User-centric Context Inference for Mobile Crowdsensing

Yifan Du, Valerie Issarny, Francoise Sailhan

User-Centric Context Inference for Mobile Crowdsensing

Yifan Du, Valérie Issarny
{firstname.lastname}@inria.fr
MiMove Team
Inria Paris, France

Françoise Sailhan
francoise.sailhan@cnam.fr
Cedric Laboratory
CNAM Paris, France

ABSTRACT

Mobile crowdsensing is a powerful mechanism to aggregate hyperlocal knowledge about the environment. Indeed, users may contribute valuable observations across time and space using the sensors embedded in their smartphones. However, the relevance of the provided measurements depends on the adequacy of the sensing context with respect to the phenomena that are analyzed. This paper concentrates more specifically on assessing the sensing context when gathering observations about the physical environment beyond its geographical position in the Euclidean space, i.e., whether the phone is in-/out-pocket, in-/out-door and on-/under-ground. We introduce an online learning approach to the local inference of the sensing context so as to overcome the disparity of the classification performance due to the heterogeneity of the sensing devices as well as the diversity of user behavior and novel usage scenarios. Our approach specifically features a hierarchical algorithm for inference that requires few opportunistic feedbacks from the user, while increasing the accuracy of the context inference per user.

KEYWORDS

Context Inference, Sensing Context, Mobile Crowdsensing

1 INTRODUCTION

Mobile crowdsensing empowers ordinary citizens to contribute data sensed or generated from their mobile devices [3, 8]. It allows acquiring hyperlocal knowledge at scale, thanks to the proliferation of mobile devices and the ubiquity of wireless broadband connection. Numerous sensor types embedded in today's smartphones contribute valuable quantitative observations about the urban environment (e.g., noise, temperature, atmospheric pressure, humidity, light, magnetism). The observations further come along with the related spatial and temporal data, which allows for the analysis of hyper-local environmental knowledge. However, the quality of the contributed measurements challenges the aggregation of such a knowledge, which depends on the *accuracy* of the contributing sensors and the adequacy of the *sensing context* [20]. Addressing the latter requires a supporting inference mechanism, which is the focus of our paper.

The accurate monitoring of the physical environment through crowdsensing obviously requires knowing the "location" of the sensing device with respect to the phenomenon being observed. However, the location must not be limited to the geographical coordinates in the Euclidean space. Indeed, the "user behavior at the location" has a significant impact on the quality of the quantitative observations contributed through mobile crowdsensing [9]. To know whether the smartphone/sensing device is in-/out-pocket, in-/out-door and under-/on-ground is particularly important because the smartphone/sensing device needs to be in a position that

enables -yet does not interfere with- sensing the physical characteristics of the surrounding [15]. The sensing context must distinguish between in-pocket and out-pocket observations because the former leads to a quite significant deviation from the ground truth and are thus not readily usable [14]. The same applies to in-door versus out-door measurements since aggregating them together to analyze environmental phenomena obviously leads to unreliable results [12]. Similarly under-ground and on-ground scenarios contribute observations that must be distinguished. Figure 1 illustrates the impact (significant bias) of the above elements of the "sensing context" on physical measurements that are collected at the "same" geographical location. The context information allows keeping more observations -and even correcting them- for aggregating environmental knowledge, rather than filtering out drastically the crowdsensed data [5].

The existing work on the inference of the crowdsensing context focuses on a single context element, while it is essential to accurately characterize the sensing context as a whole, that is, to identify whether a contributed observation is sensed with the device: in-pocket/out-pocket, in-door/out-door and under-ground/on-ground. Machine learning is an obvious candidate to systematize such a characterization. The challenge is then to ensure that the classifier accounts for the diversity of the crowd contributors. Indeed, there is a large variation in the characteristics of the contributing devices, in the behavior of the contributing users, and even in the usage scenarios. As a result, the classifier for context inference must be adaptive to each participating user, considering the user's behavior, device and contribution scenarios. This paper introduces such a user-centric approach to the inference of the sensing context, where online learning allows customizing the classifier on the user's device, while minimizing the user's involvement that is essential to motivate the engagement of a sufficiently large crowd. The paper contribution is as follows:

- We derive the features that best serve classifying each sensing context, taking into account the sensors available on today's smartphones. We then analyze the performance of candidate updatable learning algorithms to initialize the three resulting classifiers, taking into account their accuracy as well as their runtime and memory efficiency (§ 3).
- The personalization solution follows, which includes the hierarchical inference of all three context elements (in-/out-pocket, in-/out-door and under-/on-ground) that are relevant to environmental monitoring using crowdsensing, and the opportunistic update requiring very few feedbacks from the user (§ 4).
- We validate the effectiveness of the approach using simulation to assess the accuracy of the context inference in relation with the negative user feedback. We confront our
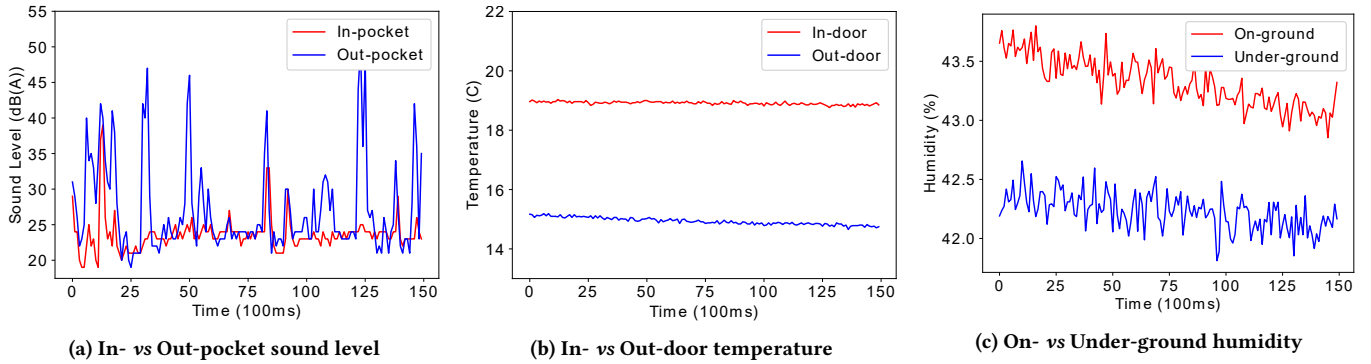
(a) In- *vs* Out-pocket sound level   (b) In- *vs* Out-door temperature   (c) On- *vs* Under-ground humidity

Figure 1: Impact of the sensing context on environmental observations

hierarchical solution with a multi-class classifier and we also assess its energy and runtime efficiency (§ 5).

To the best of our knowledge and as surveyed in the next section, no other work on the inference of the sensing context addresses the classification of the three context elements simultaneously, and deals with the efficient personalization of the classifiers on the crowdsensing device.

2 RELATED WORK

Inferring the sensing context with respect to the physical environment under scrutiny has deserved little attention, while related solutions focus on a single context element and consider different eligible features.

Features for context inference. Detecting whether a phone is in-pocket/bag or out-pocket/bag may be inferred using various embedded sensors. Reference [19] leverages features from embedded proximity and light sensor signals, while reference [14] uses additional data from a 3-axis accelerometer. Inferring whether an observation is made in-door or out-door may also rely on various sensors. For instance, previous work simply utilizes the GPS signal [2, 14]. Alternatively, the sensing service in [6] leverages three on-board sensing resources, i.e., light sensors, magnetic sensors, and cell tower signals. A similar solution is adopted in [10] although using the proximity instead of the magnetic sensor. As for the detection system in [7], it uses two features, i.e., the received signals of Wifi and the light density. The solution in [1] leverages these two features as well as activity recognition, while the approach in [16] exclusively uses radio signals and requires the signal strength of four neighboring GSM base stations. Comparatively, under-/upper-ground classification has deserved much less attention. Here, the quick barometric pressure change is suitable for detecting in-door floor transition [11], and underground scenario in particular as the air pressure allows estimating the altitude [4].

As outlined above, diverse embedded sensors may serve characterizing the context of the mobile sensing, while the resulting inferences differ with respect to their accuracy and energy-efficiency. The analysis of the state of the art further shows that the most relevant observations are those related to the physical and communication environment. Still, it is critical to account for the variation in the availability of features across the contributing devices. Our

approach addresses this requirement by being self-adaptive to the diversity of users (and devices). The user-centric classification on the user's device may be performed on any subset of the above set so as to account for the diversity of devices and related embedded sensors. The set may further be customized and freely recomposed, trading off energy efficiency and accuracy, to offer adaptiveness.

Generic context inference. Various algorithms have been considered for the inference of the mobile sensing context, among which rule-based solutions. A classifier based on the non-linearity of the sensor readings is presented in [19] for the in-pocket/out-pocket classification. Rule-based classifiers are also introduced for the in-door/out-door case but differ in the number of sub-detectors and their approach to aggregation. The algorithm in [6] uses three sub-detectors while the one in [7] uses two sub-detectors. The approach in [1] is also rule-based but hierarchical and switches among three rule-based modules so as to distinguish the contributing scenarios at a finer grain. Independent of the specifics of the rule-based approach, adequate thresholds for the classifier must be set, which is done empirically [2, 10, 11]. An alternative to the rule-based approach for the context inference is to leverage machine learning algorithms. For instance, Rana *et al.* [14] use the K-Nearest Neighbor algorithm (KNN) with the trained classifier being evaluated using cross validation. A range of machine learning algorithms for the classification of in-door *vs* out-door is investigated in [16], including the Decision Tree, Random Forest, Support Vector Machine, KNN, Logistic Regression, Naive Bayesian, and Neural Network; they conclude that KNN performs the best with an average accuracy.

As outlined above, the existing work on context inference focuses on the classification of a single element of the sensing context. Further, they evaluate the accuracy using a single dataset. None of them addresses the personalization of the classifiers so as to cope with the diversity of the contributing devices, users and scenarios. Differently, our solution solves three classification problems together, and is implemented as an on-device service providing the sensing context to crowdsensing applications. We emphasize the customization of the context inference per user, which requires an online learning algorithm to keep the classifier evolving on the local device.

Towards the personalized inference of sensing contexts. Supervised learning for one analyzed environment does not always translate to another environment. Thus, the baseline learning model needs to evolve according to the environment in which it is used. For instance, the user-specific touch input model in [17] is updated using calibration input requested to the users. The method for indoor/out-door detection in [13] employs semi-supervised machine learning without user involvement to learn new information. It uses co-training classifiers requiring two Naive Bayes, resulting in a double computational cost on the device. Although the estimation of the class of a new instance does not involve the user, it is not the ground truth. Rather than excluding user involvement, we believe that the opportunistic participation of the user to gather the ground truth allows a more effective personalization of the classifiers.

We introduce three classifiers, *aka*, learning models, to comprehensively characterize the mobile context, that is, whether the sensing device is: in-pocket/out-pocket, in-door/out-door, and under-ground/on-ground. We denote the corresponding classifiers M_{pocket}, M_{door}, and M_{ground}. We specifically leverage an online learning approach to deal with the diversity of the contributing devices, users and usage scenarios, while offering a resource-efficient solution that minimizes the required user participation.

3 CLASSIFIER INITIALIZATION

Following the above state of the art analysis and taking into account the sensors embedded in today's smartphones, we find out the most relevant feature set for each of our three classifications using the dataset $DATASET_1$. $DATASET_1$ contains $20k$ labeled entries contributed by a single user; it includes all the candidate features and covers uniformly the three contexts to be classified.

3.1 Feature Selection

As illustrated in Table 1, not all the features are relevant to each of the three classifications M_{pocket}, M_{door} and M_{ground}. We select the features that induce both high information gain and gain ratio.

Feature set for M_{pocket}. The top three features for the M_{pocket} classification are proximity, temperature and light density. Proximity is widely used for the in-pocket detection. Light density remains an obvious, although less significant, candidate. While temperature is an additional relevant feature, not all devices are able to provide it.

Feature set for M_{door}. The M_{door} classification uses GPS accuracy, abstract RSSI, GSM RSSI, Wifi RSSI, light density and temperature. Although GPS is the most relevant feature to distinguish in- and out-door scenarios, it is also *the most* power consuming feature. Rather than totally disabling it, it can be occasionally used depending on the preference of the user. Wireless RSSI shows a lower, but still significant, contribution to classification because it is impacted by walls and obstacles (Non-line-of-sight) in buildings.

Feature set for M_{ground}. Even though the under-ground scenario is considered as a sub-case of the in-door scenario, it requires abstract RSSI, GPS accuracy, temperature, GSM RSSI, pressure, Wifi RSSI and humidity. Pressure and humidity are decisive as long as the device provides them. The wireless RSSI contributes because the network connectivity is usually weak in an under-ground compared to on-ground environment.

3.2 Initial Training

We train the M_{pocket}, M_{door} and M_{ground} classifiers using the most significant features associated with each of them. Although performing a training for each smartphone/sensor model would lead to a better classification, this is hardly feasible given the diversity of smartphones/sensors. Our classifiers must be effective both in terms of classification accuracy and time/space cost, especially with respect to their local inference/update on the device. There are various algorithms eligible to address the classification problem although fewer are updatable. We have specifically selected six candidate updatable algorithms: *Hoeffding Tree (Very Fast Decision Tree), IBk (Instance Based K-nearest neighbors classifier), KStar (Instance-based Learner), LWL (Locally Weighted Learning), updatable Naive Bayes, and SGD (Stochastic Gradient Descent)* [18]. Using $DATASET_1$, we have compared the candidate algorithms for M_{pocket}, M_{door} and M_{ground} according to the following metrics: *Size* -the serialized model size; *CVCA* -the 10-fold Cross Validation Classification Accuracy; *OLR* -the Online Learning Runtime and *IR* -the Inference Runtime.

Due to the lack of space, we provide in Table 2 the result obtained with M_{ground}; same trends are observed for M_{pocket} and M_{door}. In particular, all the algorithms provide a high CVCA. Depending on the algorithm, the sterilized size significantly differs because IBk, KStar and LWL are storing the training instances in the learning model, so the size of the classifier gets proportional to the size of the training dataset. IBk and LWL have a greater OLR compared to others. IBk, KStar and LWL all have much longer IR than Hoeffding Tree, Naive Bayes and SGD. The OLR and IR remain low for Hoeffding Tree, Naive Bayes and SGD. We finally selected the Hoeffding Tree as training algorithm because it is characterized by the highest accuracy and lowest space/time costs. The initial classifier is trained once on a computer.

4 ONLINE PERSONALIZATION

The classifiers for the sensing context need to be personalized, i.e., deployed on the smartphones so as to evolve according to the specifics of the device, user and even scenario contributing the crowdsensed observations. In particular, the online learning and classifiers update cope with the following aspects:

(1) *Biases in the feature value across diverse device models:* The values collected for the same feature on different devices can show significant biases due to the diversity of device models.
(2) *Availability of features depending on the device and user preference:* The feature availability depends on both the capability of the given device and the configuration set by the owner who decides which embedded components are switched on/off.
(3) *Classification on new scenarios not covered during the initial training:* The classification may not work in a new physical region, another season, another city, etc. In other words, the initial classifier may not cover scenarios that will be encountered across time and space.

Metric	Feature	Proximity	Temperature	Light density	GPS accuracy	Abstract RSSI	GSM RSSI	Wifi RSSI	Pressure	Humidity
M_{pocket}	info. gain	0.931	0.344	0.310	-	-	-	-	-	
	gain ratio	0.720	0.172	0.155	-	-	-	-	-	
M_{door}	info. gain	-	0.228	0.255	0.974	0.794	0.738	0.320	-	-
	gain ratio	-	0.114	0.127	0.715	0.493	0.370	0.180	-	-
M_{ground}	info. gain	-	0.485	-	0.434	0.547	0.463	0.181	0.376	0.276
	gain ratio	-	0.243	-	0.318	0.340	0.232	0.170	0.188	0.138

Table 1: Potential features and information gain / gain ratio above 0.1

Metric/Model	H.Tree	IBk	KStar	LWL	Naive Bayes	SGD
Size (kB)	13	1763	1763	1763	4	6
CVCA (%)	100	100	100	98.060	97.105	100
OLR (ms)	0.024	4.628	0.062	6.720	0.009	0.111
IR (ms)	0.061	15.160	238.916	128.149	1.223	0.018

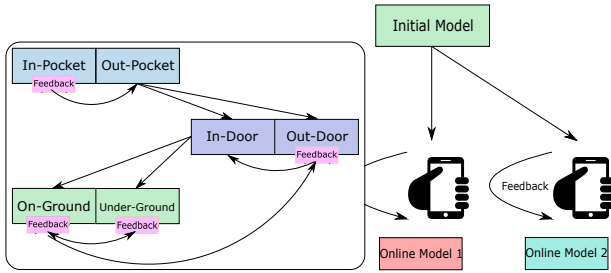Table 2: Evaluation of the learning algorithms for M_{ground}

Figure 2: Online learning for personalization - The initial model is deployed on the participating devices, and becomes a local user-centric model on each device, which is evolving through feedbacks; The feedback is collected only following some inference results and the inferences are hierarchical

Figure 2 summarizes the design rationale of our online learning solution: the initially trained classifier is deployed on the participating devices at the time of the installation of the embedding crowdsensing middleware/application, e.g., [5]. While the inference of the sensing context is running on the device, feedback is requested to the user to assess the correctness of the inference result. The collected opportunistic feedback is then converted to a labeled training instance that updates the current learning models. We boost the online learning by creating a new instance from the feedback and using it multiple times for the model update. The feedback requirement should be limited as much as possible to minimize the burden on the user, while still enhancing the accuracy of our three classifiers M_{pocket}, M_{door} and M_{ground} over time. We achieve this by applying a hierarchical inference and update of the three classifiers (Algorithm 1). This follows from the predominant role of the in-pocket classifier over the two others and of the in-door classifier over the under-ground one when sensing the physical environment. In more detail, a crowdsensed measurement is relevant for the analysis of most environmental phenomenon if out-pocket; an in-pocket device has less opportunity to be contributing to the mobile crowdsensing, e.g., [15]. The in-door/out-door detection is

---

Algorithm 1 Hierarchical inference and update

Input: A feature vector $\vec{f}$ with a feedback $u \in U$ (options)
Output: The integrated inference result c shown to the user
$\quad u \in \emptyset \subset U$ *possible if the user does not give feedback*
1: $\quad c \leftarrow null$
2: $\quad$ **if** $M_{pocket}(\vec{f}) = inpocket$ **then**
3: $\quad\quad c \leftarrow inpocket$ **and** $U \leftarrow \{outpocket\}$
4: $\quad\quad$ **if** $u = outpocket \in U$ **then**
5: $\quad\quad\quad$ update M_{pocket} using instance $(\vec{f}, u)$
6: $\quad\quad$ **end if**
7: $\quad$ **else if** $M_{door}(\vec{f}) = outdoor$ **then**
8: $\quad\quad c \leftarrow outdoor + outpocket$ **and** $U \leftarrow \{indoor\}$
9: $\quad\quad$ **if** $u = indoor \in U$ **then**
10: $\quad\quad\quad$ update M_{door} using instance $(\vec{f}, u)$
11: $\quad\quad$ **end if**
12: $\quad$ **else if** $M_{ground}(\vec{f}) = underground$ **then**
13: $\quad\quad c \leftarrow underground + indoor + outpocket$ **and** ...
$\quad\quad\quad U \leftarrow \{onground\}$
14: $\quad\quad$ **if** $u = onground \in U$ **then**
15: $\quad\quad\quad$ update M_{ground} using instance $(\vec{f}, u)$
16: $\quad\quad$ **end if**
17: $\quad$ **else**
18: $\quad\quad c \leftarrow onground + indoor + outpocket$ **and** ...
$\quad\quad\quad U \leftarrow \{underground, outdoor\}$
19: $\quad\quad$ **if** $u = underground \in U$ **then**
20: $\quad\quad\quad$ update M_{ground} using instance $(\vec{f}, u)$
21: $\quad\quad$ **else if** $u = outdoor \in U$ **then**
22: $\quad\quad\quad$ update M_{door} using instance $(\vec{f}, u)$
23: $\quad\quad$ **end if**
24: $\quad$ **end if**
25: $\quad$ **return** c **and** run from line 1 for next loop

---

meaningful only when the device is out-pocket and ready for sensing. Further, the under-ground/on-ground case is a sub-scenario of the in-door situation. Also, while requesting the user's feedback about a single inference may be acceptable, requesting the feedback about three inferences is too much to ask.

5 IMPLEMENTATION AND EVALUATION

We have implemented the proposed online personalization as an Android application. We rely on the WEKA library [18] that implements the Hoeffding Tree algorithm. The initial classifier is serialized and imported into the Android application, so as to be loaded at runtime and updated locally. The application collects

features, infers the sensing context, collects feedbacks and updates the classifiers accordingly. The application also implements a feature extractor and a user interface for requesting feedbacks. The final solution relies on the Hoeffding Tree and uses only negative feedbacks hierarchically so as to keep the amount of feedbacks to a minimum and limit the burden put on the end user.

5.1 Accuracy and Number of Feedbacks

We use a new dataset, $DATASET_2$, which includes $20k$ labeled entries collected from a single user and that uniformly covers the combination of the 3 contexts. Compared to $DATASET_1$, $DATASET_2$ collects data at different places and time periods, and uses a different device model whose sensors come from distinct manufacturers; further, during the data collection, the user switched off/on the Wifi and the GPS modules. We assess the accuracy of our hierarchical algorithm according to the number of negative user feedbacks (i.e., when the inference is wrong). We performed 500 runs of experiment where, at each run, we randomly select 30 entries of $DATASET_2$ and the ground truth serves assessing the correctness of the inference. Over the 500 runs, at most 9 inferences were wrong for M_{pocket}, 12 for M_{door}, and 13 for M_{ground}. Also, as depicted in Figure 3, the classification accuracy gets better 99%, 86% and 71% of the time over the 500 runs for M_{pocket}, M_{door} and M_{ground}, respectively. The enhanced accuracy for M_{pocket} and M_{ground} is high compared to the initial accuracy, and is less significant for M_{door}. While the accuracy for M_{pocket} remains stable, the classification accuracy for M_{door} and M_{ground} slightly varies before reaching 91% and 90%, respectively. Summarizing, our hierarchical algorithm enhances the classification accuracy most of the time while limiting the amount of (negative) feedbacks requested to the end user.

5.2 Hierarchical vs Multi-class Classifiers

An alternative to our approach lies in performing a single and multi-class classification, which distinguishes 8 combinations of in/out-pocket, in/out-door and under/on-ground. To compare the multi-class classifier with our hierarchical classifiers, we performed 100 runs of experiment using the settings of § 5.1. As illustrated in Figure 4, the mean of the enhanced accuracy (triangles in plot) always remains below 45%. Also, the initial and enhanced classification accuracy of a multi-class classifier is 1/2 time lower compared to hierarchical classifiers (Figure 3). Besides, the multi-class classifier requires more feedbacks compared to the hierarchical classifiers: Up to 28 feedbacks are required. Also, with the multi-class classifier, the user must select among 7 options instead of one or two options. As illustrated in Figure 5, our hierarchical classifier involves a much shorter update and a slightly shorter inference compared to the multi-class classifier because the hierarchical classifiers do not always perform all the classifications.

Overall, using hierarchical classifiers offers many advantages: (1) Using a specific classifier for each context element implies that the classification accuracy remains high for each of them; (2) Each classifier only relies on the most relevant features, which reduces the inference and update execution time; (3) A classifier is easily added/removed/replaced when a new context element is handled for the benefit of the crowdsensing application; (4) Hierarchical classifiers limit the number of inferences that are triggered; (5)

The user feedback required for the personalization of hierarchical classifiers is simple and reduced.

5.3 Energy and Resource Efficiency

The power consumed by the GPS module, proximity sensor and Wifi component is relatively high (resp. 50, 3 and 2 mA) compared to the one consumed by the light/ magnetic/pressure/humidity sensor, which remains bellow 1.2 mA. Figure 6 evaluates the impact of disabling these three power-consuming components for which we rely on another dataset, $DATASET_3$, that includes $20k$ entries obtained when these three components are disabled. When the GPS, proximity and Wifi are disabled, the classification accuracy drops down with a decrease of respectively 50%, 40% and 20%. With feedbacks, the mean classification accuracy increases by 10%, 8% and 1% respectively, while the maximum classification accuracy with feedbacks reaches 74%, 99% and 98% respectively. Overall, our approach personalizes the classifiers and deals with (power-consuming) features that may be disabled by the user.

Our implementation requires $100MB$ of memory on a smartphone (Qualcomm Snapdragon 636) and leads to an increase of 5% of the CPU. The inference and update of contexts necessitate around $3MB$ of memory and an increase of 5% of the CPU. The inference execution time is on average $0.2ms$, $0.1ms$ and $0.1ms$ for M_{pocket}, M_{door} and M_{ground}, respectively, while the execution time necessary to update the model is $7.3ms$, $7.5ms$ and $10.0ms$ on average. Summarizing, our approach allows adapting the tradeoffs between power consumption and accuracy, while inducing limited resource consumption on the smart phone.

6 CONCLUSION

Context-awareness is essential to the development of mobile crowdsensing applications that can be informed when a suitable sensing context is detected so as to maximize the effectiveness of the crowdsensing application without requesting the end-user to explicitly provide contextual details. This paper proposes to classify the context under which the crowdsensor operates and infer whether the smartphone is in/out-pocket, in/out-door and under/on-ground. We leverage online supervised learning to deal with the diversity of devices, the different user behaviors and novel scenarios. We compare six online learning models to select the algorithm achieving the best efficiency, which is expressed in terms of classification accuracy, runtime and memory consumption. In particular, we introduce a hierarchical algorithm that requests few feedbacks and hence reduces the burden put on the user. Experiment results show that despite few feedbacks, there is a high probability to get a good/enhanced classification accuracy. Furthermore, compared to a single multi-class classifier, our hierarchical approach of context inference and update is more efficient in terms of both execution time and feedback collection. Our approach is also flexible because the sensing context can be inferred even when some sensor(s) or communication module(s) are unavailable or switched off.
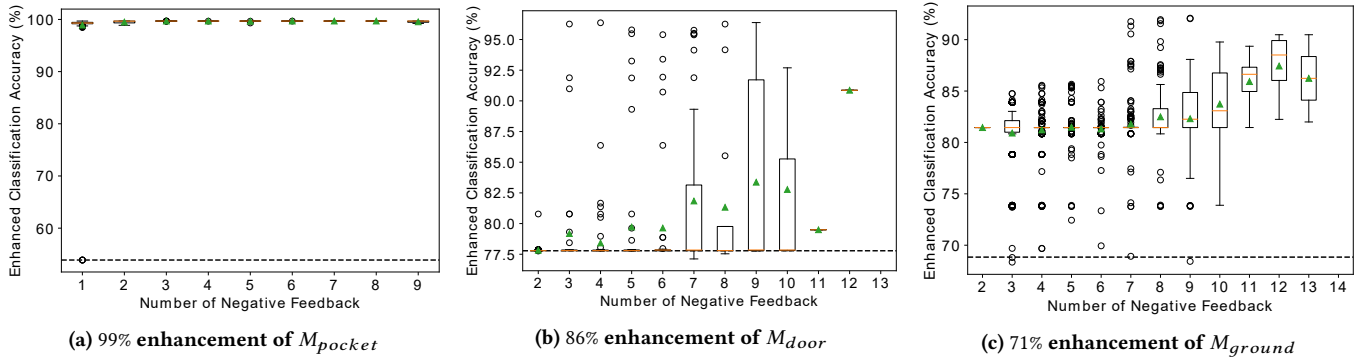
(a) 99% **enhancement of** M_{pocket}

(b) 86% **enhancement of** M_{door}

(c) 71% **enhancement of** M_{ground}

Figure 3: Box plots showing the improvement of the hierarchical H.Tree classification accuracy *wrt* the number of feedbacks compared to the original accuracy obtained without feedback (dashed line)
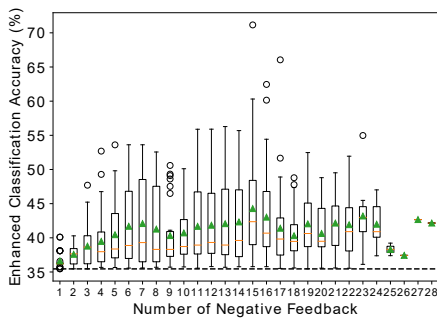

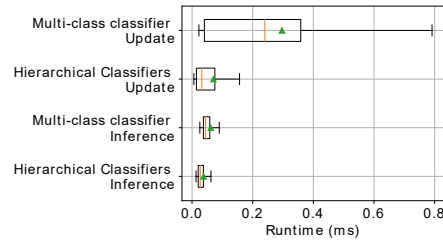**Figure 4: Multi-class H.Tree classification accuracy box plot according to the number of feedbacks**

Figure 5: Execution time of single multi-class H.Tree *vs* hierarchical binary-class H.Trees
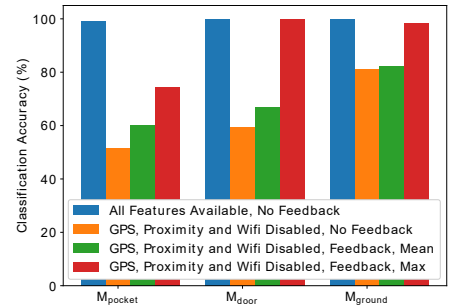

Figure 6: Classification accuracy under different power modes

REFERENCES

[1] M. Ali, T. ElBatt, and M. Youssef. 2018. SenseIO: Realistic Ubiquitous Indoor Outdoor Detection System Using Smartphones. *IEEE Sensors* 18, 9 (2018).

[2] K. Chen and G. Tan. 2017. SatProbe: Low-energy and fast indoor/outdoor detection based on raw GPS processing. In *IEEE INFOCOM*.

[3] B. Guo, Z. Wang, Z. Yu, Y. Wang, NY. Yen, R. Huang, and X. Zhou. 2015. Mobile Crowd Sensing and Computing: The Review of an Emerging Human-Powered Sensing Paradigm. *Comput. Surveys* 48, 1 (2015).

[4] S. Hyuga, M. Ito, M. Iwai, and K. Sezaki. 2015. Estimate a user's location using smartphone's barometer on a subway. In *Proc. ACM MELT*.

[5] V. Issarny, V. Mallet, K. Nguyen, PG. Raverdy, F. Rebhi, and R. Ventura. 2016. Dos and don'ts in mobile phone sensing middleware: Learning from a large-scale experiment. In *Proc. ACM Middleware*.

[6] M. Li, P. Zhou, Y. Zheng, Z. Li, and G. Shen. 2014. IODetector: A Generic Service for Indoor/Outdoor Detection. *ACM Transactions on Sensor Networks* 11, 2 (2014).

[7] S. Li, Z. Qin, H. Song, C. Si, B. Sun, X. Yang, and R. Zhang. 2017. A lightweight and aggregated system for indoor/outdoor detection using smart devices. *Future Generation Computer Systems* (2017).

[8] J. Liu, H. Shen, and X. Zhang. 2016. A Survey of Mobile Crowdsensing Techniques: A Critical Component for the Internet of Things. In *IEEE ICCCN*.

[9] S. Liu, Z. Zheng, F. Wu, S. Tang, and G. Chen. 2017. Context-aware data quality estimation in mobile crowdsensing. In *IEEE INFOCOM*.

[10] Z. Liu, H. Park, Z. Chen, and H. Cho. 2015. An Energy-Efficient and Robust Indoor-Outdoor Detection Method Based on Cell Identity Map. *Procedia Computer Science* 56 (2015).

[11] D. Luo, H. Luo, and C. Zili. 2015. An Indoor Scene Recognition Algorithm Based on Pressure Change Pattern. In *IEEE ICICTA*.

[12] MK. Marina, V. Radu, and K. Balampekos. 2015. Impact of Indoor-Outdoor Context on Crowdsourcing based Mobile Coverage Analysis. In *Proc. ACM AllThingsCellular*.

[13] V. Radu, P. Katsikouli, R. Sarkar, and MK. Marina. 2014. A semi-supervised learning approach for robust indoor-outdoor detection with smartphones. In *Proc. ACM SenSys*.

[14] R. Rana, CT. Chou, N. Bulusu, S. Kanhere, and W. Hu. 2015. Ear-Phone: A context-aware noise mapping using smart phones. *Pervasive and Mobile Computing* 17 (2015).

[15] R. Ventura, V. Mallet, V. Issarny, PG. Raverdy, and F. Rebhi. 2017. Evaluation and calibration of mobile phones for noise monitoring application. *The Journal of the Acoustical Society of America* 142, 5 (2017).

[16] W. Wang, Q. Chang, Q. Li, Z. Shi, and W. Chen. 2016. Indoor-Outdoor Detection Using a Smart Phone Sensor. *Sensors* 16, 10 (2016).

[17] D. Weir, S. Rogers, R. Murray-Smith, and M. Lüchtefeld. 2012. A user-specific machine learning approach for improving touch accuracy on mobile devices. In *Proc. ACM UIST*.

[18] IH. Witten, E. Frank, MA. Hall, and CJ. Pal. 2017. *Data mining practical machine learning tools and techniques*. Morgan Kaufmann.

[19] J. Yang, E. Munguia-Tapia, and S. Gibbs. 2013. Efficient in-pocket detection with mobile phones. In *Proc. ACM UbiComp*.

[20] O. Yurur, CH. Liu, Z. Sheng, VCM. Leung, W. Moreno, and KK. Leung. 2016. Context-Awareness for Mobile Sensing: A Survey and Future Directions. *IEEE Communications Surveys & Tutorials* 18, 1 (2016).