Trading Energy for Accuracy in Mobile Interruptiblity inference

ALEKSANDAR CUCULOSKI and VELJKO PEJOVIĆ,
Faculty of Computer and Information Science, University of Ljubljana, Slovenia

Untimely interruptions from our mobile devices may have a significant impact on our work performance, stress and well-being, and in critical situations, such as when driving, can even have fatal consequences. State of the art approaches to inferring interruptiblity of mobile users harness an array of sensors available on our devices. Yet, the energy consumption of these sensors clashes with the need to preserve the most precious of the device's resources – its battery charge. In this work we revisit the sensor-based approach to interruptiblity inference and examine the trade-off between a sensor's energy use and its contribution to interruptiblity modelling. Our findings, based on a two week long field study with 14 users demonstrate that turning on additional sensors indeed improves interruptiblity inference, but at a cost of increased energy consumption. We then propose an interruptiblity management systems that uses the classifier confidence as a knob allowing fine-grain tuning along the trade-off front, thus enabling user- and application- specific energy-optimal interruptiblity management.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing systems and tools**; *Empirical studies in HCI*; • **Computer systems organization** → Embedded and cyber-physical systems.

Additional Key Words and Phrases: mobile sensing, interruptiblity, approximate mobile computing

ACM Reference Format:
Aleksandar Cuculoski and Veljko Pejović. 2020. Trading Energy for Accuracy in Mobile Interruptiblity inference. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers (UbiComp/ISWC '20 Adjunct), September 12–16, 2020, Virtual Event, Mexico.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3410530.3414429

1 INTRODUCTION

The growth of mobile computing rendered notifications the most direct and ubiquitous means of capturing the attention of and initiating contact with almost any individual in the world. Yet, as multiple applications, multiple sources of information, and multiple mobile, wearable, and ambient computing devices around us harness notifications, the management of these notification becomes a critical issue for efficient human-computer interaction.

Untimely interruptions from ubiquitous computing notifications can result in disruptions to our work schedules, leading to poor performance [4], can increase stress [12] and anxiety [3], and can even be fatal in certain critical environments, such as cockpits. Consequently, notification management solutions have been developed to prevent notifications from activating at inopportune moments [13, 15, 18, 20]. These solutions almost exclusively rely on sensing capabilities of modern mobile computers. They sense the context of a user and harness machine learning models to infer whether a user is interruptible or not.

Focusing on the attention management side of the experience, however, these solutions remain completely oblivious to the most critical issue hampering the further progress of mobile computing – the limited battery charge of mobile devices. Sluggish advances in the area of battery technology, compared to a rapid growth of the mobile computation, storage, and sensing capabilities, has been noted for more than a decade [5, 16]. Mobile sensors in particular consume a significant amount of charge. Frequent sampling that today's notification management solutions require further exacerbates the energy shortage problem in mobile computing, limiting the real-world applicability of these solutions.

In this work we investigate the discrepancy between the interruptiblity inference capabilities and energy consumption requirements of the context sensing-based attention management approach. We implement a smartphone notification that allows us to perform multimodal sensing together with notification usage monitoring. Using the data collected from 14 users of the above application, we construct machine learning models of user interruptiblity. Using the best performing models we then conduct a study of the trade-off between the informativeness of individual sensor's data for the purpose of interruptiblity inference and the energy cost of sampling that sensor. Finally, we propose a confidence-based method enabling real-time adaptation of the accuracy-energy efficiency performance of the mobile notification management system.

2 RELATED WORK

The effects of interruptions and the means of managing mobile interruptions have been widely explored in the area of human-computer interaction [1, 14, 19]. Survey papers, such as [1] by Anderson et al. examine the theoretical underpinnings and analyze the practical realizations of approaches to better manage interruptions. The meta-analysis provides foundations for sensor-based mobile interruptiblity inference. Such a means of inferring interruptiblity has been attempted by Okoshi et al, who in their mobile sensing system detect breakpoints between physical activities/application usage [15]. The approach is grounded in the theory stating that natural breakpoints between tasks indeed represent good moments to interrupt.

An ever expanding array of sensors with which modern mobile computing devices are equipped resulted in other approaches that use broader contextual information for interruptiblity inference. Pejovic and Musolesi present InterruptMe, as system that samples location, acceleration, and time-related information from smartphone sensors to infer whether a user is going to (positively) react to a notification generated by the system [18]. Zuger and Fritz

UbiComp/ISWC '20 Adjunct, September 12–16, 2020, Virtual Event, Mexico

ACM ISBN 978-1-4503-8076-8/20/09…$15.00
https://doi.org/10.1145/3410530.3414429

expand the used sensing modalities further by introducing physiological sensors sampled from wearable devices that the participants are equipped with [21]. Yuan et al. go beyond context sensing and demonstrate the importance a user's personality for gauging her interruptiblity [20].

To the best of our knowledge, however, none of the above mobile interruptibility inference methods address a pressing issue of energy consumption of frequent mobile sensing. More than a decade ago Pardiso and Starner warned about the sluggish improvement of the battery capacity in contrast to the rapid growth of processing, storage, and communication capacities of mobile devices [16]. Consequently, a call has been issued for more judicious and efficient means of using the available energy. Sensors can have a tremendous impact on battery life, if engaged frequently. The fact that not all sensors consume the same amount of energy, indeed, the difference can be of an order of magnitude or more, and that not all sensed data are equally informative in all situations, opens up the opportunity for improving the energy efficiency through a smarter use of on-device sensors. Hermann et al., for instance, examine the possibility of improving sensor's energy efficiency through the user's context. They propose an application that changes the sampling rate of sensors based on the user's changing context to which they claim that it can improve the battery lifetime by 5x [10]. In our work we build upon such efforts, and in accordance with the philosophy of approximate mobile computing [17], examine the opportunities for selective sensor sampling with the goal of trading the energy use for variable accuracy mobile interruptiblity inference.

3 METHODOLOGY

Despite a number of works related to interruptiblity inference from mobile sensing data, to the best of our knowledge, there are no suitable open datasets that could be used for the examination of the informativeness of different sensors in interruptiblity models. Therefore, our methodology encompasses first, collecting the interruptiblity data through a custom Android application and a two-week long user study, and second, constructing interruptiblity models, gauging sensor informativeness, and analyzing the models' accuracy – energy consumption trade-off.

3.1 Notification Management Behaviour Monitoring App

Our data gathering application is designed to capture all notification-related events on a mobile device. The application is passive, in a sense that it does not create its own notifications, but monitors notifications sent by all other applications installed on the device. While more challenging to implement, this ensures that the dataset we collect is ecologically valid.

There is no standard definition of interruptibility. Therefore, in accordance with the literature standards, we assume that a person is interruptible in a moment of a notification arrival, if they clicked on the notification with the intent of reading the content/using the notification-generating application. In our Android app, for each posted notification, we record the time of arrival, as well as the time of a notification being cleared. There are numerous reasons for a notification being cleared, such as a user dismissing it, a user clicking on it, the app itself clearing it, and others. Starting from Android API 26 it is possible to programmatically get the reason for the notification clear event. For APIs lower than 26 such a functionality is not available and we define the following heuristic: we check the event log and if the application that posted the notification was called to the foreground immediately after the notification was cleared, we label this as a clicked-on notification, i.e. a user being interruptible, otherwise we label it as removed, i.e. a user not interruptible.

We use the AWARE framework [8] to perform comprehensive context sensing both when a notification is posted as well as when it is interacted with. The following sensors are sampled:

- **Battery** we programmatically obtain the battery status – whether the phone was charging or discharging;
- **Light** ambient light value in Lux;
- **Proximity** a binary proximity value correspoding to whether the sensor is covered or not;
- **Screen** a binary value corresponding to screen locked or unlocked;
- **Volume** categorical volume state representing whether the phone is set to Ring, Vibration, Silent or Do Not Disturb;
- **Google Activity Recognition (GAR)** a user's activity as detected by the GAR API classifiers categorized into: still, walking, running, in a vehicle, etc. [2], sampled for a maximum of five seconds;
- **Location** GPS-assisted current location information in longitude and latitude captured using Google Play Location Services, sampled for a maximum of five seconds;
- **Linear accelerometer** uses the accelerometer and gyroscope to determine the users acceleration while ignoring the gravitational acceleration. Sampled for one minute, after which the output is resampled to 2048 sampled for further Fast Fourier Transform (FFT) processing;
- **Bluetooth** a set of nearby Bluetooth devices' IDs sampled over one minute;
- **WiFi** a set of nearby Wi-Fi access points' IDs sampled over one minute;

3.2 Data Collection Campaign

We recruited 19 volunteers that would have the app constantly running for two weeks. The demographic of the users ranged from 19 to 50 years old out of which 12 were students and 7 were employed. 14 users ranged from 19-30, 3 users from 30-40 and 2 users from 40-50 years old. For five users, however, we noticed rather inconsistent behaviour where the data would be collected for a few days only. The Crashalytics monitoring embedded in our application did not report any issues, therefore we could not identify the reason for the missing data. Thus, in the rest of the work we focus on the data from the 14 users for which the data was arriving throughout the study period.

3.3 Data Preprocessing and Descriptive Statistics

In total we have detected 8495 notifications that were interacted with, out of which 5324 were clicked on, while the rest were removed by the user[1]. The distribution of notifications varies over different users, the lowest being 63 notifications, while the highest is 1341 notifications received during the two week period. The percentage of the Clicked notifications varies between 25% and 83% for different users. In Figure 1 we show the cumulative number of notifications in our dataset at each hour of day during the two week period. We see that received notifications follow the expected circadian pattern, where very few notifications are received very late in the night/early in the morning. The response rate increases throughout the day, as people tend to click on notifications more frequently towards the end of the day.

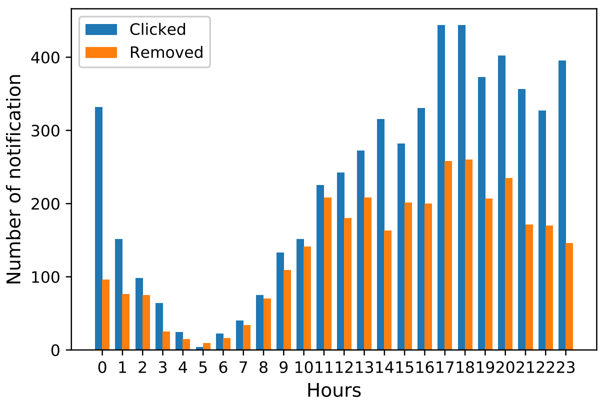

Fig. 1. Cumulative number of notifications at each hour of day for all users.

The more time passes from the point of a notification arrival till the time the notification is handled, the less relevant the sensed context is. Thus, we analysed the time it took users to click on a notification and found out that 99% of the answered notifications were handled within 100 seconds from their arrival. Based on this observation, we limit the time it took a user to answer a notifications to 100 seconds – all notifications that exceeded that time limit were labeled as removed notifications.

Next we examine the relationship between different contextual dimensions and the way users handle notifications. We first preprocess certain dimensions:

Location We use DBSCAN algorithm to find two largest clusters' centroids for data collected during the day and during the night. Data within 200m radius of these centroids are labelled as "HOME" and "WORK" respectively. The remaining are labelled "OTHER". The

Bluetooth The idea behind the Bluetooth sensor is to find fine-grain locations such as an office at work or a user's living room. For each user we calculate a distance matrix capturing pairwise Jaccard distance between lists of BT devices sensed when a notification arrives. We then run a DBSCAN clustering algorithm to extract top N clusters corresponding to most frequent locations described by BT environment, and then label the data with these information. The approach automatically adapts to semantically similar locations – e.g. if a user frequently visits her office, described by a set of Bluetooth devices in the space, an additional Bluetooth device that has not been recorded before, for instance Bluetooth headphones, would not get in the way of recognising the space as the user's office.

Wi-Fi We follow the same procedure as with the Bluetooth sensor, to extract top N Wi-Fi environment locations.

Linear accelerometer We apply the Jigsaw approach for accelerometer feature extraction [11]. The extracted features belong either to the time or the frequency domain. From the time domain we extracted: mean, variance and mean crossing rate. To get the frequency domain we preform the FFT and extract: spectrum peak, sub-band energy, sub-band energy ratio and spectral entropy, where the sub-band energy is calculated on four frequency sub-bands: B1(0-1]Hz, B2(1-3]Hz, B3(3-5]Hz and B4(5-16]Hz. The sub-band ratios were between B1 and B2, B3 and B4, $B1 \cup B2$ and $B3 \cup B4$.

The Bluetooth and Wi-Fi locations are person-specific and will be used for personalised modelling in Section 4. Thus, we do not investigate them further in this section, nor we examine the accelerometer features which are all continuous. Of the remaining dimensions, the most interesting observations pertain to screen, volume, and location data, which we show in Figure 2. We observe that most of the notifications are received when the recipient is at home. Home is also the place where the response rate to notifications is the highest. The finding hints towards the potential importance of the location sensor in interruptiblity inference, despite its relatively high energy cost. The second observation is that the response rate varies with the Volume settings. Here we confirm somewhat surprising findings from previous studies – users tend to be more reachable when the phone is set to Vibration or Silent, compared to phone set to Sound alert. This has been explained by the fear-of-missing-out, a user's internal urge to check for notifications. Furthermore, we see that users tend to not click on notifications when the screen is Locked. This could imply that they either do not see the notifications when the screen is locked, or that the summary shown on the screen (e.g. which app sends the notification) is sufficient to convey the relative importance of the notification. Finally, the descriptive analysis does not point to a single contextual dimension that would be discriminate enough to reliably classify a situation as interruptible or not. Therefore, in the next section we explore machine learning models that use multidimensional input for interrutpiblity classification.

4 INTERRUPTIBLITY MODELLING

The dataset we collected contains rich contextual information (see Section 3.1) together with the information on how a user reacted to a notification. Our approach to modelling is grounded in a well-known observation that certain contextual factors reflect a user's interruptibility [1]. We extract features from the sampled sensor data and use different machine learning algorithms, including K

[1]Unlike earlier work, we focused only on the notifications that a user explicitly interacted with. This is motivated by the observation that nowadays notifications serve for more than actionable interactions – for instance, they inform a user that the battery is full, that a music player is switching to another song, etc. Such notifications are not of interest for the interruptiblity inference addressed in this paper.

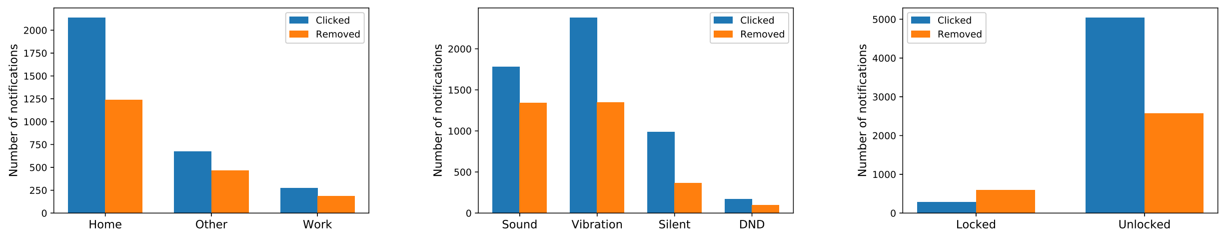

Fig. 2. The distribution of clicked/removed notifications depending on a user's Location (left), Volume settings (center), and Screen state (right).

	Baseline	KNN	NB	SVM	RF	DT	ADA
Accuracy	0.62	0.66	0.72	**0.73**	**0.73**	0.72	0.72
Recall	/	0.76	0.88	**0.92**	0.90	0.91	0.90
Precision	/	0.72	0.71	0.71	**0.72**	0.71	0.71
F1	/	0.74	0.79	**0.80**	**0.80**	0.79	0.79

Table 1. Performance of general models. Sensors used: proximity, screen, volume, battery, and location

	KNN	NB	SVM	RF	DT	ADA
Accuracy	0.81	0.79	**0.84**	0.74	0.80	0.73
Acc. std. dev.	0.08	0.10	0.07	**0.05**	0.09	0.07
Recall	0.95	0.89	**0.97**	0.83	0.96	0.84
Precision	0.78	0.81	0.79	**0.82**	0.78	0.81
F1	0.86	0.85	**0.87**	0.82	0.86	0.82

Table 2. Performance of general models. Sensors used: proximity, screen, volume, battery, and location

Nearest Neighbour (KNN), Naive Bayesian (NB), Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), and AdaBoost (ADA) to predict whether a user is going to click on a notification within 100 seconds of its arrival or not. Our approach represents a validation replication of similar studies, such as InterruptMe. A notable difference is that unlike [18], we collect data pertaining to all applications actually used by our volunteers, rather than generate our own "artificial" notifications.

4.1 General models for interruptibility interface

Interruptiblity models can be constructed jointly for a larger population or they can be tailored to individuals. We first examine general models, as such models are of a greater practical use, since they do not require the data to be collected for each new individual who uses the notification management system. When constructing these models we use all of the sensing modalities except for the WiFi and Bluetooth environment, as they do not represent generalizable data, but unique locations relevant for individual users (e.g. one's office, living room, etc.).

The results of leave-one-person-out evaluation are summarized in Table 1. The results show that RF and SVM-based models performed the best. With the recall above 90% and precision above 70% these two models are significantly more successful than InterruptMe at inferring interruptibility – the best classifier presented in [18] achieves 64% precision and 41% recall. We believe that a dataset based on the actual real-world app notifications, a wider range of features, and an extensive classifier validation procedure contributed to this improvement.

4.2 Personal models for interruptibility interface

Capturing users' unique behaviour with respect to reactions to notifications requires personalised models. The trade-off between using the general and the personal model is that for the personal model we would need to gather data for a certain amount of times, possibly weeks, while a pre-trained general model remains readily usable. However, the inclusion of sensing data that is highly specific to an individual, such as her WiFi and Bluetooth environment, combined with the fact that the model is shaped around a single user is likely to result in a significant boost in the model accuracy.

We build personalised models following a similar train-validate-test approach as in Section 4.1. Instead of leave-one-person out approach, we split an individual user's data so that 70% of the data is used for training and validation, and the remaining 30% for testing. The former set is further split 70-30 into a training and a validation set. The exhaustive feature selection on the validation set yields the following features: screen, volume, battery, location, activity, WiFi, Bluetooth, and acceleration.

The results of the personalised models evaluation are shown in Table 2. The best accuracy result, achieved by the SVM (84%), is more than 10% better than the best accuracy achieved by a general model (73%). As noted in [18], the main goal of a notification management system is to not interrupt at inappropriate times, as we assume that opportunities for interruption are ample. Therefore, we are particularly concerned with the precision of the models, and we observe that RF wins in that respect, achieving 82% precision in our evaluation.

5 ENERGY–ACCURACY TRADE-OFF

Results from the previous section confirm that sensor data can indeed be used to infer a user's interruptiblity. Yet, the data comes at a cost since sensors represent one of the major consumers of a mobile device's energy [10]. To the best of our knowledge, the investigation on the cost of mobile sensing for interruptibility inference and the

Sensor	Curr. drawn (mA)	Sense t (s)	Total (mAh)	Src
Acc	80	60	1.33	[9]
BT	25	60	0.42	[6]
WiFi	100	60	1.67	[7]
Loc	140	13	0.5	[9]

Table 3. Energy spent by different smpartphone sensors.

exploration (as well as the exploitation) of the trade-off between the energy use and interruptiblity inference accuracy has not been attempted, yet. In this section we first discuss the energy cost of interruptibility sensing and then propose a practical method for adapting the interruptiblity inference according to the given energy efficiency and/or accuracy requirements.

Not all sensing is necessarily power hungry. Of the information used for interruptiblity models presented in Section 4 sensing battery charge, screen state, volume, proximity, light, and activity, through Google Activity Recognition service, incurs negligible energy cost. Sensing other modalities, such as accelerometer, Bluetooth, WiFi, and GPS incurs non-negligible cost, yet, as apparent from Table 3 the cost of sensing varies across different sensors.

5.1 Confidence-Based Energy-Efficient Inference

Theoretically, in a given situation we can reduce the energy used for interruptiblity inference by relying on sensors that are sufficient to give us a correct answer and consume as little energy as possible. Yet, we do not know in advance whether the answer provided by a model using a selected subset of sensors was correct or not. Relying on an observation that certain classifiers, such as Random Forest and Naive Bayesian, provide the information about *the confidence in the inference* we propose a method that each time a notification is about to be sent out iteratively queries the interruptibility model and turns on additional sensors, until the model achieves the desired inference confidence level.

Our proposed method works as follows. First, low-cost sensors – battery, screen, volume, proximity, light, and activity recognition – are sampled. The information is fed to the model (RF or NB) and the predicted outcome together with the confidence is calculated. The desired confidence level is previously selected and if the model does not reach the given level, additional sensing of accelerometer, Bluetooth, WiFi, and the GPS location is conducted iteratively. We select this order based on the energy cost of the above sensors, yet any other order is possible, depending on the sensor availability and the cost of sensing on different platforms. Iterative querying stops once the confidence threshold is reached, or once all the sensors are used for the inference.

To evaluate the potential of the method, we restrict our dataset to six users whose data contained all sensing modalities[2] and construct a personalised model for each user. We use 70% of a user's data for model training and 30% for the testing in a trace-based simulation behaving according to the approach described in the previous paragraph. We experiment with different desired inference confidence level ranging from 50% to 90% and observe the change in the actually achieved inference accuracy in the simulation as well as the additional energy needed for the inference calculated over all data points in the simulation. Note that the proposed approach at each encountered notification instance dynamically decides whether to use an additional sensor or not, i.e. a certain target accuracy level does not imply that the same set of sensors will be turned on at each notification point.

The results of our simulation are shown in Figure 3. The higher the target confidence level is, the greater the achieved interruptiblity inference accuracy is. Additional energy is shown normalised to the difference between the case where all sensors are used and where only the baseline sensors are used. We observe that in all cases RF delivers higher accuracy for the given additional energy than NB. Furthermore, we see that our confidence level-based approach achieves the highest accuracy (88% with RF set to 90% target confidence level) with about 30% less additional energy used compared to a traditional approach that relies on all the sensors at each data point. If energy efficiency is a priority, our method also allows for judicial use of additional energy – with less than 10% normalised additional energy our method improves the accuracy from 80% to 83% in case of RF. Even more importantly, our method enables dynamic system tuning with a single software-based parameter – inference confidence: when energy is scarce, the target confidence can be loosened to preserve energy, when energy is not an issue (e.g. a phone is predicted to be charging soon), the target confidence level is increased to achieve a higher inference accuracy.

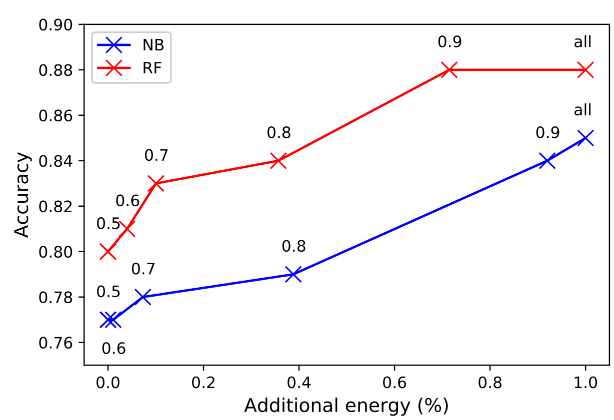

Fig. 3. **Energy vs. Accuracy trade-off.** Numbers next to the points indicate the requested classifier confidence. The higher the requested confidence is, the higher the interruptiblity classifier accuracy is, at the expense of additional energy.

6 DISCUSSION, LIMITATIONS, AND IMPLICATIONS

In Section 5.1 we present, to the best of our knowledge, the first practical method for energy-aware interruptiblity inference. The

[2]Our data collection app allows users to switch of certain sensing modalities they are not comfortable with. Consequently, a number of users in our study disabled GPS, Bluetooth or WiFi sensing for a period or the whole duration of the study.

method's reliance on a single knob – inference confidence – makes it oblivious to the actual underlying machine learning model, as long as the said knob value can be calculated. Besides Random Forest and Naive Bayesian, we envision our approach being used with more advanced deep learning models, where the final layer's softmax output could be used for the confidence calculation. The method can also be used irrespective of whether the underlying model allows missing values at the input. If it does, we can maintain a single model and supply an input vector containing only the data pertaining to the currently sampled sensors. Otherwise, we can maintain multiple models corresponding to different combinations of sensing modalities.

With respect to limitations, we note that in this paper we performed an evaluation over a pre-collected trace. While this directly translates to practical situation when a non-critical notification arrives and the system has to decide whether to show it to a user or not, one can also envision a different use case where a suitable moment for interruption needs to be identified within a certain timeframe (e.g. up until 60 minutes from now). In such a case, the interruptiblity needs to be frequently re-evaluated. The existing solutions rely on periodic sensing [18] or context change detection [15]. In future we plan to expand our method to efficiently monitor the change of inference confidence. We believe that reinforcement learning-based approach grounded in sensor informativeness is a promising approach for tackling this problem.

7 CONCLUSIONS

In this work we have revisited the context-aware approach to mobile interruptiblity inference. We have confirmed that context, as sensed by mobile devices, indeed reflects a user's interruptiblity. Our personal models achieve up to 84% accuracy, outperforming previous work our approach is based on. We then investigate the interplay of sensor sampling energy consumption and the accuracy of the models constructed from different subsets of the sensed data. Relying on a machine learning model confidence as a knob, we propose trade-off control mechanism that allows fine-grain tuning between the achieved inference accuracy and the energy use. We believe that our approach brings context-aware interrutpiblity inference a step closer to a wider practical adoption.

8 ACKNOWLEDGMENTS

The authors would like to than the study participants. This work was supported by Slovenian Research Agency (grants no. N2-0136 and P2-0098).

REFERENCES

[1] Christoph Anderson, Isabel Hubener, Ann-Kathrin Seipp, Sandra Ohly, Klaus David, and Veljko Pejovic. 2018. A Survey of Attention Management Systems in Ubiquitous Computing Environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2 (Jul 2018), 1–27. https://doi.org/10.1145/3214261

[2] Activity Recognition API. [n.d.]. URL: https://developers.google.com/location-context/activity-recognition. [Accessed: 13. 9. 2019].

[3] Brian Bailey, Joseph Konstan, and John Carlis. 2004. The Effects of Interruptions on Task Performance, Annoyance, and Anxiety in the User Interface. *INTERACT* (04 2004).

[4] Brian P. Bailey and Joseph A. Konstan. 2006. On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in Human Behavior* 22, 4 (2006), 685 – 708. https://doi.org/10.1016/j.chb.2005.12.009

[5] Fehmi Ben Abdesslem, Andrew Phillips, and Tristan Henderson. 2009. Less is More: Energy-Efficient Mobile Sensing with Senseless. In *Proceedings of the 1st ACM Workshop on Networking, Systems, and Applications for Mobile Handhelds*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/1592606.1592621

[6] Igor Crk, Fahd Albinali, Chris Gniady, and John Hartman. 2009. Understanding Energy Consumption of Sensor Enabled Applications on Mobile Phones. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference* 2009 (09 2009), 6885–8. https://doi.org/10.1109/IEMBS.2009.5333609

[7] WiFi Scan Energy. [n.d.]. URL: https://source.android.com/devices/tech/power/values. [Accessed: 8. 5. 2020].

[8] Denzil Ferreira, Vassilis Kostakos, and Anind K Dey. 2015. AWARE: mobile context instrumentation framework. *Frontiers in ICT* 2 (2015), 6.

[9] Jeff Sharkey Google I/O. [n.d.]. URL: http://www.kandroid.org/board/data/board/AndroidBeginner/file_in_body/1/W_0300_CodingforLife-BatteryLifeThatIs.pdf. [Accessed: 9. 5. 2020].

[10] Ralf Herrmann, Piero Zappi, and Tajana Simunic Rosing. 2012. Context aware power management of mobile systems for sensing applications. In *IPSN*, Vol. 12. 16–20.

[11] Hong lu, Jun Yang, Zhigang Liu, Nicholas Lane, Tanzeem Choudhury, and Andrew Campbell. 2010. The Jigsaw continuous sensing engine for mobile phone applications. *SenSys 2010 - Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, 71–84. https://doi.org/10.1145/1869983.1869992

[12] Gloria Mark, Daniela Gudith, and Ulrich Klocke. 2008. The Cost of Interrupted Work: More Speed and Stress. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 107–110. https://doi.org/10.1145/1357054.1357072

[13] Abhinav Mehrotra, Robert Hendley, and Mirco Musolesi. 2016. PrefMiner: mining user's preferences for intelligent mobile notification management. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 1223–1234.

[14] Abhinav Mehrotra and Mirco Musolesi. 2020. Intelligent Notification Systems. *Synthesis Lectures on Mobile and Pervasive Computing* 11, 1 (2020), 1–75.

[15] Tadashi Okoshi, Jin Nakazawa, and Hideyuki Tokuda. 2014. Attelia: Sensing user's attention status on smart phones. *UbiComp 2014 - Adjunct Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (09 2014), 139–142. https://doi.org/10.1145/2638728.2638802

[16] Joseph Paradiso and Thad Starner. 2005. Energy Scavenging for Mobile and Wireless Electronics. *Pervasive Computing, IEEE* 4 (02 2005), 18 – 27. https://doi.org/10.1109/MPRV.2005.9

[17] Veljko Pejović. 2019. Towards approximate mobile computing. *GetMobile: Mobile Computing and Communications* 22, 4 (2019), 9–12.

[18] Veljko Pejovic and Mirco Musolesi. 2014. InterruptMe: Designing Intelligent Prompting Mechanisms for Pervasive Applications. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Seattle, Washington) *(UbiComp '14)*. Association for Computing Machinery, New York, NY, USA, 897–908. https://doi.org/10.1145/2632048.2632062

[19] Liam D Turner, Stuart M Allen, and Roger M Whitaker. 2015. Interruptibility prediction for ubiquitous systems: conventions and new directions from a growing field. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*. 801–812.

[20] Fengpeng Yuan, Xianyi Gao, and Janne Lindqvist. 2017. How busy are you? Predicting the interruptibility intensity of mobile users. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 5346–5360.

[21] Manuela Züger and Thomas Fritz. 2015. Interruptibility of Software Developers and Its Prediction Using Psycho-Physiological Sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2702123.2702593