# Mindful Interruptions: A Lightweight System for Managing Interruptibility on Wearables

Claudio Forlivesi[*], Utku Günay Acer[*], Marc van den Broeck[*], Fahim Kawsar[*†]

[*]Nokia Bell Labs, [†]TU Delft
firstname.lastname@nokia-bell-labs.com

## ABSTRACT

We present the design, development, and evaluation of a personalised, privacy-aware and multi-modal wearable-only system to model interruptibility. Our system runs as a background service of a wearable OS and operates on two key techniques: i) online learning to recognise interruptible situation at a personal scale and ii) runtime inference of opportune moments for an interruption. The former is realised by a set of fast and efficient algorithms to automatically discover and learn interruptible situations as a function of meaningful places, and physical and conversational activities with active user engagement. The latter is substantiated with a multi-phased context sensing mechanics to identify moments which are then utilised to delivery notifications and interactive contents at the right moment. Early experimental evaluation of our system shows a sharp 46% increase in the response rate of notifications in wearable settings at the expense of negligible 6.3% resource cost.

## CCS CONCEPTS

•**Human-centered computing → Activity centered design;** *Mobile computing; Ambient intelligence;* **Mobile devices; Interaction design;**

**ACM Reference format:**
Claudio Forlivesi[*], Utku Günay Acer[*], Marc van den Broeck[*], Fahim Kawsar[*†]. 2018. Mindful Interruptions: A Lightweight System for Managing Interruptibility on Wearables. In *Proceedings of 4th ACM Workshop on wearable systems and applications , Munich, Germany, June 10, 2018 (WearSys'18),* 6 pages.
DOI: 10.1145/3211960.3211974

## 1 INTRODUCTION

The era of wearables has arrived. As more and more established forms (e.g., a timepiece, a ring, a pendant) get a digital makeover, they are reshaping our everyday experiences with new, useful, exciting and sometimes entertaining services. These forms typically have a unique set of characteristics, i.e., they are aesthetically appealing, ergonomically comfortable, socially acceptable, and develop an intimate relationship with their owners. Besides the critical functional values, these aspects determine the success and sustainability of wearable devices. While some of these aspects are more to do with design and ergonomics, interruptibility plays a critical role in maintaining sociality and intimacy of a wearable device. Indeed, for wearables to have a broader impact, the next generation wearables must expand their understanding of their users to determine the opportune moments of interaction, e.g., pushing the notification to a user at the right moment.

Understanding human interruptibility has a profound implication on computational user experience in a mobile setting. Naturally, many studies have explored this challenging topic and more recently in the context of mobile notification management [5, 8, 10, 11, 13, 14]. We aim to further augment this rich body of literature by presenting a multi-modal wearable sensing system to model interruptibility, however in a wearable-only setting. We first systematically explore different design challenges in building an interruptibility management solution for a wearable device, namely - personalisation, privacy awareness, user engagement, intelligibility, efficiency and continuous learning. Guided by these design considerations, we then present an end-to-end system that runs as a background service of a wearable OS and applies an online learning technique to recognise interruptible situations which are then utilised for contextual interruptions. To this end, the contribution of this paper is twofold.

- We present techniques to track the spatiotemporal activity trajectory of a user to model their interruptibility as a function of conversational and physical activity at different semantic spatial zones. These learned interruptible situations are then used to discover opportune moments to interrupt the user. We describe these techniques in an accessible and reproducible form.
- We present end-to-end wearable-only implementation of these techniques with benchmarks that show the efficacy of the proposed system including results from a real deployment case study.

Early experimental evaluation of our system demonstrates a sharp 46% increase in the response rate of interactive contents in a crowd-sourcing application at the expense of negligible 6.3% resource cost. Taken together this and the rest of the insights, our work contributes in accelerating the effort of designing interruptibility-aware applications.

## 2 DESIGN DECISIONS

Managing interruptibility on wearable in a mobile setting demands careful design consideration as it has multi-faceted implications on user experience. Taking a user-centred view, we have considered


*WearSys'18, Munich, Germany*

978-1-4503-5842-2/18/06…$15.00
DOI: 10.1145/3211960.3211974

the following design aspects while architecting our solution; the first two act as our guiding principles, whereas the last two are the operational principles.

**Personalisation and Privacy Awareness**: Given the intimacy of a wearable with its user, it is naturally expected that the wearable should have a sound understanding of the activity and lifestyle of its user to offer personalised services. This requires collecting, processing, and interpreting data from embedded sensors. However, this personalisation and the underpinning interpretation should not come at the expense of user privacy. So, the solution needs to be stand-alone without requiring any external network connectivity or remote processing. We have designed our solution with this view, i.e., all sensor data acquisition, processing, and interpretation are performed locally on the wearable to develop a personalised and privacy-aware view of users behaviour which is then used to manage interruptibility.

**User Engagement and Intelligible**: Many past studies have shown that an intelligent system without informed feedback and user engagement yield poor user experience [3, 6]. Building on these lessons, in our solution we actively engage the users in defining the behaviour of the system, e.g., qualifying a location with an interruptibility label and identifying moments of interruptibility. The composite effect of this active engagement is that the users are fully informed of the operational behaviour of our solution. For example *interrupting with a notification in the afternoon at a popular area while walking, while delaying the interruption in another case wherein the same location, in that same afternoon when a user is walking and talking with someone*

**Fast and Efficient**: Given the requirement of end-to-end execution of our solution locally and the constraint of low compute and energy resources of wearable devices, it is essential to design the system with ultra efficiency applying system optimisations at multiple phases. To this end, we have developed a set of simple and fast algorithms for discovering meaningful locations, identifying contextual situations, learning user preferences, and applying all these at runtime for interrupting the user at the right time for best user experience - while all are running locally without any network or remote processing.

**Online and Continuous Learning**: The last operating principle we applied in our solution is - continuously learn user behaviour and update the interruptibility model to accommodate changing user context. As we will discuss in the next section, we automatically discover meaningful locations and contextual activities in those locations based on a user's spatiotemporal activity trajectories. We frequently, i.e., every-time the wearable is charging, revisit these trajectories to update our rule-base that is then qualified by the user as needed. This continuous learning enables our solution to reflect a user's behaviour on the operational behaviour of the system in an intelligible way.

In the next section, we discuss how these design decisions are manifested in our system for managing interruptibility.

## 3 SYSTEM DESCRIPTION

The principal objective of our solution is to identify opportune moments for an interruption in a mobile setting. We achieve this by modelling spatiotemporal situations of a user, qualifying them with interruptibility labels and then using this behavioural understanding to drive opportunistic interactions. To this end, we have designed a system comprising following critical components as illustrated in Fig. 1.

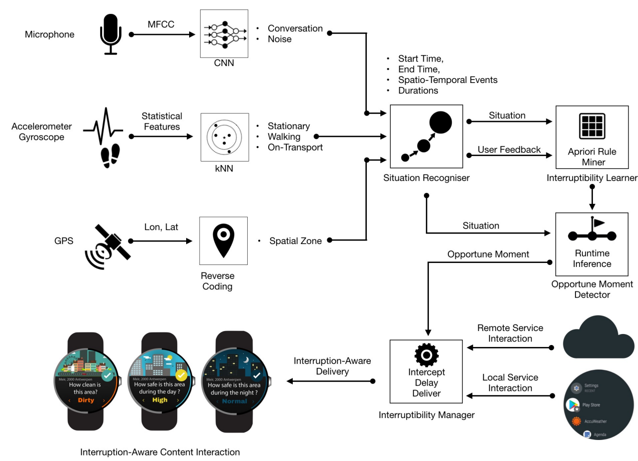


**Figure 1: On-Wearable Interruptibility Management System**

### 3.1 Multi-Modal Sensing and Context Models

A user's spatiotemporal activity trajectory is the primary input to our system. To construct this trajectory, we leverage three sensing modalities - motion, audio and GPS.

**Modelling Motion**: Motion sensing is constituted by onboard accelerometer and gyroscope to detect physical activity. We are interested in three-movement states: [*stationary*, *walking*, *on-transport*] and these are modelled by processing the raw accelerometer and gyroscope samples from the wearable. We use 5-second window frame with 95% overlap and then extract a set of time domain (mean, median, percentile, and RMS), and frequency domain (spectral energy, information entropy) features borrowing guideline from [4]. We pass these features to a k-nearest neighbours (K-NN)[1] classifier to extract physical activity label. We use 1000Hz sampling rate for the sensors, however, produce activity labels at a granularity of 30 seconds.

**Modelling Audio**: Audio is a versatile sensing modality and has been used for various complex tasks in recent literature, e.g., speech recognition, keyword spotting, acoustic scene detection, emotion and stress detection etc. We are interested in a simple task recognition, i.e., if there is a conversation going on or not. Our system listens to the onboard microphone for 3 seconds in every 30 seconds (please see later on how this listening is scheduled for optimising energy footprint) and extracts 13 MFCC features from a 16-bit-PCM audio data following a sliding window approach (25 ms-long window and overlap of 10 ms). MFCC features are then passed to a couple of classifiers - each one composed of a CNN

[1]We have tried a variety of other shallow and deep classifiers (e.g., linear SVM, RBF SVM, decision tree, random forest, multi-layer perceptron, AdaBoost, etc.), and selected the one that yielded the best performance both concerning accuracy and resource footprint.

followed by a SoftMax layer - for detecting the presence of human conversation in audio signals.

**Modelling GPS**: Location is known as the most powerful context for describing the human context. In our solution, we track only outdoor location by sampling GPS once every 30 sec. We apply reverse geocoding to extract high-level location out of raw longitude and latitude using a reference file achieved in our system Note that, we do not incorporate any indoor location sensing in our system, neither do we associate further semantics using an external database (e.g., Foursquare). We consider these aspects as the future avenue of this work.

Collectively, these three sensor modalities produce continuous spatiotemporal activity trajectory of a user as a tuple {*time, location, physical_activity, conversation_activity*} every *30 seconds* which is then used by the rest of the systems to determine opportune moments for an interruption.

## 3.2 Situation Recogniser

The primary objective of this component is to discover meaningful spatial zones for a user by mining spatiotemporal activity trajectories. Once a set of zones are discovered they are qualified by active user engagement for interruptibility, and from then on physical and conversational activities are tracked on zones that are marked as interruptible for mining user-defined rules.

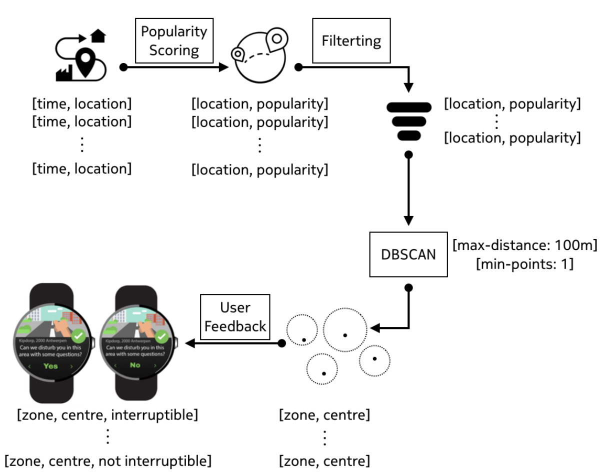


**Figure 2: Operating Model of the Situation Recogniser**

Spatiotemporal clustering is an active area of research, and there are many techniques proposed in the literature, e.g., k-means, constrained k-means, DBSCAN, ST-DBSCAN, spectral clustering, dynamic graph cutting, etc. to name a few. We can apply many of these techniques to discover meaningful locations. However, given the low resource budget under which we are operating here, we have developed a multi-phased algorithm adapting DBSCAN. We are only interested in identifying meaningful spatial zones for a user and to accomplish this we have devised a popularity metric for a location which is computed as follows.

$$P_i = wa_d^i + (1 - w)a_f^i \tag{1}$$

$$a_d^i = \frac{d(a^i)}{d_{max}(a)} \tag{2}$$

$$a_f^i = (1 + log\frac{|S|}{|a_s^i \in S|})^{-1} \tag{3}$$

Here $P_i$ is the popularity of a location $i$, $|S|$ is the total number of locations in the trace within a given time window, and $a_s^i$ is the number of times location $i$ has been visited and $d(a^i)$ is the total duration of the dwell time in that location $i$ and $d_{max}(a)$ is the maximum duration across all locations. $w$ is a weight parameter, and we set it to 0.6 to give higher weight to the dwell time duration. A higher $P$ value indicates the stronger popularity of a location, and a lower value indicates the reverse.

Once we calculate the popularity of all locations, we filter out those that are below the threshold for popularity score. Then these locations are passed to DBSCAN with haversine distance metric, *epsilon* = $100m$ and *min-points* = 1 to construct clusters of meaningful locations. Once this is computed, we ranked them by computing the aggregated popularity of all cluster members and determine the centroid of the cluster as the representative of the zone. After that, when a user is within the 100m radius of a zone's centroid, we ask a question to the user through notifications to qualify whether s/he can be interrupted in that area. A zone is marked accordingly with user feedback, and if a zone is interruptible, then physical and conversational activities are tracked as long as the user is within 100m radius of the centroid of that zone for extracting interruptibility rules. The whole process is illustrated in Figure 2.

## 3.3 Interruptibility Learner

The objective of this component is to determine a set of contextual situations in which a user can be interrupted. In the earlier section, we discussed the discovery of interruptible spatial zones. For learning interruptibility in those zones, we need to identify further the right moments tailored to a user temporal activity trajectory. To this end, we divide a day in four periods: *[morning: 06:00 - 11:59, afternoon: 12:00 - 17:59, evening: 18:00 - 23:59, night: 00:00 - 05:59]*, and create 24 unique situational contexts where each context is described with a combination of one of 4 periods (*i.e., morning, afternoon, evening, night*), one of two conversational activity labels (*i.e., conversation, noise*), and one of the three physical activity labels (*i.e., stationary, walking, on-transport*). The objective of the learner is to attach an interruptibility label against each of these contexts.

There are multiple ways to achieve this objective. Given the simplification of the learning task, we can apply supervised learning techniques with a pre-trained model that can be adapted after the cold-start period. Recent literature has also shown more straightforward associative rule mining with commendable success in learning user preferences [9]. We have decided to adopt the latter considering the low-resource footprint, and the attractive intelligibility property of associative rule miner. However, in contrast to [9], where content is used in open-ended context tuples to learn user preference, we seek to determine how many spatiotemporal activity conditions are interruptible from a fixed set of 24 contexts for each interruptible zone in a content-agnostic way.

To this end, we have used apriori algorithm [2] to learn the context in an automated way. During the cold-start period for each zone, we programmatically interrupt the user for each context condition. The response (*i.e., acknowledge and decline*) to the interruption is then feed into the learning engine to automatically determine the interruptibility label for underlying condition based on *support - how frequently this interruptible context appears in a user trace* and *confidence - how often the context is positively marked for interruptibility.* As we will describe later, we repeat this process together with interruptible zone detection to continuously learn changing user contexts. The outcome of this module is a set of settings defined by specific spatiotemporal physical and conversational activities combinations during which the user is interruptible.

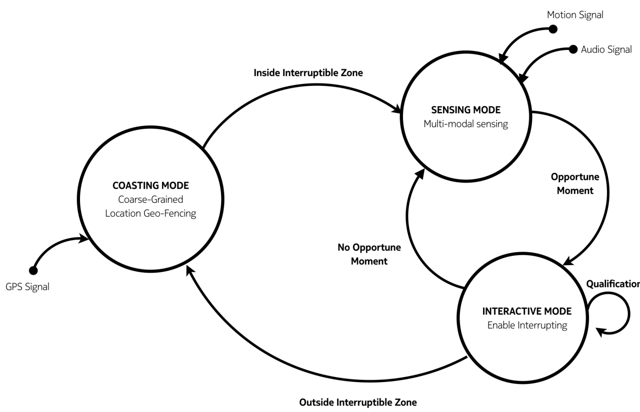


**Figure 3:** **Multi-phased Opportune Moment Detector**

## 3.4 Opportune Moment Detector

This component compares current user situation with the learned behavioural patterns to discover opportune moments to interrupt the user. However given the low-resource budget of a wearable device, this component works in a multi-phased way switching between three different operational modes (as illustrated in Figure 3).

- *Coasting Mode*: In this mode, only coarse-grained location is detected to conserve battery power whenever the user is detected outside of any geofenced region of previously identified meaningful interruptible zones.
- *Sensing Mode*: In this mode, a user is detected to have reached a geo-fenced region of a meaningful interruptible zone and sensors are activated to record spatiotemporal physical and conversational activities which are then matched against the mined ruleset from the interruptibility learner.
- *Interactive Mode*: In this mode, a mined rule is found to match the current trajectory causing the moments to be marked for interruption as necessary. This last mode is also used to interrupt a user to obtain positive or negative feedback on the matched context to mine the rules as explained earlier.

The component exposes these identified moments to the interruptibility manger for relevant actions.

## 3.5 Interruptibility Manager

This module utilises the opportune moments to negotiate with native OS in delivering content at the right moment. This component has a set of alternatives to process an interactive content or notification once intercepted. It can

(1) It can push immediately to a user without considering user's context (the status quo).
(2) it can cancel the content without showing to users (expected to yield negative user experience).
(3) it can delay the content for a specific amount of time; if the opportunity is not identified within a predefined period then push.
(4) it can delay it until the right context is detected, and once the opportune moment is detected the content is pushed to the user.

The first two options are not ideal for the case in context, and the third option is effectively equivalent to option one although it might yield energy conservation [1]. As such, we have selected option four as the guiding principle of this component. However, as we have discussed earlier, the Opportune Moment Detector component works in a multi-phased way in together with the situation recogniser and the interruptibility learner to determine the delivery of content. This is illustrated in Figure4 Essentially, when a user is in coasting mode, all content is ignored as defined by the user. In the sensing mode, a message is only delivered to the user at an opportune moment. Otherwise the content is delayed. If the user switches to coasting mode, e.g., s/he moves to a non-interruptible zone the content is cancelled.

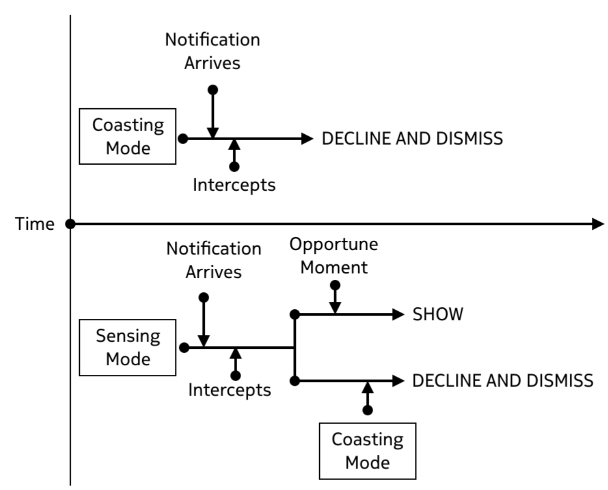


**Figure 4:** **Operational Strategy of Interruptibility Manager**

## 3.6 Online Learning and System Optimisations

One of our design goals is to learn user behaviour continuously with a regular update to the learning model. In this spirit, the situation recogniser re-computes the interruptible zones on a configurable time basis (e.g., every week, every two weeks, etc.) when the wearable is charging. We apply a similar principle to the interruptibility learner. Collectively, this strategy helps us to keep the model fresh and reflective of a user's behavioural pattern.

We have applied several optimisations on the continuous acquisition and processing of sensor data. GPS is only sampled once every 30 seconds, and audio is recorded for 3 seconds in every 30 seconds. The system uses geo-fencing to separate the zones where the device should start sensing from those where it should remain in low-power coasting mode. A threshold at the location of the fence prevents the device to switch between coasting and sensing mode continuously. Also, we have architect our context model applying bleeding-edge acceleration techniques, e.g., compression, GPU-offloading, and processor parallelisation. The details of this techniques are out of the scope of this work.

### 3.7 Implementation

These system components are implemented as a set of Android Services on top of Android Wear (now Wear OS) v1.5, targeting Android Platform 22, potentially working on almost every Wear OS device in circulation. In our case, we have used LG Urbane 2 watch. These services communicate with each other and with the Android Notification Manager via Android Intents and deliver the information contextually.

## 4 EVALUATION

There are several evaluation aspects of the system presented, e.g., performance and efficiency of the context models, situation recogniser, interruptibility learner, moment detector, etc. Given the scope of this paper, we will only reflect on a subset of the evaluations. Future documentation of this work will include a more detailed assessment. In the following, we primarily look at the execution performance and resource footprint of a subset of components of our system. For the benchmark, we have used LG Urbane 2 watch running Wear OS v1.5. LG Urbane 2 watch features a SnapDragon 400 chip (Quad-core 1.2 GHz Cortex A7 and GPU Adreno 305) with 768 MB RAM and 4GB Flash storage and is equipped with 570 mAh battery.

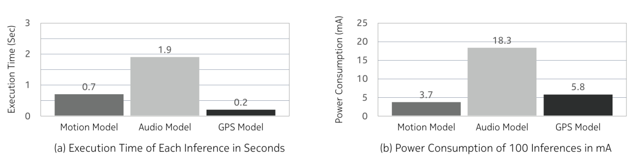


**Figure 5: Execution Performance and Resource Footprint of Different Context Models**

We begin by looking at the (a) execution time and (b) resource footprint of different context models. As illustrated in Figure ref-fig:model, all three models execute efficiently with audio models running CNN yields the highest. Corresponding energy consumption is relatively low. With this footprint, the system's overall battery consumption in the worst case is 36mA per hour under the assumptions that 120 inferences are performed continuously, e.g., in the sensing mode. This corresponds to 6.3% battery life for our host device, i.e., LG Urbane 2 watch.

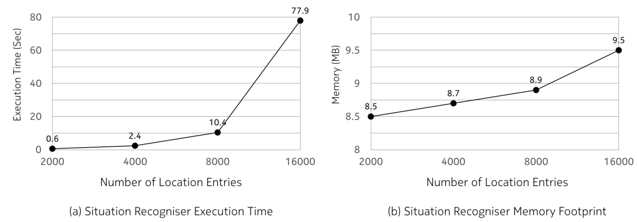


**Figure 6: (a) Execution Time and (b) Memory Footprint of the Situation Recogniser**

Next, we look at the execution time and memory footprint of the situation recogniser. As illustrated in Figure 6 even with 16K entries execution time is 77.9 seconds with only 9.5MB memory cost which we consider very fast and efficient for a wearable host. Similarly, Figure 7 depicts the performance of the interruptibility learner and shows that for 100K rules can be learned under 10 minutes. In practice, we consider this situation will never happen given the constrained set of contexts that we aim to learn. Please note that we do not show the energy footprint of these processes as these components run only when the device is charging.

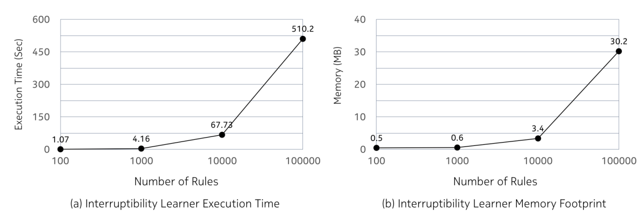


**Figure 7: (a) Execution Time and (b) Memory Footprint of the Interruptibility Learner**


**Figure 8: Crowd-sourcing App for Usability Assessment**

While we envision the primary use of our solution is in wearable notification management, we have evaluated this system in a crowd-sourcing solution with a mobile workforce of a national postal service. Six postal workers wore the watch embedded with our solution for ten days to collect spatiotemporal data (e.g., cleanliness of a street, the safety of a neighbourhood, etc. see Figure 8) opportunistically in addition to their primary task of delivering letters and parcels. In the first eight days, we trained the systems to discover situations and learn personalised rules while the workers

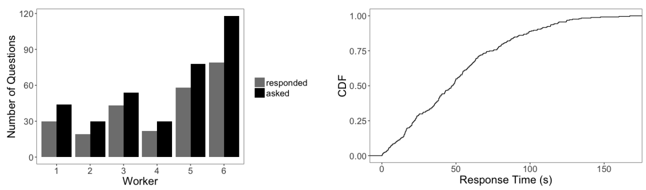


**Figure 9: (a) Response Rate and (b) Response Time of Crowd-sourced app users**

gather data with random interruption and in the last two days, we applied our opportune moment detector and interruptibility manager. Figure 9[a] shows the number of queries presented to each worker and number of queries responded by them in the last two days. While the response rate per worker varies between 63% and 79%, the overall rate is 70%. This is in sharp contrast to the first eight days during which only 24% queries were responded. Figure 9[b] on the other hand shows the cumulative distribution function of response time showing the average response time to queries is 53 seconds.

## 5 RELATED WORK

Many studies have explored the subject of interruptibility in the literature [10]. More recently, several works have studied interruptibility in mobile settings. These works have concentrated on either identifying opportune moments to present the notifications or presenting relevant notifications to users. Mehrotra *et al.* uses a set of classifiers on the content of the notifications [11], whereas InterruptMe uses classifiers on user context data [14] to learn such moments. Okoshi *et al.* argue that the breakpoints in users' lives present opportune moments to interrupt them. They detect these breakpoints using their physical activities and their engagements with the devices [12]. User engagement with the notifications also depends on the modality of the notification [7]. While it is well-known that parameters driven from smart device usage and its embedded sensors can yield information regarding the user context, it has shown that they can be used to indicate user engagement, hence suggesting opportune moments for notification delivery [8]. On relevant notification delivery, PrefMiner learns user preferences for receiving notifications by extracting association rules that are based on both the notification content and user context [9]. Once these are mined, PrefMiner filters out unwanted notifications for users. The effect of push notifications on the energy consumption of mobile devices have been studied in [1] and it has been shown that network-centric heuristics to delay the delivery of the notification can reduce energy consumption by 15%. Building on this rich body of literature, we present a wearable-only, personalised and privacy aware interruptibility management solution with a set of light-weight and efficient algorithms. To best of our knowledge, this is the first system that aims to offer a content-agnostic generic system service for contextual interruptibility management.

## 6 CONCLUDING REMARKS

In this paper, we have described a wearable-only multi-modal system to model interruptibility. We discussed in detail a set of design principles - personalisation, privacy awareness, intelligibility, efficiency, continuous learning - that guided our development. We then described different system components that learn to recognise interruptible situation at a personal scale with active user engagement, and then leverage that learning to deliver contents at the right moment of opportunities. We presented the detail of a set of algorithms that drive these operational behaviours and demonstrated that with a negligible resource cost of 6.3%, our solution could increase user's response rate by 46%. We expect techniques discussed, and insights reported in this work will accelerate the effort of designing interruptibility-aware applications especially in the wearable landscape.

## REFERENCES


[1] U. Acer, A. Mashhadi, C. Forlivesi, and F. Kawsar. Energy Efficient Scheduling for Mobile Push Notifications. In *Proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services on 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MOBIQUITOUS '15, 2015.

[2] R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.

[3] V. Bellotti and K. Edwards. Intelligibility and accountability: Human considerations in context-aware systems. *HumanfiComputer Interaction*, 16(2-4):193–212, 2001.

[4] D. Figo, P. C. Diniz, D. R. Ferreira, and J. a. M. Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal Ubiquitous Comput.*, 14(7):645–662, Oct. 2010.

[5] N. Kern and B. Schiele. Context-aware notification for wearable computing. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, ISWC '03, 2003.

[6] B. Y. Lim, A. K. Dey, and D. Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 2119–2128, 2009.

[7] A. Mashhadi, A. Mathur, and F. Kawsar. The myth of subtle notifications. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, pages 111–114, 2014.

[8] A. Mathur, N. D. Lane, and F. Kawsar. Engagement-aware computing: Modelling user engagement from mobile contexts. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 622–633, 2016.

[9] A. Mehrotra, R. Hendley, and M. Musolesi. Prefminer: Mining user's preferences for intelligent mobile notification management. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 1223–1234. ACM, 2016.

[10] A. Mehrotra and M. Musolesi. Intelligent notification systems: A survey of the state of the art and research challenges. *CoRR*, abs/1711.10171, 2017.

[11] A. Mehrotra, M. Musolesi, R. Hendley, and V. Pejovic. Designing content-driven intelligent notification mechanisms for mobile applications. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, pages 813–824, 2015.

[12] T. Okoshi, J. Ramos, H. Nozaki, J. Nakazawa, A. K. Dey, and H. Tokuda. Reducing users' perceived mental effort due to interruptive notifications in multi-device mobile environments. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, pages 475–486, 2015.

[13] T. Okoshi, K. Tsubouchi, M. Taji, T. Ichikawa, and H. Tokuda. 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)*, pages 100–110, March 2017.

[14] V. Pejovic and M. Musolesi. Interruptme: Designing intelligent prompting mechanisms for pervasive applications. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pages 897–908, 2014.