibility is critical for success and is in-line with similar work [18]. We also used Android’s default parameters where possible, including using the default tone, vibration pattern and LED pattern.

4.1 Data Collection: Interruption experiment

We developed an Android smartphone application, called ImprompDo, designed to deliver notifications, collect detailed context data and record response behaviour. So that a participation incentive naturally exists beyond our research purpose [18], this took place implicitly behind the functionality of a to-do list productivity tool.

Application setup After installation the user is guided through a short setup process of consenting to participate in the study and authorising access to their existing to-do lists (Todoist or Google Tasks). The user is then presented with optional preferences (Figure 3), modifiable at any time. Notifications could begin from the start of the next hour.



GLOBAL SETTINGS

Notifications
Receive push notifications. ON

Notification Availability
Between 8:00 - 23:00

Notification Frequency
On average, every hour

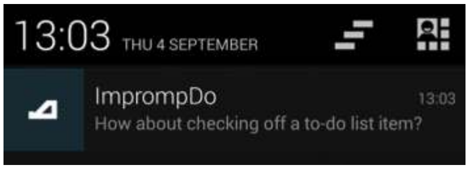
SYNC TO-DO LIST SERVICES

Google Tasks ON

Fig. 3. The preferences screen

Notification delivery and response Notifications were delivered within a user defined hour range (9am to 9pm by default) and maximum frequency (from once an hour up to once a day). If an interruption occurs, a response follows the same process as any other notification (shown in Figures 4-6). Although it would be possible to check whether the user did complete the to-do item (D4), these decisions are largely dependent on individual to-do list usage behaviour, hence to maintain generality, we chose not to consider this step.

Fig. 4. The application icon shown for an example ImprompDo notification

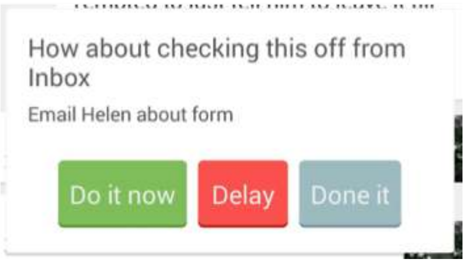


13:03 THU 4 SEPTEMBER

ImprompDo 13:03

How about checking off a to-do list item?

Fig. 5. The notification drawer with a ImprompDo notification summary



How about checking this off from Inbox

Email Helen about form

Fig. 6. The application content shown if the ImprompDo notification is consumed.

At the beginning of each period, a random trigger was chosen that dictates if and when the notification would occur. Inspired from related works, these triggers were: at a random time, at the end of a period of acceleration, a temporal online learning model (using hours) and a multi-modal online learning model using logistic regression with features extracted from captured context. This follows a 1 x 4 repeated measures within-subjects design implemented as a *N of 1 randomized trial* [15, 20]. This intended to prevent skewness by not splitting users into groups for each trigger, and because user participation time "in-the-wild" cannot be guaranteed.

Each notification prompts the user once and remains active for 30 seconds, or until the user either selects a to-do list item action (Figure 6) or dismisses it. After 30 seconds, the interruption is deemed unsuccessful in producing a response and is removed. We assume that this window provides sufficient opportunity for the user to respond if they were physically interrupted and could respond immediately. This design choice intended to keep the local context at the time the interruption consistent with the context if the user were to respond.

4.2 Data Collection: Implicit sensing

To capture context before interruption and during response we chose data sources that would be readily available for a real-world smartphone application - without adding any extra permissions or user tasks. With little co-agreement across the literature [5], we adopted a bottom-up approach of collecting from a variety of local hardware sensors and software APIs. The data sources chosen to extract feature variables from were: linear acceleration (pseudo-sensor), gravity (pseudo-sensor), light sensor, proximity sensor, charging state, screen on/off state, lock/unlock state, volume state and the current timestamp. Whilst these are a subset of what the smartphone is capable of, these represent what is typically available on an Android smartphone, across manufacturers and models.

Additional context about what the user is doing could be provided by calendar data or activity recognition. However, calendar data has high granularity and is often incomplete, especially beyond working hours. Detailed activity recognition is restricted by the inability for current smartphones to do this accurately and efficiently. Other environmental data such as location and ambient noise (microphone) were omitted due to the additional permissions required. This also has a tenuous link to the to-do list application and may deter users from participating and could introduce a behavioural bias [13].

Data was collected implicitly as notifications were delivered and remained active. Sampling begins from 5 seconds before the interruption until the notification is consumed/dismissed by the user (D3) or otherwise it expires 30 seconds after interruption (Figure 7). Sampling consists of taking sets of raw data vectors containing a reading from each data source. As readings are delivered by Android asynchronously, a short window is opened to listen for readings. It is

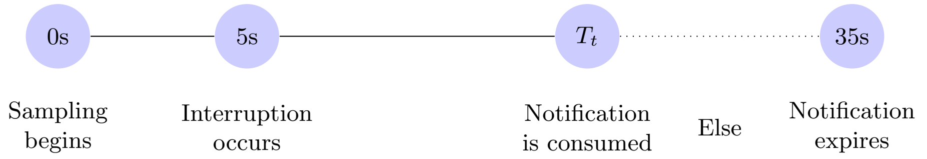


Fig. 7. Visualisation of the implicit data collection. Sampling occurs from 5s before delivery up until the notification is consumed at time T_t ($5s < T_t < 35s$) or it expires.

Decision	**Continue on event**	**Exit early event**
D1*	Screen is switched on	Screen stays off until timeout
D2*	Device is unlocked	Screen is switched back off
D3	Notification is tapped on	Notification is dismissed or timeout
D4	Application dependent	Application dependent

Table 1. Decision outcomes observed through progressive smartphone events. *Only observable if an Android device is not-in-use.

closed when either: at least 1 reading is collected from all data sources or 2 seconds has elapsed. When closed, the most recent readings are taken to minimise variance between reading times. If no readings were available after 2 seconds, the reading for that data source is set to null in the vector. An attempt is made to open a new sampling window immediately, however this is subject to device speed and system stability. The collection of raw data vectors is then examined to extract decision behaviour, using the criteria detailed in Table 1.

5 Case Study Results

After a period of 6 months, we analysed the aggregated data of all users to determine whether: capturing detailed response behaviour is beneficial; different contexts before an interruption are correlated with different behaviour; and whether partial and complete responses could be predicted. A summarised breakdown of the dataset is shown in Table 2. It should be noted that user participation was voluntary and could have occurred any time within the 6 months for any duration. Missing data prevented the extraction of whether the device was in-use or not at the time of the interruption or the decision behaviour afterwards, in 1287 of 11,346 cases (11.343%). Of the remaining 10,059 cases, the majority of notifications were not completed, with only 1056 of 10,059 (10.498%) consuming the notification - consistent with other studies (e.g., [15]).

5.1 Capturing partial response behaviour

Identifying partial responses, and therefore preventing them from being misinterpreted as null-responses, is possible under our approach. When the device was not-in-use at the time of interruption, partial responses occurred in 1126/7737 cases (14.553%). Whilst this may appear small, this increases the total distribution of cases where some degree of interruptibility was shown from 1056 cases (across in-use and not) to 2182 in 10,059, representing a substantial 106.629% increase. Including notification dismissals when the device was in-use would increase this further. Overall, this suggests that whilst users aren't interruptible most of the time, decomposing interruptibility through our model reveals additional cases to potentially consider. If further APIs are made available (e.g.

Metric	**Value**
Total number of users with >1 notification	93
Total number of notifications	11,346
Total number of days	178
Average number of notifications	122
Average number of days	26.457
Interruptions when the device was not-in-use	7737
- Null responses	5939
- Partial responses	1126
- Complete responses	672
Interruptions when the device was in-use	2322
- Null responses	1747
- Partial responses	191
- Complete responses	384
Interruption cases with missing data	1287
- Unknown if in-use or not-in-use	1267
- Unknown response behaviour	20

Table 2. A summary breakdown of the ImprompDo dataset.

notification drawer UI events), then further cases could be captured when the device is in-use.

By assessing the exact decision stage the user discontinued responding, we can also infer why they may have exited. For example, if the user turns the screen on to show the application icon, but goes no further, they don't know what the exact to-do list item was, so their reason for stopping cannot be due to undesirable content.

5.2 Correlating context to decision behaviour

Before conducting our analysis we performed Kolmogorov-Smirnov tests, which determined the presence of non-normal variable distributions. Therefore, non-parametric equivalents to t-tests were used. For variables with 2 possible values, Mann-Whitney U tests were used. Kruskal-Wallis 1-way ANOVAs were used for those with more than 2 values, to reduce the likelihood of Type I statistical errors. We began with analysing whether a particular trigger was significantly better at producing at least partial responses. From pairwise post-hoc tests from a Kruskal-Wallis test we found that no trigger was significantly better than all others. As a result we chose to analyse the significance of each variable individually, towards building other multi-modal prediction models.

To evaluate the prospects for prediction, we analysed whether the context sampled in the 5 seconds before an interruption was correlated to the outcomes of each decision (continue or exit). Raw sensor readings on Android devices have previously been shown to be inconsistent [8] and require filtering [9]. Therefore to stabilise the readings over the 5-second period, we took the mean value of each

	Not-in-use			In-use
Feature Variables	**D1**	**D2**	**D3**	**D3**
Accelerating* True, False	.186	.458	.072	**.000**
Ambient Light** Dark, Dim, Light, Bright	**.000**	**.039**	**.000**	**.000**
Screen Covered* True, False	**.000**	.187	**.000**	**.005**
Volume State** Silent, Vibrate, Audible	**.000**	**.009**	**.011**	**.000**
Orientation** Flat, Upright, Other	**.000**	.098	**.000**	**.000**
Charging State* True, False	**.000**	**.001**	.145	.177
Time of Day** Morn, Aftrn, Eve, Nght	**.002**	.125	.936	**.000**
Day of the Week**	.509	.794	.100	**.000**
Number of cases (n)	7737	1798	1469	2322

Table 3. P-values indicating significance of each feature on each decision. Bold values show significance using $p < 0.05$. * Mann-Whitney U Test ** Kruskal-Wallis 1-way ANOVA

data source and categorised the result (shown in Table 3). For the "Accelerating" variable, a high-pass noise filter was also applied (threshold = $0.1m/s^2$) to ensure that acceleration was substantial.

Each variable was then tested for statistical significance by analysing the distribution of its values across each decision outcome (continue or exit). Table 3 provides an overview of the p-values from the analysis. Overall, the results reveal differences in the significant variables between different decisions, as well as when the device is in-use and not, showing potential for predicting response behaviour at a finer granularity than a *black-box* model.

5.3 Response prediction

To investigate the extent in which response behaviour can be predicted, we built predictive models from the same dataset using the machine learning toolkit WEKA. We experimented with whether having a different model for each decision (with {continue, exit} classes) would perform better than a multi-class model predicting the exact decision the user exited the response. An exit at D3 is synonymous with a complete (consumed) response.

From frequency distributions we found that the distribution of class labels in the dataset was imbalanced. Therefore we used random-under-sampling (RUS) to prevent skewing classifier performance. We chose 7 classifiers used in related works and performed 10-fold cross validation on each model using 100 randomly

		not-in-use				**in-use**
Classifier	**Metric**	**D1**	**D2**	**D3**	**MC**	**D3**
AdaBoostM1	Precision	0.6045	**0.6064**	**0.6375**	0.2522	0.5927
	Recall	**0.6026**	**0.6045**	**0.6369**	0.2976	0.5923
BayesNet	Precision	0.5936	0.5873	0.5955	0.2532	0.4997
	Recall	0.5889	0.5831	0.5917	0.2870	0.4996
J48	Precision	**0.6065**	0.5986	0.6316	0.3376	**0.6010**
	Recall	0.6023	0.5957	0.6294	**0.3393**	**0.6002**
Logistic	Precision	0.5719	0.5791	0.6118	0.3217	0.5881
	Recall	0.5718	0.5790	0.6117	0.3272	0.5879
NaiveBayes	Precision	0.5715	0.5816	0.6195	**0.3408**	0.5889
	Recall	0.5702	0.5801	0.6174	0.3372	0.5872
RandomForest	Precision	0.5788	0.5769	0.6250	0.3277	0.5939
	Recall	0.5787	0.5768	0.6246	0.3283	0.5938
SMO	Precision	0.5664	0.5779	0.6036	0.3233	0.5941
	Recall	0.5659	0.5761	0.6017	0.3248	0.5928

Table 4. Classifier performance using models at varying granularities. Bold values indicate the highest value across classifiers.

under-sampled balanced datasets. The mean performance of each classifier is shown in Table 4.

The results show poor performance for the multi-label model in comparison to individual models dedicated to predicting continue or exit for each decision. For the individual models, the performance is in-line with similar studies (e.g., [15]), with D3 the equivalent to a black-box model. Given that this is an aggregated dataset and that humans can have varying smartphone and interruption habits this performance (of around 60%) is neither unexpected nor unreasonable. Interestingly, the variance in classifier performance for each decision and between decisions is small for both metrics and not statistically significant. Crucially, this shows that partial response behaviour can be predicted as well as complete responses. This is also beneficial for real-world implementation as the same classifier can be used for all decisions without a detrimental affect on performance.

Additionally, although having separate models increases complexity, a single computationally cheap classifier can be used without significant performance loss - improving viability as the smartphone has limited resources. Going forward, this shows that predicting response behaviour at decision-level granularity is possible. These results provide a baseline for further work on whether performance can be improved with personalisation and online learning.

6 Limitations and future work

Notifications vary in content and purpose [14], so to assess the suitability of our model we've used a notification that is representative of those where untimely delivery would be useless for both the interrupter and the interruptee. However, other types of notifications, particularly less timely notifications (e.g., digest type notifications such as Twitter's notifications) and rapport-driven notifications (e.g., instant messaging) should also be explored.

We have focused on Android smartphones running version 4.0 - 4.4, which represented between 85%-90% of the market distribution within Android versions at the time of the study. However, the delivery and response conventions to notifications can vary across operating systems, inhibiting a *one-size-fits-all* case study. Theoretically, the premise of the model remains, the exact decisions and indicators of decisions being made may need to be adapted from *a priori* knowledge of these systems.

Going forward, several steps to potentially improve prediction performance can be taken. Firstly, the performance of alternative training methods, such as online or evolutionary learning, could be explored. Secondly, further work could explore whether personalised training data could improve performance or a mix with aggregated data. Thirdly, we've intentionally used typical data sources available for Android devices. If technical constraints and behavioural bias risks can be mitigated, the predictive power of other data sources should be explored. Finally, if additional measures to implicitly observe decision behaviour become available, this should be explored, particularly for when the device is in-use.

For the model itself, an ongoing research question remains in distinguishing between cases where the user wasn't physically interrupted (e.g. the smartphone wasn't near them) and those where they were but they didn't perform any observable actions on the device, i.e. null-responses.

7 Conclusion

Smartphone notifications have extended the diversity and frequency of interruptions we receive throughout our daily lives. Intelligent systems for inferring interruptibility and likely success of a timely response are highly desirable to the remove the reactive burden placed on users. The current conventions for modelling the response behaviour to notifications heavily rely on complete responses and the user to provide labelling. However in reality notifications have high variability [14], a user might be interruptible but not for all notifications [12].

For intelligent systems which seek to decide whether to push or delay a notification in situ, we explore whether the natural decision process that the user goes through when being interrupted [12] can be observed for smartphone notifications. We present a model of up to 4 sequential decisions a user faces when receiving and responding to notifications and find support for our hypothesis that decomposing how a response is made is worthwhile. Through an "in-the-wild" case study, we observe that including partial responses when the device is not-in-use increased the number of cases where some degree of interruptibility was shown by 106.629% - reducing false-negative misclassifications.

Additionally, we find that this is achievable without explicit user annotations through implicitly observing how the user interacts with the device. From this we identify that different features in the context before an interruption are significantly correlated to different partial and complete response behaviour. Finally, we attempt to predict the extent in which a user pursues a response, with accuracy in-line with related work in the area, but with the benefit of also predicting partial responses.

References

1. Adamczyk, P., Bailey, B.: If not now, when?: the effects of interruption at different moments within task execution. In: Proc. CHI'04. pp. 271–278. ACM (2004)
2. Bailey, B., Iqbal, S.: Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management. ACM Transactions on Computer-Human Interaction (TOCHI) 14(4), 21 (2008)
3. Fogarty, J., Hudson, S., Atkeson, C.G., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J., Yang, J.: Predicting human interruptibility with sensors. ACM Transactions on Computer-Human Interaction (TOCHI) 12(1), 119–146 (2005)
4. Grandhi, S., Jones, Q.: Technology-mediated interruption management. International Journal of Human-Computer Studies 68(5), 288–306 (2010)
5. Ho, J., Intille, S.: Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. In: Proc. CHI'05. pp. 909–918. ACM (2005)
6. Horvitz, E., Apacible, J.: Learning and reasoning about interruption. In: Proc. ICIMI'03. pp. 20–27. ACM (2003)
7. Iqbal, S., Bailey, B.: Effects of intelligent notification management on users and their tasks. In: Proc. CHI'08. pp. 93–102. ACM (2008)
8. Lathia, N., Rachuri, K., Mascolo, C., Rentfrow, P.: Contextual dissonance: Design bias in sensor-based experience sampling methods. In: Proc. UbiComp'13. pp. 183–192. ACM (2013)
9. Liu, G., Hossain, K.M.A., Iwai, M., Ito, M., Tobe, Y., Sezaki, K., Matekenya, D.: Beyond horizontal location context: measuring elevation using smartphone's barometer. In: Adjunct Proc. UbiComp'14. pp. 459–468. ACM (2014)
10. Mathan, S., Whitlow, S., Dorneich, M., Ververs, P., Davis, G.: Neurophysiological estimation of interruptibility: Demonstrating feasibility in a field context. In: In Proceedings of the 4th International Conference of the Augmented Cognition Society (2007)
11. McFarlane, D.: Interruption of people in human-computer interaction: A general unifying definition of human interruption and taxonomy. Tech. rep., DTIC Document (1997)
12. McFarlane, D., Latorella, K.: The scope and importance of human interruption in human-computer interaction design. Human-Computer Interaction 17(1), 1–61 (2002)
13. Miller, G.: The smartphone psychology manifesto. Perspectives on Psychological Science 7(3), 221–237 (2012)

14. Okoshi, T., Ramos, J., Nozaki, H., Nakazawa, J., Dey, A., Tokuda, H.: Attelia: Reducing users cognitive load due to interruptive notifications on smart phones. In: Proc. PerCom'15. IEEE (2015)
15. Pejovic, V., Musolesi, M.: Interruptme: designing intelligent prompting mechanisms for pervasive applications. In: Proc. UbiComp'14. pp. 897–908. ACM (2014)
16. Pielot, M., de Oliveira, R., Kwak, H., Oliver, N.: Didn't you see my message?: predicting attentiveness to mobile instant messages. In: Proc. CHI'14. pp. 3319–3328. ACM (2014)
17. Pohl, H., Murray-Smith, R.: Focused and casual interactions: Allowing users to vary their level of engagement. In: Proc. CHI'13. pp. 2223–2232. ACM (2013)
18. Poppinga, B., Heuten, W., Boll, S.: Sensor-based identification of opportune moments for triggering notifications. Pervasive Computing, IEEE 13(1), 22–29 (2014)
19. Rosenthal, S., Dey, A., Veloso, M.: Using decision-theoretic experience sampling to build personalized mobile phone interruption models. In: Pervasive Computing, pp. 170–187. Springer (2011)
20. Sidman, M.: Tactics of scientific research: Evaluating experimental data in psychology. Basic Books New York (1960)
21. Smith, J., Lavygina, A., Ma, J., Russo, A., Dulay, N.: Learning to recognise disruptive smartphone notifications. In: Proc. MobileHCI'14. pp. 121–124. ACM (2014)
22. Ter Hofte, H.: Xensible interruptions from your mobile phone. In: Proc. MobileHCI'07. pp. 178–181. ACM (2007)