Noise-Sensing Using Smartphones: Determining the Right Time to Sample

Willian Zamora
Departament of Computer Engineering (DISCA)
Universitat Politècnica de València
Valencia, Spain
wilzame@posgrado.upv.es
Universidad Laica Eloy Alfaro de Manabí
Manta, Ecuador

Carlos T. Calafate
Departament of Computer Engineering (DISCA)
Universitat Politècnica de València
Valencia, Spain
calafate@disca.upv.es

Juan-Carlos Cano
Departament of Computer Engineering (DISCA)
Universitat Politècnica de València
Valencia, Spain
jucano@disca.upv.es

Pietro Manzoni
Departament of Computer Engineering (DISCA)
Universitat Politècnica de València
Valencia, Spain
pmanzoni@disca.upv.es

ABSTRACT

In this paper, we propose to use smartphones as an environmental sensor to measure noise pollution. We focused our study on determining the precise context for the capture of environmental noise through smartphones, and performed an analysis of the impact that the sensing task collection will have on energy consumption. To this purpose, we define different contexts to determine whether adequate environment sampling conditions are met, and we then apply classification algorithms to generate the most accurate decisions trees automatically. An analysis of resource consumption requirements associated with the different trees obtained shows that, despite their high accuracy, the resource consumption levels were prohibitive for this kind of applications. Thus, we propose an alternative decision tree that maintains the accuracy levels of automatically generated trees while significantly reducing the resource consumption introduced by the latter. Experimental results show that our proposed decision tree can reduce the energetic impact of our target application by about 60% when compared to the optimum theoretical tree generated through automatic classification procedures.

CCS CONCEPTS

• **Computing methodologies → Classification and regression trees**;

KEYWORDS

CrowdSensing; smartphone; noise sensing; decision tree; weka

MoMM '17, December 4–6, 2017, Salzburg, Austria

ACM ISBN 978-1-4503-5300-7/17/12…$15.00
https://doi.org/10.1145/3151848.3151868

ACM Reference Format:
Willian Zamora, Carlos T. Calafate, Juan-Carlos Cano, and Pietro Manzoni. 2017. Noise-Sensing Using Smartphones: Determining the Right Time to Sample. In *MoMM '17: The 15th International Conference on Advances in Mobile Computing & Multimedia, December 4–6, 2017, Salzburg, Austria.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3151848.3151868

1 INTRODUCTION

Noise pollution is a serious problem for human beings because it can affect their health both physically and psychologically. In fact, the World Health Organization emphasizes that the effects of excessive noise are irreversible [1]. Also, different organizations and countries have regulated and stressed the importance of the control of environmental noise on the life quality of the population [2, 7].

The traditional method to determine the noise level in the environment is through professional sonometers (Sound Level Meter); this type of device has a built-in microphone having high accuracy, but they usually remain at static positions to study noise variations along time, failing to provide a complete view of noise in a city.

On the other hand, technological advances, together with the increasing integration of new sensors on smartphones (e.g. ambient light, accelerometer, proximity), have enabled novel solutions to emerge, being crowdsensing the most prominent one. In particular, participatory solutions such as NoiseTube [6], NoiseSPY [4] and EarPhone [9] use smartphones to measure environmental noise levels in urban areas, and then generate pollution maps by combining this information with geolocation data.

Smartphone context awareness is critical to determine the right time and place to capture the noise, avoiding samples with little or no representativeness. Also, another factor to consider in the design of smartphone applications is battery consumption. Considering the aforementioned issues, the solutions of crowdsensing for ambient noise readings requires new approaches to data collection that can reduce the level of authorization required by users and maximize energy efficiency.

In this paper, we focus on application design issues; specifically, we attempt to detect which is the most appropriate time and context to obtain these noise samples, while simultaneously minimizing

power consumption. To achieve it, we start by first analysing different usage contexts, and then attempt to obtain decision trees [5] able to simultaneously maximize decision accuracy while maintaining resource usage at a minimum. The results obtained show that, compared to automatically generated solutions, our optimized decision tree is able to reduce smartphone resource consumption by 60%.

The paper is organized as follows: In section 2 we introduce the proposed archictecture for crowdsensing solutions. Then, in section 3, we define the contexts and algorithms used to automatically determine the best sampling time, and describe the tool used to obtain the candidate trees. The procedure followed to obtain an alternative decision tree able to optimize resource consumption while maintaining the desired decision accuracy, is then detailed in section 4. Finally, section 5 presents our conclusions and refers to future works.

2 CROWDSENSING ARCHITECTURE

In a previous paper [12] we proposed a generic crowdsensing architecture. In particular, we defined a set of elements and an infrastructure that allow monitoring noise pollution at different places, and in a transparent manner. Our proposal integrates two types of elements: Mobile Noise-Sensing Clients (MNSC) and Cloud Data Collection (CDC). Those two elements are connected to each other through a data transmission network. Particularizing our proposed architecture to the scope of the present work, the MSNC is the mobile phone or the set of mobile phones that will provide sound sensing operations by capturing noise data, and that will then deliver that data to the CDC. Regarding the CDC, it is a single server or a server farm that allows receiving, processing, analyzing, and sharing the sensed data. In this work, we will focus on the MNSC, and in particular we will focus on how to determine the right time and context to capture meaningful noise samples while avoiding an excessive resource usage. This architecture is shown in Figure 1. In our architecture, we define a module that is responsible for analyzing the right time to undertake the sensing task. So, once a task notification is received from the server, it will evaluate the user context to determine if the required conditions are met, and if so then proceed with the sensing task. In our proposed architecture, this module belongs to the Client Sensor Manager (CSM), which is in charge of managing the smartphone sensors. The other modules, CDM, CIM, and CCM, are parts of our overall proposal, and are further detailed in [12].

3 CLASSIFICATION TECHNIQUES

In this section, we define a series of context analysis and actions undertaken at the smartphone at the time of assessing whether the moment and context is adequate for noise capturing. In addition, we define the algorithms used for classification, and present the obtained results.

3.1 Attribute Context

There are certain conditions that should be considered when using smartphones as a noise measurement tool. For instance, there are several situations that make it inadequate to sense environmental noise, as user intervention is affecting the overall result. Examples

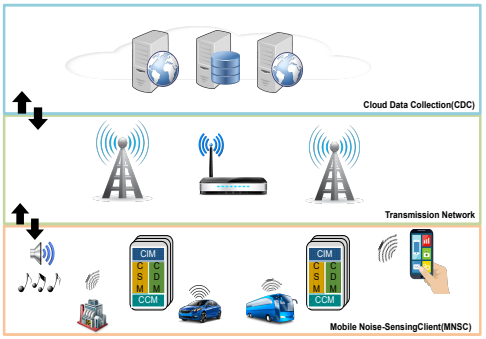


Figure 1: Overview of the proposed crowdsensing architecture.

Table 1: Attribute taxonomy.

Categories	Attribute
Task	TaskDate, TaskArea
Sound	Speaker, Music, ActiveCall, MicroPhone, ActiveAplication
Status	PhoneStatus, Locked, Camera, Keyboard, Location

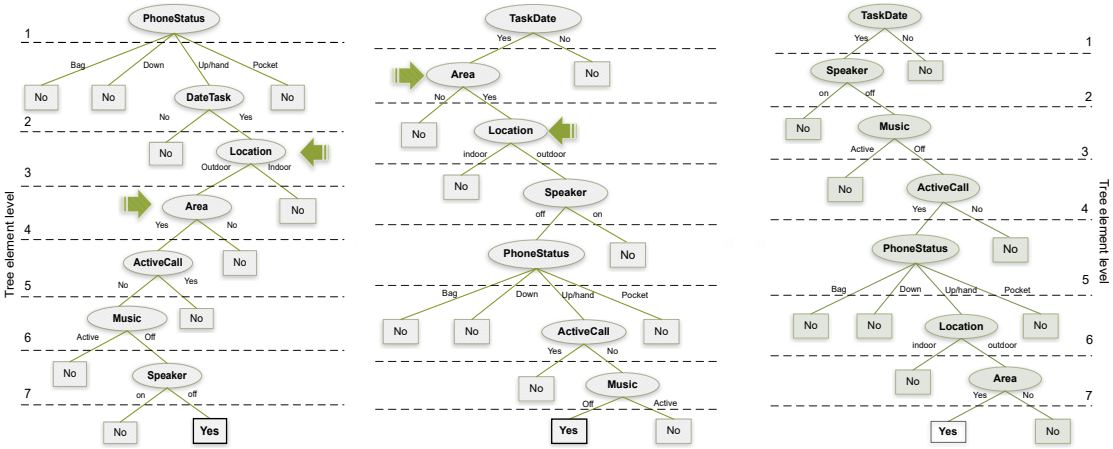
of such situations include: talking on the phone, listening to music using the loudspeaker, or keeping the smartphone in a pocket/purse.

In this paper we seek to assess the adequacy of a particular situation to take environmental noise samples. Specifically, we want to optimize the sequence followed to access the different built-in states and sensors (i.e. GPS, accelerometer) for such assessment to be made. We define a series of attributes that contribute to determining whether the ideal conditions for environmental noise sampling are met. These attributes are shown in Table 1, and were classified according to the conditions and characteristics of the smartphone. For the first category, we consider the tasks as a set of actions to be taken by the smartphone at a time and location with the aim of measuring noise pollution (e.g., measure the noise level in the centre of Valencia, from 17:00 pm to 18:00 pm). In particular, it defines the task of sensing. Regarding the second category, it refers to the attributes that produce a sound phenomenon that can affect the noise readings. The third category refers to the conditions of the smartphones (e.g., the phone can be active or idle). These two last categories refer to the optimal instant to perform the measurement.

Based on these 13 nominal attributes defined above, we have generate a list of all possible combinations (16384 cases), being all of them tagged as admissible or not from the perspective of environmental noise assessment.

3.2 Classification Algorithm

In this work we have used a decision tree as the methodology to classify the different contexts defined. In particular, we have used two algorithms known as J48 [8] and random tree (RT) [3] that are able to achieve accurate estimations. We relied on the implementation of these algorithms provided by the Weka tool [11].



(a) J48 algorithm. (b) Random Tree algorithm. (c) Proposed resource-efficient tree.

Figure 2: Visualization of the automatically generated trees.

This tool provides outputs that allow finding the best algorithm for data classification for different datasets.

3.3 Weka-based automatic classification

Using as input the 16384 contexts referred in Section 3.1, we relied on the Weka tool to provide an automatic classification of these different cases. As an output, Weka generated two decision trees, one using the J48 algorithm, and another one using the RandomTree algorithm.

Figure 2 (a) and (b) shows the obtained decision trees. Regarding their accuracy, the J48 algorithm achieves a 100% accuracy, while for the RandomTree algorithm, the accuracy achieved is slightly reduced (99.89%), with an absolute error of 0.001. Overall, it is worth mentioning that attributes *ActiveAplication, Locked, Microphone, Keyboard*, and *Camera* have been discarded as being unnecessary by these algorithms.

From a resource consumption perspective, we can observe that the location-related attributes are located in the fourth node of both trees, as signalled by the red arrows. This leads us to think that the resource consumption associated to these decision trees can be excessively high. In short, these two trees offer a viable theoretical solution, but they are not optimal from a software design perspective, and they do not seem at all optimal regarding time and energy consumption. So, in the next section, we will analyze the different issues involved in order to properly optimize the decision process; in particular, we will propose an alternative decision tree that is more resource efficient than its automatically generated counterparts (see Figure 2 (c)).

4 TASK SEQUENCING OPTIMIZATION

Once the candidate trees were obtained, our next objective was to determine the optimal strategy for collecting noise pollution data through smartphones. To achieve this purpose, in this section we will first analyze the computation time associated to each tree element. Secondly, we will analyze the energy consumption required. Finally, we will present our proposal for a balanced tree in terms of computation time and energy savings.

4.1 Computation Time

To analyse the computation time associated to each particular task, a specific application was developed to allow evaluating each attribute individually, and following a repeatable and reliable procedure. In general, 100 independent readings were obtained for each attribute to be measured, and we took their average value. A Samsung S7 Edge model running the Android 7.0.2 operating system was used for testing. Below we detail some relevant characteristics of the most critical attributes:

For the *Task*, *TaskDate*, and *TaskArea* attributes, the developed application reads the data from an internal database (i.e., SQLite), and then these values are instantiated in a class for later use. We assume that the server application had previously sent these tasks to the smartphone, and so they are available for processing. In particular, the TaskArea attribute was implemented as a class that compares a polygon of n sides with the current position given by the GPS sensor. This class returns true if the smartphone is located inside such polygon.

Regarding the *Speaker*, *Music*, and *ActiveCall*, attributes, the developed application works by making calls to the corresponding API offered by the operating system, and to analyse the reply returned.

PhoneStatus. The implementation of this attribute was carried out through a service that reads the proximity, light, and accelerometer sensors. In particular, this service ends when the results are obtained. Notice that we rely on the results of a previous study [9, 10] to determine, based on the sensor feedback, whether the smartphone is being held, it is stored in a backpack, or it is in a pocket. So those previous results allow us to estimate the complexity of such predictions.

Location. The location attribute is considered a critical factor because of its high consumption and processing time, meaning that a more detailed analysis is required. We implemented a service where we read the latitude and longitude of the GPS sensor embedded

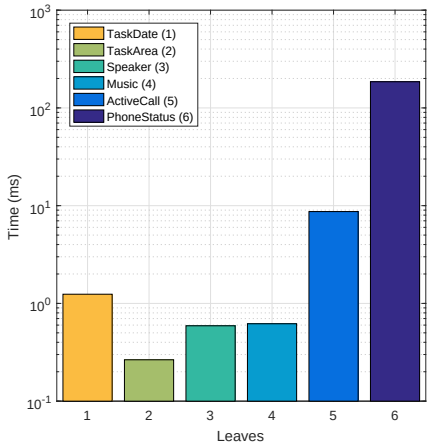


Figure 3: Processing time for the different tree elements.

in the smartphone. Also, we recorded the prediction accuracy to have a greater control of the positions obtained. A time-stamp was recorded when the GPS obtained the first coordinate. In particular, the design of our solutions aim at outdoor locations alone, meaning that we will also use the GPS to discriminate between both cases (indoor vs. outdoor). To assess the ability of the GPS sensor to differentiate between both environments, we first analyzed the accuracy results achieved inside a building (near a window to get worst case conditions), as well as outdoors, in an open environment. For each case, 30 records were taken at two different times: mid-morning, and mid-afternoon. Our goal is checking whether the readings obtained through our application show differentiating features for these environments. Overall, we find that 99% of the location measurements are obtained in less than 4000 ms, with just sporadic values found in the 4 to 6 second range. We also find that a GPS accuracy of 40 meters or less is typically only obtainable outdoors, while indoors the accuracy is typically much higher, thus allowing to clearly discriminate between both cases. Hence, based on these results, we are able to validate attribute *location* in the scope of our tree, and we will set its duration to 4000 ms, as its provides the necessary trade-off between consumption and accuracy.

Finally, Figure 3 shows the computation times associated to each key element of the tree (excluding the *Location* parameter). In this figure, it is clearly noticeable that the *PhoneStatus* attribute is the one consuming more resources in our tree i.e., 185 ms, followed by the *ActiveCall* attribute, that needs about 9 ms.

4.2 Energy consumption

After determining the computation times associated to each decision attribute, we then proceeded to analyse the energy consumption of the different decisions trees. A background service was implemented on the smartphone that is periodically reading the different attributes of the proposed tree; for all cases we set the sampling period to 4 seconds. Before each test, it is verified that the smartphone's battery is at 100%, and that Internet access is disabled. To obtain representative results, the evaluation lasted for 1 hour.

Table 2: Estimation of energy consumption for each tree level with the different algorithms used.

Tree element	Random tree (μAh)	J48 (μAh)
1	0.00354	5.2761
2	0.0076	0.0354
3	160.00	160.00
4	0.0168	0.0076
5	5.2761	0.2479
6	0.2479	0.0176
7	0.0176	0.0168

Table 2 presents as a summary the energy consumption estimation associated to each attribute on the tree. In particular, we can observe that the atributes corresponding to the GPS and PhoneStatus present the highest energy consumption values.

4.3 Proposed tree and performance improvement

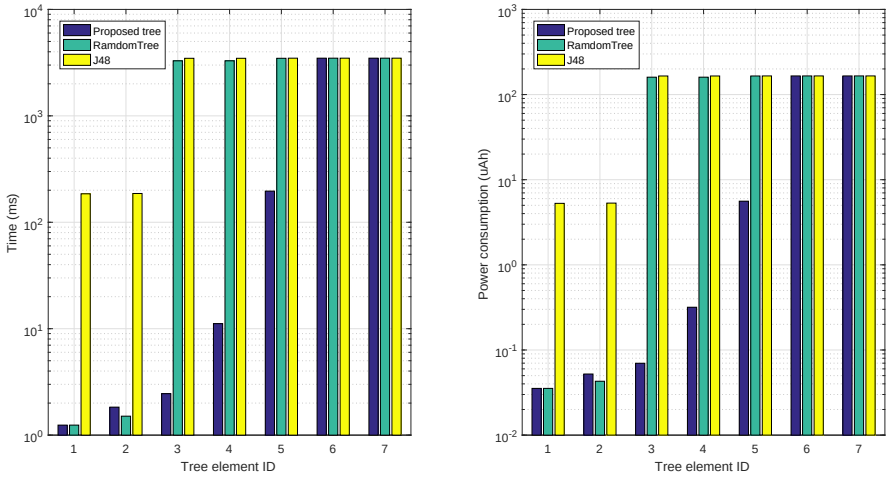
Once we obtained the computation time overhead and the energy consumption associated to each attribute, our next goal was to propose an alternative decision tree that is more resource efficient. For this purpose, we designed a tree in such a way that its elements are organized and balanced according to the desired objective of reducing time and energy overhead. In particular, for the *TaskDate, Speaker, Music, ActiveCall, PhoneStatus,* and *Location* attributes, we followed a sequential order by considering the computation time calculated earlier. Figure 2 (c) shows the proposed tree which, similarly to the J48 algorithm, is able to achieve a decision accuracy of 100%. Notice that, in this alternative tree, the *location* attribute is located near the tree bottom, thereby optimizing the overall system resources whenever a previous attribute allows discarding the noise sampling procedure by not meeting the required conditions. Besides, we can observe that the *area* attribute remains just below the *location* attribute due to its direct dependency, being this an attribute with a lower computation time, but nevertheless highly relevant in terms of the final decision.

Finally, Table 3 summarizes the performance benefits introduced by our alternative decision tree. In particular, it shows both the accumulated and average values for the time overhead and energy consumption associated to the three decision trees being compared. We find that our tree is 58% and 59% lower than the RandomTree algorithm in terms of computation time and average energy consumption, respectively. For the J48 tree, improvements are further boosted to 60%, while maintaining the same decision accuracy.

Overall, we consider that the process followed in this paper to achieve a tree that is both precise and resource efficient is critical to enable the development of a crowdsensing application aiming at a widespread adoption and usage, an issue to be addressed in the next research steps.

5 CONCLUSIONS AND FUTURE WORK

In this article, we proposed to use smartphones as environmental noise sensors as part of a complete crowdsensing architecture. In



(a) Time leaves. **(b) Energy consumption leaves.**

Figure 4: Time and energy consumption at different tree levels.

Table 3: Estimation of computational requirements and energy consumption for the different decision trees.

Total # elem.	Algorithm	CPU time (ms)		Energy (μAh)	
		$\sum$	avg	$\sum$	avg
7	Proposed tree	3483.89	897.72	165.60	42.16
7	RandomTree	3483.89	2127.84	165.60	102.09
7	J48	3483.89	2244.44	165.60	105.41

particular, we addressed the issue of optimizing the decision process that precedes actual noise sampling by determining whether the required conditions are met or not. With this purpose, we first defined a set of contexts for the smartphones, and then, through different algorithms, we automatically generated two decision trees able to meet the decision requirements. However, we found that a theoretical classification does not necessarily provide a balanced tree in terms of computation overhead and energy consumption. Through a detailed analysis of these two performance factors, we determined which attributes had the most negative impact on resources, and then proposed a tree redesign so as to improve resource consumption as much as possible.

Experimental results show that our proposal obtained a relative saving of nearly 60% in terms of both energy consumption and computing overhead, while maintaining the same accuracy as the best decision tree obtained through automated systems (J48).

As future work, we plan to integrate our proposal in a full crowdsensing architecture to achieve distributed noise measurements.

ACKNOWLEDGMENT

This work was partially supported by the "Programa Estatal de Investigación, Desarrollo e Innovación Orientada a Retos de la Sociedad, Proyecto TEC2014-52690-R"Spain, and the "Universidad Laica Eloy Alfaro de Manabí," and the "Programa de Becas SENESCYT de la República del Ecuador."

REFERENCES

[1] 2010. WHO | Chapter 4. *WHO* (2010). http://www.who.int/whr/2002/chapter4/en/index8.html
[2] European Environment Agency. 2016. Noise European Environment Agency. (2016). https://www.eea.europa.eu/themes/noise/intro
[3] Hanen Bouali and Jalel Akaichi. 2014. Comparative Study of Different Classification Techniques: Heart Disease Use Case. *2014 13th International Conference on Machine Learning and Applications* (2014), 482–486. https://doi.org/10.1109/ICMLA.2014.84
[4] Eiman Kanjo. 2010. NoiseSPY: A Real-Time Mobile Phone Platform for Urban Noise Monitoring and Mapping. *Mobile Networks and Applications* 15, 4 (aug 2010), 562–574. https://doi.org/10.1007/s11036-009-0217-y
[5] G. Kesavaraj and S. Sukumaran. 2013. A study on classification techniques in data mining. *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)* (2013), 1–7. https://doi.org/10.1109/ICCCNT.2013.6726842
[6] Nicolas Maisonneuve, Matthias Stevens, Maria E. Niessen, and Luc Steels. 2009. NoiseTube: Measuring and mapping noise pollution with mobile phones. In *Environmental Science and Engineering (Subseries: Environmental Science)*, Paulina Golinska, Marek Fertsch, and Jorge Marx-Gómez (Eds.). Environmental Science and Engineering, Vol. 3. Springer Berlin Heidelberg, Berlin, Heidelberg, 215–228. https://doi.org/10.1007/978-3-540-88351-7_16
[7] NIOSH. 2013. NIOSH Publications and Products - Occupational Noise Exposure (98-126). (06 2013). http://www.cdc.gov/niosh/docs/98-126/
[8] Tina R Patil and S S Sherekar. 2013. Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification. *International Journal Of Computer Science And Applications* 6, 2 (2013).
[9] Rajib Rana, Chun Tung Chou, Nirupama Bulusu, Salil Kanhere, and Wen Hu. 2015. Ear-Phone: A context-aware noise mapping using smart phones. *Pervasive and Mobile Computing* 17, PA (feb 2015), 1–22. https://doi.org/10.1016/j.pmcj.2014.02.001 arXiv:1310.4270
[10] Muhammad Shoaib, Stephan Bosch, Ozlem Incel, Hans Scholten, and Paul Havinga. 2015. A Survey of Online Activity Recognition Using Mobile Phones. *Sensors* 15, 1 (2015), 2059–2085. https://doi.org/10.3390/s150102059
[11] University of Waikato. 2017. Weka 3 - Data Mining with Open Source Machine Learning Software in Java. (2017). http://www.cs.waikato.ac.nz/ml/weka/
[12] Willian Zamora, Carlos T. Calafate, Juan-Carlos Cano, and Pietro Manzoni. 2016. A Survey on Smartphone-Based Crowdsensing Solutions. *Mobile Information Systems* 2016 (2016), 1–26. https://doi.org/10.1155/2016/9681842