Predicting User Responsiveness to Smartphone Notifications for Edge Computing

Andreas Komninos[1](✉), Elton Frengkou[1] and John Garofalakis[1]

[1] University of Patras, Rio 26504, Greece
akomninos@ceid.upatras.gr

Abstract. Edge computing requires the addressing of several challenges in terms of privacy, complexity, bandwidth and battery life. While in the past attempts have been made to predict users' responsiveness to smartphone notifications, we show that this is possible with a minimal number of just three features synthesized from non-sensor based data. Our approach demonstrates that it is possible to classify user attentiveness to notifications with good accuracy, and predict response time to any type of notification within a margin of 1 minute, without the need for personalized modelling.

Keywords: Smartphones, Notifications, Attentivity, Responsiveness.

1 Introduction

Smartphone notifications are part-and-parcel of everyday interaction with our devices and the services we expect to receive from them. Unfortunately, despite several years of research, managing these notifications remains a mostly manual task, in which the users are unassisted by intelligence embedded in the operating system, or even applications themselves. Recent work has shown that it is possible to assist users by anticipating opportune moments to issue notifications (i.e. moments in which the user is likely to be attentive to the device, and also able to engage with the notification content [1]). Other work has attempted to examine the role of modality in attracting the user's attention (e.g. [2]). Generally, to intelligently manage notifications, Anderson et al. [3] propose a four-stage system approach, starting with acquiring sensor data, processing sensor data, inferring context from sensor data, building interruptibility models and finally managing an incoming notification (selecting modality, or deferring it). However, all previous approaches assume the analysis of large volumes of data in the cloud, collected by a large variety of device sensors and context data sources, placing not just a strain on the device battery but also doubts on the applicability of these techniques, both due to privacy concerns, and device constraints (e.g. network availability): Precisely the challenges that modern *Edge Computing* paradigms aim to address.

In this field of research, the least studied aspect is the issuing of a notification under the selection of an appropriate modality. The choice of modality is a balancing act that requires awareness of the user's context (e.g. social surroundings, time of day, likely activity etc.), and also the notification's context (e.g. the event it relates to, its general importance, its relative importance to the user's current task, whether the user's immediate attention is required, the cost or impact of not attending to the event etc.).

For any researcher that has even trivially worked in context awareness, it is easy to see that these types of context are difficult to acquire – even if it were indeed possible, it remains entirely plausible that the user might prefer a system not to have such extensive and intimate knowledge about their context for privacy reasons. Thus, on one hand we have today's approach, in which the user is notified immediately using any modality that the service developers consider appropriate to attract attention, and on the other hand, a shift of the locus of control towards a notification management system which determines the appropriate modality to use given the user's and notification's contexts. To give an example, such a system might dynamically alter pre-programmed notification modalities to issue trivial notifications using the device LED only while the user is working, and allow the device to sound and vibrate only if there's an incoming call from a stressed colleague working on the same project.

From the above it is plain to see that both approaches have problems: Leaving the locus of control entirely up to the users causes frequent disruption and frustrations and also leads to missing important notifications, as the users resort to coarse handling strategies such as setting the phone to silent (which affects all notifications). On the other hand, shifting the locus of control towards the system can still have grave consequences when the system doesn't get it right and ends up causing the user to miss important information. In this sense, the situation becomes akin to text entry autocorrects and the embarrassing moments it has caused, shared across the web – usually it works well, but when it doesn't, the cost to the user can be very serious. In this context, we present here an analysis of real-world notifications, and discuss a model to predict the users' engagement with notifications which aims to use minimal data sources, in order to preserve user privacy and minimize the resources required for predicting user responsiveness in AmI environments using edge computing architectures.

2 Related Work

Predicting interruptibility and opportune times to deliver smartphone notifications is the subject of several recent research efforts. Much of the research is summarized by the excellent recent survey in [3] so we will not repeat it here, however, we lay out the parameters of recent important work related to our topic in **Table 1** so that our own contribution may be placed in context with past work.

Our work complements existing approaches in a number of ways. In contrast with most previous work, our analysis comes from user's behaviour with notifications from any app and the OS itself, and not from a notification issued by a single application. We also refrain from using privacy-sensitive features (e.g. user location or application type). Further, all previous approaches consider the just the device ringer mode (e.g. [4][5][6]), which does not actually equate to the modality with which a notification is delivered, as we explain next. In our work, we algorithmically determine the true actual modality with which the notifications were delivered. Another differentiation is that we employ only opportunistic data collection without sensor sampling, to minimize impact on the user's battery. Lastly, while previous work rests on classification (e.g. is the user reachable, or attentive to their device), we also report the results of a regression-based approach to quantify the reaction time to notifications.

Table 1. Overview of recent notification behaviour prediction research

Paper	Notification source	Users	# Notifications	Context features	Measurement	Performance
Okoshi et al. [1]	Single app	687,840	N / A	387	Response time	49.7% reduction
Pielot et al. [4]	Single app	337	78,930	201	Notification acceptance	0.31 (F_1 score)
Turner et al. [5]	Single app	93	11,396	9	Notification acceptance (multi-level)	~80% prec.
Poppinga et al. [7]	Single app	314	6,581	9	Notification acceptance (binary)	77.48% acc.
Pielot et al. [6]	Messaging apps	24	6,423	17	Attentivess (binary)	68.71% acc.
Okoshi et al. [8]	Single app	30	2,162	45	Response time	12% reduction
Turner et al. [9]	Single app	93	11,396	9	Notification acceptance (multi-level)	34-65% acc.
Mehrotra et al. [10]	All apps	35	70,000	14	Notification acceptance (binary)	70-80% specificity

3 Data Collection

3.1 Understanding the Android Notification System

All previous in-the-wild studies that we have found rely on the Android OS, which allows application programmers to specify *desired* notification modalities in their code. Hence a notification may be programmed to request from the device any combination of modality during issue, including the device LED, sound and vibration. Users can specify a ringer mode for their device, either manually, or, in later versions of the OS, via context-driven rules (e.g. set to completely silent between certain hours, or only allow certain applications at these hours). The ringer mode may *suppress, but does not add* beyond the programmed modality requests (thus will not add a LED illumination, vibration or sound to a notification which is not programmed to have one). Furthermore, the Android OS allows users to suppress notification modalities for individual applications. The locus of control in the way a modality is used to issue a notification is shown diagrammatically in **Fig. 1**. The OS overrides programmed modality requests based on a range of possible user settings.

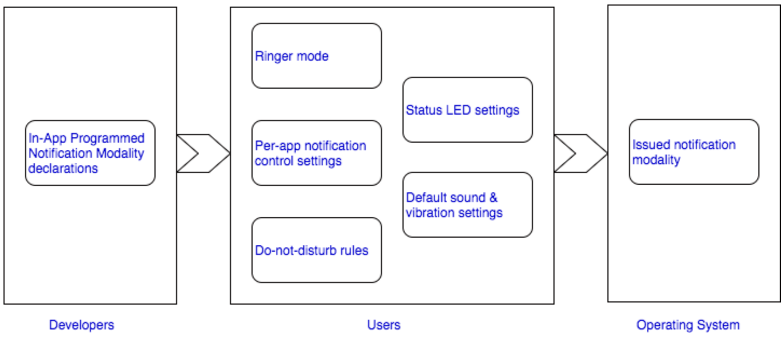

Fig. 1. Locus of control over the actual modality with which a notification is issued in the Android OS. App developers request a combination of modalities, but the OS may not necessarily honour these, depending on the user's individual settings and preferences.

Another point is that some notifications are not dismissible until a task completes or a user performs some action (e.g. showing a downloading progress bar, or a low disk-space event) and that some persist during ongoing events (e.g. a phonecall).

To infer thus reliable conclusions on how a notification modality influenced response time, a study should capture all types of information (what were the programmed notification modalities, user per-app preferences and what was the current ringer mode at the time of notification). If only the current ringer mode, or programmed notification modality are captured, then we cannot know with any certainty what modality was actually used to issue a notification and, consequently, estimate the effect an individual modality might have had on the users' response time.

3.2 Collected data

To collect data for our analysis, we built a simple logging application which works silently as UI-less background process. Previous works have employed a range of sensors to detect context (e.g. [8]). A downside of this approach is that frequent sensor sampling drains the users' battery. We adopted here a more opportunistic approach, sampling context information *only* at the time when a notification was issued, *without using any hardware sensor* data (e.g. accelerometer, GPS). Another consideration in our approach was the number of features sampled. In [4], the researchers collected data for 201 features, without any justification relating to their use. We believe this indiscriminate collection of data without at least some evidence to support their selection is unnecessary and of course, consists a significant privacy violation with doubtful utility (only a handful of features were shown to have some impact on engagement). The concerns over privacy are mentioned as a challenge in [3] and thus we selected to collect data only for features that relate immediately to the direct perceptibility of a notification. The application subclasses the Android "notification listener" service. The service is triggered, after the user has granted the relevant one-time permission, every time a notification is issued by any application or the OS itself. This service is also triggered upon dismissal of a notification. The callback methods in the service allow access to all the information related to the notification. Therefore for every notification we captured the following features relating either to the notification itself, or the device state at the time of issuing the notification:

Table 2. Captured notification and device state features

	Feature	Description	Values
Notification features	Time posted	Timestamp at issue time	Unix time (seconds)
	Time removed	Timestamp at dismissal time	Unix time or blank (self-cancelling notifs.)
	Package name	Package identifier of the application	String literal
	Sound	Use a custom sound at issue time?	Sound clip URI
	Default sound	Use the device defaults at issue time?	True | false
	LED	Use a custom LED colour and pattern at issue time?	LED on/off time pattern
	Default LED	Use the device defaults at issue time?	True | false
	Vibration pattern	Use a custom vibration pattern at issue time?	Vibration on/off time pattern
	Default vibration	Use the device defaults at issue time?	True | false
	Notification flags	A bit mask with flags related to the notification.	1: should use LED 2: ongoing notification 4: insistent notification 8: only-once 16: auto-cancelling 32: no-clear 64: foreground 128: high priority 256: local only 512: group summary
Device features	Ringer mode	The current device ringer mode	0: silent (LED only) 1: vibration & LED 2: sound, vibr. & LED
	Interactive	Whether the device is in a current "interactive" state (ready to interact with the user)	0: device sleeping 1: ready to interact
	Screen state	The current screen state	1: off 2: on 3: dozing 4: dozing - suspended
	Allow lock-screen notifs.	Are notifs. allowed to be displayed on the lock-screen of the device?	True | false

For later versions of the Android OS, the captured notification shows not the programmed modalities, but the modalities that this notification was allowed to have based on the user's preferences (through the notification channel settings). Thus, this set of features covers the issues identified in section 3.1. The application was installed on the smartphones of 26 participants and was left to log data for a period of 4 weeks (28 days), after obtaining informed consent. The participants were computer science students, aged 18-22, recruited via social media and did not receive compensation. Of these participants, some dropped out of the study, others removed our service's access permissions after some time and some returned only very small datasets. As a result, our analysis proceeded with data from 14 (6f) participants.

4 Data Analysis

4.1 Feature transformation and cleansing

As stated above, to determine the actual modality with which a notification was delivered to a user, we need more than just ringer mode information only. Having collected both the requested delivery modalities (programmed modalities and filtering by per-app settings) *and* the ringer mode, it is quite easy to derive simple rules for determining the actual delivery modality used. Hence, in contrast with work such as [10] where these are not combined, we derive the synthetic binary variables "Had LED", "Had Sound" and "Had Vibration". From these values we determine the modality combination that was used to deliver a notification (one of 8 possible combinations). Furthermore, by subtracting the dismissal time from the issuing time, we derive the feature "Response Time" in seconds. Finally, from the issuing time, we calculate the hour of day in which the notification was issued ("Hour").

In terms of the data, we collected 176,195 notifications from the 14 participants. The full dataset can be obtained freely at https://github.com/komis1/ami2018-notifications. For analysis, we removed those notifications which the user could not manually dismiss and all notifications pertinent to ongoing events (e.g. phonecalls, downloads etc). We noticed also that on some devices the OS was generating many notifications which it issued and dismissed at the same time, so any notifications that had a response time of zero were also filtered out. Examining the reaction time, we found that there existed several outliers in terms of reaction time, and since 90% of the sample had a reaction time <1376 seconds (~22 minutes), we pruned the set at that threshold.

4.2 Statistical processing

In the following description, all statistical tests are chosen based on the distribution of the relevant variables (normal or otherwise). After pruning, the resulting notification set consisted of 12,612 notifications, showing that less than 10% of the notifications received by a user are actually *interactive* notifications, i.e. events for which the user's attention is required by an application or service. These had a mean response time μ=103.30s (σ=220.163s). The median was 16.00 seconds (compared to 6.15 *minutes* found in [6]) and 3rd quartile was 75.00s. As can be seen, the distribution of response time follows a power-law curve (**Fig. 2** left).

How responsive are users to notifications throughout the day? Another interesting observation is the diurnal distribution of issued notifications (**Fig. 2** right, shaded area), were we can see an increase of interruption after midday (skewness -.908, kurtosis -.357), after which the level of interruption remains relatively constant until approximately midnight, where it starts to decrease. This pattern resembles closely the one in [7] although that result discusses the rate of engaging with content in a notification issued by a single app, where as we could not know whether our participants engaged, or simply dismissed the notifications issued to them. To quantify the disruptiveness of the interruption, we looked at the distribution of response times per hour of day (**Fig. 2** right). As expected, notifications issued deep in the night have a visibly longer mean response time compared to the rest of the day. A Kruskal-Wallis H test (owing to the non-normal distribution of response time in each hour bin) reveals that response time is indeed distributed differently throughout the day with statistical significance ($\chi^2_{(23)}$=123.142, p<0.01). Interestingly, we observe that users seem to respond to notifications with almost the same speed as the day progresses, despite the considerable increase in volume of received notifications. The times of 8am – 11am seem to be those when users are least attentive to their notifications, something that may be partly explainable by the fact that they receive fewer notifications at these time, hence do not anticipate having to engage with them and are not proactively attending to their phone.

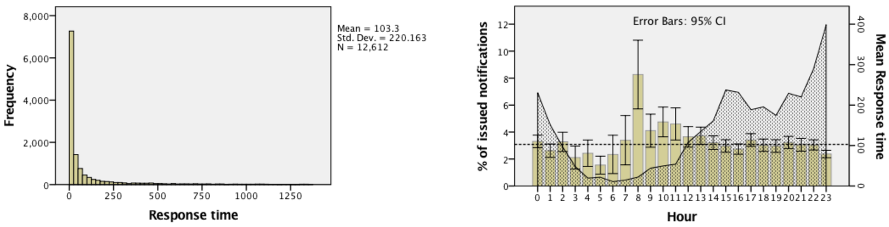

Fig. 2. Distribution of response time to notifications (left) and diurnal notification volume (right). The reference line is set at 103.3 seconds (overall average response time).

How is response time affected by ringer mode and actual delivery modality? The next step was to examine the effect of modality on response time. We note that the majority of notifications (46.61%) was delivered using the "LED & Vibration (LV)" combination, followed by "LED & Sound & Vibration (LSV)" (35.97%) and "No modality (NM)" (12.71%) (Fig. 3 left). A Kruskal – Wallis H test confirms that the distribution of response time within all the modality categories is different with statistical significance ($\chi^2_{(7)}$=383.757, p<0.01). The three modality combinations accounted for >95% of the notifications and thus we selected them for further pairwise comparisons, using Mann-U tests with post-hoc Bonferroni correction, setting the statistical significance level at p=0.017. In these tests, we find statistically significant differences across all comparisons, showing that NM (μ=133.13, σ=239.686) has the slowest response time compared to LSV (μ=119.49, σ=240.058) (Z=-8.763, p=0.00) and LV (μ=75.26, σ=182.594) (Z=-16.925, p=0.00). The difference between LV and LSV is also statistically significant (Z=-9.960, p=0.00).

The result indicates that users are more attentive to their devices when the modality used is LED & Vibration only, seemingly confirming the findings in [11][2] where it was found that when the phone was set to "vibration only" ringer mode, the response time to notifications was faster. Further examination of response time by ringer mode in our dataset, also corroborates previous results (**Fig. 3** left). A Kruskal-Wallis H test confirms the observed differences are distributed differently with statistical significance ($\chi^2_{(2)}$=50.262, p<0.01). Using Mann-U tests with post-hoc Bonferroni correction, setting the statistical significance level at p=0.017, we can confirm when the device ringer mode is set to "Vibrate only" (μ=81.46, σ=189.050) the response time is faster than "Silent" (μ=117.25, σ=234.029, Z=-6.902, p<0.01) and than "Sound and Vibrate" (μ=110.27, σ=228.993, Z=-4.693, p<0.01). We also noted that "Sound and Vibrate" has a statistically significant lower response time than "Silent" (Z=-4.485, p<0.01).

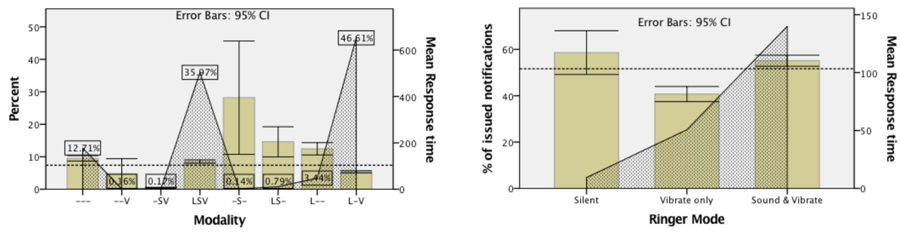

Fig. 3. Response time to notifications according to actual delivery modality (left) and ringer mode (right). The reference line is set at 103.3 seconds (overall average response time).

Naturally, an assumption could be made here that "silent" mode has the slowest reaction time because it would be a type of mode typically associated with contexts where no disturbance is required. Such a context might be night time, when users go to sleep. On the other hand, we noticed in **Fig. 2** (right) that users seem to be, if anything, more attentive to their devices at these hours. We noticed that users place their device on "silent" mode not just at night time, but also frequently during the day too (10am-5pm). In fact we also notice increased use of the "vibrate only" mode in these hours. In a sense that can be expected – these are normal class-going hours for students (**Fig. 4** left). However, when we plot the diurnal distribution of the actual delivery modality of the notifications, a totally different picture emerges (**Fig. 4** right). The discrepancy between user ringer mode and actual delivery of the notification is immediately obvious.

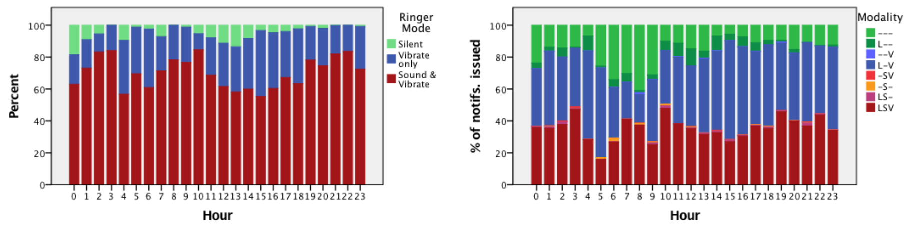

Fig. 4. Distribution of notifications according to ringer mode (left) and actual delivery modality (right).

The next question thus becomes, can the ringer mode, or actual delivery modality be used to predict response time to notifications? Bivariate Spearman's rho correlations between the mean response time in each hourly slot and the percentage of notifications delivered under each of the ringer mode settings at that time, reveal no statistically significant correlation (**Table 3**). On the other hand, a similar analysis between mean response time in each hourly slog and the percentage of notifications delivered with each of the 8 modality combinations, revealed a statistically significant negative correlation with LV (**Table 3**).

Table 3. Bivariate correlation coefficients and statistical significance between percentage of notifications under specific modality or general ringer mode and response time.

Modality	---	--V	-SV	LSV	-S-	LS-	L--	L-V
Spearman's ρ	0.237	0.226	-0.25	0.322	0.148	0.176	0.297	**-.524**
p-value	0.266	0.289	0.238	0.125	0.490	0.410	0.159	**0.009**
Ringer Mode	Silent	Vibrate	Sound & Vibrate					
Spearman's ρ	0.254	-0.283	0.156					
p-value	0.231	0.179	0.468					

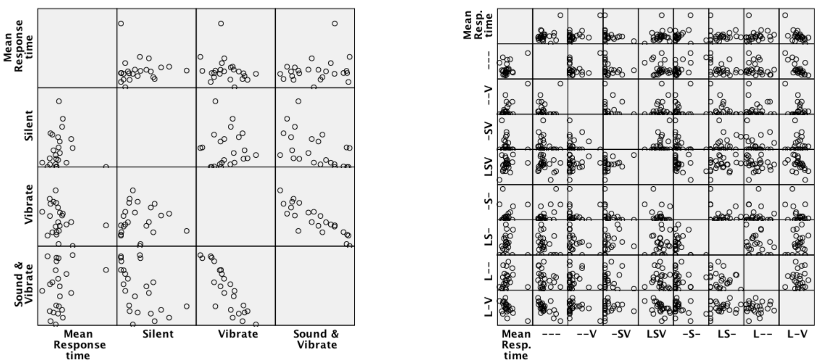

Fig. 5. Correlation plots for mean response time to notifications, and percentage of notifications delivered under given ringer mode (left) or with actual modality (right)

Further analysis with linear modelling shows that using the percentage of notifications delivered under each ringer mode does not sufficiently explain the variability in response time (R^2=0.65). This finding is in line with [6], where it was found that ringer mode is a rather weak predictor of message notification attentivity. On the other hand, linear modelling with the percentage of notifications under their actual delivery modality, the model fit is quite good (R^2=0.830), showing that the use of actual delivery modalities is a much better predictor for a population's attentivity to issued notifications.

What is the role of device and screen operational state in response time? A further set of features related to the device state ("interactive" or not) and the screen state, as discussed above. To explain this further, a device is in an "interactive" state when it is ready to interact with the user. The screen is typically "on" at that time, but might be temporarily turned off (e.g. by the proximity sensor while the user is taking a call). The device is considered to be "interactive" also while at the "dreaming" state (akin to a screensaver mode). If the device is not interactive (i.e. sleeping), it typically requires the device to be woken up by pressing the power button. The actual state of the screen regardless of its "interactive" mode, can be separately captured at any time. Of these two features (interactive and screen state), we are mostly interested in "interactive" because when combined with the screen state ON value, we can be relatively sure that the user is currently engaged with the device, while a non-interactive state shows that the user's attention is away from the device at the time of notification.

We encountered 3 of the four possible screen states in our set. **Table 4** shows the distribution of notifications by screen state and interactive state. As can be seen, our participants mostly received notifications while their device was at an interactive state with the screen turned on (69.63% of all notifications) or when the device was not interactive with the screen also turned off (29.25%). Together these represent 98.88% of all cases and can be synthesized into another feature (*Interactive-S*). For these two predominant device states we plotted the diurnal distributions (**Fig. 6**) and note that the probability of a notification arriving while the device is non-interactive is considerably less in the later hours of the day (after 3pm and until 4am). This gives an indication of when our users were mostly active on their devices and presumably more likely to respond to a notification quickly.

Table 4. Distribution of received notifications based on device state. Combinations that synthesize the Interactive-S feature are highlighted bold.

Screen state	Interactive		Frequency	Percentage
Off	TRUE		81	0.64%
	FALSE		**3689**	**29.25%**
		Total	*3770*	*29.89%*
On	**TRUE**		**8782**	**69.63%**
	FALSE		44	0.35%
		Total	*8826*	*69.98%*
Dozing	TRUE		0	0.00%
	FALSE		16	0.13%
		Total	*16*	*0.13%*

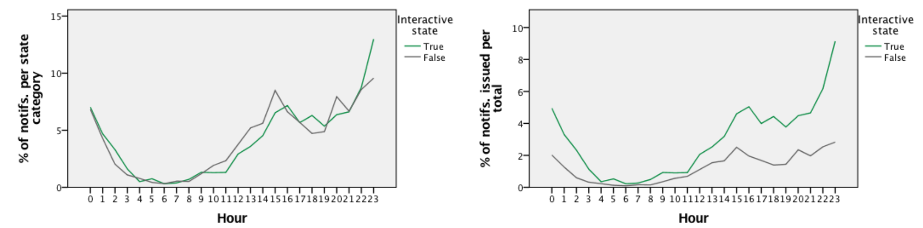

Fig. 6. Diurnal distribution of notifications received in each device interactivity state, as a percentage of each state's total (left) and of the total number of notifications (right)

On one hand, these observations might explain the relatively steady diurnal reaction time observed, as users were mostly engaged with their devices while receiving notifications, thus exhibited similar engagement behaviour. On the other, we might have expected a rather more immediate response than the average 103.30 seconds given that the participants were already interacting with their device. Indeed, it appears that whether the device is in an interactive state or not, does not give useful insight to the responsiveness to incoming notifications. Linear modelling shows that the variability explained by the percentage of notifications received under either interactivity state is low (R^2=0.087). This is a unique finding that shows that conceptually, there exists a lower temporal boundary for engagement and that responsiveness to notifications is a matter of conscious decision by the user, and not dependent on whether a notification is immediately noticed.

5 Predicting Reaction to Notifications with Machine Learning

In [7][6], the researchers attempt to predict whether the user is "likely" to attend to a notification. This is defined in [7] as a binary response to engaging with a notification regardless of response time, and in [6] using the median of response times to notifications as a threshold for classifying the user as having "high" or "low" attentiveness to the messages they receive. Following the latter approach, we selected two thresholds for attentiveness to notifications: "*Extremely attentive*", using our median (T=16 seconds) and "*Highly attentive*", using the average response time of T=103 seconds and "*Moderately attentive*" using the T=6.15 minute threshold in [6]. Therefore we attempt to answer the question "Will the user respond to a given notification once it has been issued within, or outwith temporal threshold *T*?".

We aimed to examine how predictive modelling using the "*raw*" data set captured from our logging application compared to a synthesized data ("*synth*"), as well as an extended set of our synthesized data containing additional features ("*synth+*") that we thought might be interesting to investigate (**Table 5**). The synth dataset is directly derived from the raw data, which can then be discarded to minimize the privacy risk to the user. For analysis, we used a range of classification algorithms (Naïve Bayes, SVM with Multiquadric kernel, Random Forest, NN). For each algorithm, a 10-fold cross-validation was performed using stratified sampling. The results are shown in **Table 6**.

Table 5. Features used for classification. "Interactive-S" is the synthesized interactive state as shown in Table 3. Attentiveness is the *target* variable.

Data set	Features	Values	Set size
Raw	Attentiveness	High | Low	12,612
	Hour	[0-23]	
	Interactive	True | False	
	LED	True | False	
	Sound	True | False	
	Vibration	True | False	
	Ringer mode	[Silent, Vibrate Only, Sound & Vibrate]	
	Screen state	[Off, On, Dozing]	
Synthetic	Attentiveness	High | Low	12,471
	Hour	[0-23]	
	Interactive-S	True | False	
	Modality	[---,--V,-SV,LSV,-S-,LS-,L--,L-V]	
Syn-thetic+	Attentiveness	High | Low	12,471
	Hour	[0-23]	
	Interactive-S	True | False	
	Modality	[---,--V,-SV,LSV,-S-,LS-,L--,L-V]	
	Lockscreen Notifications	True | False	
	Package name	<string value>	
	User ID	[1-14]	

Table 6. Classification average F_1-score results

Classifier	Raw dataset	Synth Dataset	Synth+ dataset
Bayes – 16s	66.32%	66.40%	65.11%
Bayes – 103s	71.97%	87.38%	87.57%
Bayes – 369s	88.41%	95.32%	89.99%
RF – 16s	66.67%	66.76%	67.03%
RF – 103s	88.35%	88.35%	88.35%
RF – 369s	95.42%	95.42%	95.42%
SVM – 16s	67.12%	67.12%	67.12%
SVM – 103s	88.35%	88.35%	88.35%
SVM – 369s	95.42%	95.42%	95.42%
DT – 16s	68.35%	66.79%	66.16%
DT – 103s	88.17%	88.25%	88.04%
DT – 369s	95.30%	95.40%	95.35%

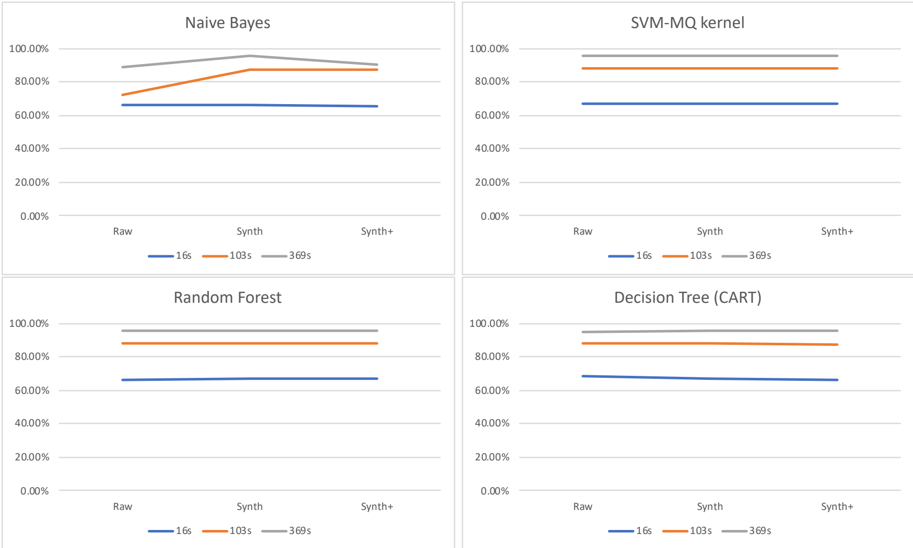

Fig. 7. Classification performance (F-score averages).

The immediate observation from the above is that prediction performance is strongly dependent on the attentiveness level set as the classification target. All classifiers perform exceptionally well (even the computationally inexpensive ones like baseline Naïve Bayes and Decision Tree) at the “moderately attentive” threshold from literature. The result is explainable as this threshold is well above the average response time in our dataset. Interestingly, good performance is obtained for the 103s “highly attentive” threshold, reaching close to 90% for all classifiers and independent of the feature set used, showing that it is possible to infer whether a user will engage with a notification within 103 seconds from issuing, with high accuracy. From there on, performance degrades to about 65% for all classifiers for the “extremely attentive” (16s) threshold. Given that this threshold is the median of our dataset, the classifier’s performance is considered to be better than random (50% chance), but still improvements might be possible to make in this regard.

Apart from these findings, it should be considered that the obtained results emerge from a very limited feature set, compared to previous literature that presents results based on large feature sets. We show that the baseline performance using the raw feature set is maintained when reducing the features through sensible combination to just four (in the “Synth” dataset). Additional features in the “Synth+” dataset show no significant performance gain. This result hints that application type doesn’t seem to strongly affect response times, thereby user responsiveness to notifications seems to be a *behavioral attitude* [3, table 3] which covers all types of notification, rather than being selectively applied to certain types. We also note that user ID doesn’t seem to improve performance, hinting that personalization of predictive models might not be necessary, at least for homogeneous population types such our participants. This result confirms the findings in [5] where it was found that personalized models only have benefits in very specific prediction targets and that general models are overall more successful.

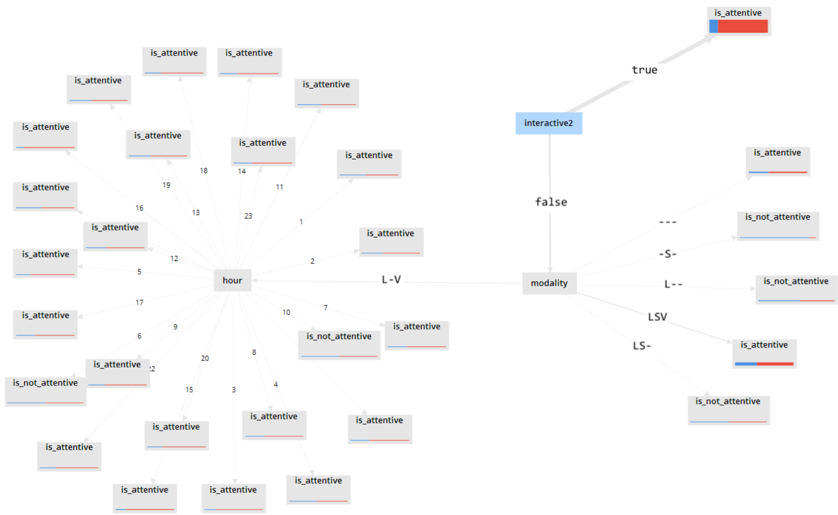

Fig. 8. Decision tree output (Synth dataset, 103s threshold). The blue proportion is the % of cases labelled as "not attentive" and red are the "attentive" cases. Root node is highlighted blue.

To further investigate, we examine the most interesting case, i.e. the "Highly attentive" threshold on the "Synth" dataset (least features), using the output from the computationally cheap Decision Tree (**Fig. 8**). We notice that the most important feature is "*Interactive-S*". If the device is not in an interactive state, then the *Modality* is the most important feature. The results show that using all modalities (LSV) the user is attentive to the notifications. When using only vibration-type modality (L-V), the result is dependent on the *Hour* of day. For "silent" notifications (i.e. no modality or LED only) is most likely attentive to the notifications, and for "sound only", users seem not attentive.

As a final step, we also used the decision tree modelling procedure with the "Synth" dataset, to perform regression on the dataset and examine the model's ability to predict the reaction time to notifications. With this procedure (10-fold cross validation) we achieve an average root mean squared error of μ=212.857 sec (σ=11.306s), which is about 3.5 minutes. The mean predicted response time (μ=102.91s, σ=64.77) is quite similar to the actual response time mean (μ=102.69s, σ=219.45) but as expected, the distributions are quite different and is skewed towards "later" predictions. When grouping by the feature value combinations, the response time Δ_{RT} (predicted – actual) is greatest for feature values [Interactive-S=0; modality=-S-; hour=6]:668s and [Interactive-S=0; modality=LS-; hour=14]: -1137s. However, we note that these feature value combinations only have one instance. For reference, best performance is observed for feature values [Interactive-S=1; modality=L-V; hour=21]: -0.006s (350 instances). When pruning the result set to include only those feature combinations whose instances make up for 95% of the dataset, and calculate the actual average Δ_{RT}=-3.41s (σ=39.68) and $|\Delta_{RT}|$=20.03s (σ=34.37s). Therefore, we can claim that the *usable* prediction result of the regression modelling is actually quite good: We can predict the time of response to notifications of any type an accuracy spanning at most 1 minute, using just 3 features.

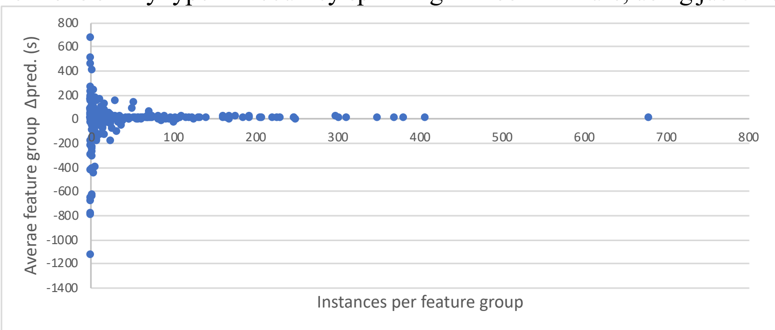

Fig. 9. With more instances per feature group, the average difference between actual and predicted response time converges to near zero.

6 Conclusions

As computational demands for resources in AmI environments increase in scale and complexity, a return to computing at the edge of the cloud is (somewhat ironically) seen by the community as the next forward step in context awareness [12]. The necessity of localized scalability and masking of uneven conditioning were foreseen in 2001 by [13], precisely to address the challenges of response time, battery life, bandwidth saving and data safety and privacy that edge computing aims to address.

Our exploration of real-world notifications is just the 2nd paper in literature (besides [10]) to address a body of notifications from all apps that a user has enabled on their device. In contrast with previous research, we have attempted to reduce the feature set for our classifier and prediction algorithms as much as possible. This approach results in lower storage, computational complexity, increased privacy for users and therefore better performance and power saving, if the classifiers and regressors are ran locally on the user device, as well as bandwidth saving if the data is uploaded for cloud processing. We demonstrate that "brute forcing" features into predictive models doesn't necessarily equate to better results, as we achieve very good outcomes in terms of both predicting the level of attentiveness and the actual response time to notifications, with just three synthesized features from a small number of raw data features (that can be discarded). The natural next step would be to explore the performance of our model in the wild, assessing its performance with unseen data and using it to drive intelligent notification deferment and modality choice policies. It would also be good to investigate gender aspects with a larger group of users, since in the current study, the group sizes are too small to provide adequate power for anything other than very large effects.

Our users were a homogeneous group and this places some limits on the generalizability of our findings beyond such users. However, this limitation also indicates that might be little need for personalized modelling in homogeneous user groups, which can overcome the "starting user" problem by partially sharing data between peers (e.g. groups of friends, relatives or the local community): in a sense, how we respond to notifications is, partly, a result of what others around us are doing. In future work we aim to examine privacy-preserving architectures for sharing such context and behaviour with new users transitioning across different local or regional cultures at different scales (e.g. tourists in new countries, or locals visiting a restaurant for the first time), in order to help devices automatically adapt their notification management support for users.

References

1. Okoshi, T., Tsubouchi, K., Taji, M., Ichikawa, T., Tokuda, H.: 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). pp. 100–110 (2017).
2. Mehrotra, A., Pejovic, V., Vermeulen, J., Hendley, R., Musolesi, M.: My Phone and Me: Understanding People's Receptivity to Mobile Notifications. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. pp. 1021–1032. ACM, New York, NY, USA (2016).
3. Anderson, C., Hübener, I., Seipp, A.-K., Ohly, S., David, K., Pejovic, V.: A Survey of Attention Management Systems in Ubiquitous Computing Environments. arXiv:1806.06771 [cs]. (2018).
4. Pielot, M., Cardoso, B., Katevas, K., Serrà, J., Matic, A., Oliver, N.: Beyond Interruptibility: Predicting Opportune Moments to Engage Mobile Phone Users. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1, 91:1–91:25 (2017).
5. Turner, L.D., Allen, S.M., Whitaker, R.M.: Reachable but not receptive: Enhancing smartphone interruptibility prediction by modelling the extent of user engagement with notifications. Pervasive and Mobile Computing. 40, 480–494 (2017).
6. Pielot, M., de Oliveira, R., Kwak, H., Oliver, N.: Didn'T You See My Message?: Predicting Attentiveness to Mobile Instant Messages. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 3319–3328. ACM, New York, NY, USA (2014).
7. Poppinga, B., Heuten, W., Boll, S.: Sensor-Based Identification of Opportune Moments for Triggering Notifications. IEEE Pervasive Computing. 13, 22–29 (2014).
8. Okoshi, T., Ramos, J., Nozaki, H., Nakazawa, J., Dey, A.K., Tokuda, H.: Attelia: Reducing user's cognitive load due to interruptive notifications on smart phones. In: 2015 IEEE International Conference on Pervasive Computing and Communications (PerCom). pp. 96–104 (2015).
9. Turner, L.D., Allen, S.M., Whitaker, R.M.: Push or Delay? Decomposing Smartphone Notification Response Behaviour. In: Human Behavior Understanding. pp. 69–83. Springer, Cham (2015).
10. Mehrotra, A., Musolesi, M., Hendley, R., Pejovic, V.: Designing Content-driven Intelligent Notification Mechanisms for Mobile Applications. In: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 813–824. ACM, New York, NY, USA (2015).
11. Pielot, M., Church, K., de Oliveira, R.: An In-situ Study of Mobile Phone Notifications. In: Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services. pp. 233–242. ACM, New York, NY, USA (2014).
12. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge Computing: Vision and Challenges. IEEE Internet of Things Journal. 3, 637–646 (2016).
13. Satyanarayanan, M.: Pervasive computing: vision and challenges. IEEE Personal Communications. 8, 10–17 (2001).