Irina Diaconita, Andreas Reinhardt, Frank Englert, Delphine Christin, Ralf Steinmetz: *Do You Hear What I Hear? Using Acoustic Probing to Detect Smartphone Locations*. In: Proceedings of the 1st Symposium on Activity and Context Modeling and Recognition (ACOMORE), pp. 1-9, March 2014.

# Do You Hear What I Hear? Using Acoustic Probing to Detect Smartphone Locations

Irina Diaconita*, Andreas Reinhardt†, Frank Englert*, Delphine Christin‡ and Ralf Steinmetz*
*Multimedia Communications Lab, Technische Universität Darmstadt, Darmstadt, Germany
†School of Computer Science and Engineering, The University of New South Wales, Sydney, Australia
‡Secure Mobile Networking Lab, Technische Universität Darmstadt, Darmstadt, Germany

***Abstract*—Many context-aware smartphone applications depend on specific conditions for gathering data, e.g., specific phone locations or orientations. As a result, the significant overhead of keeping all this information in mind is imposed on their users. Besides averting the interest of potential application users, these requirements defeat one of the main purposes of these mobile data collection, namely simplifying life through mobile sensing applications. This is not a problem that solely affects the users, but the developers of the applications alike. As even the most diligent users often do not manage to follow the strict data collection guidelines at all times, errors in the collected data may ultimately lead to the provision of wrong services and thus to degraded application quality.**

**In this paper, we thus present a solution to determine the location of a phone in order to support context-aware applications. It offers the possibility to detect the position of the phone with an accuracy of 97%, as well as being able to correlate it with the type of the location of the user. Our system can be used to improve existing mobile sensing applications by facilitating various services that depend on the phone location, e.g., seamlessly adapting the ringtone volume or setting a phone's flight mode.**

## I. Introduction

Mobile phones have become ubiquitous and offer an increasing number of innovative applications, which may contribute to improve the user's quality of life. For example, the embedded sensors can be used to measure the noise pollution in urban areas [1] or monitor the user's sport activities [2]. The quality of the provided services, however, often depends on the location of the mobile devices. For example, mobile phones carried in bags or pockets will collect different sound levels than when a mobile device is being held in user's hands.

Clearly, information about the smartphone's current location is thus not only beneficial to weed out sensor samples collected when noise is present or the mobile phone's sensor is covered. They are also very valuable in order to improve the accuracy and/or quality of context-aware applications that rely on the collected sensor data.

Within the scope of this paper, we hence propose a novel method to identify the position of the user's mobile phone in different contexts. This information can then be leveraged in the proposed applications and included in the computation of the application outcomes. Our solution is based on short bursts of audio signals emitted and recorded by the mobile phones. While these audio signals are inaudible by the users, the differences in signal attenuation reveal the nature of the material surrounding the mobile phones. In comparison with existing solutions, our method solely relies on on-board sensors and enables an identification of multiple phone locations in various user contexts.

Furthermore, our approach can serve as an enhancement to existing mobile sensing approaches, which often depend on the quality of the data and therefore have certain constraints for the user, like always carrying the phone in a certain position. Our solution would allow such applications to assign weights and confidence levels to the detected user contexts based on the phone's position. Thus they could improve their performance and, at the same time, their user acceptance, by limiting the overhead for the user.

We have implemented our solution on both Samsung Galaxy Nexus and Galaxy S3 smartphones. In order to assess its classification accuracy, we have recorded more than 7,800 audio signals and analyzed different machine learning solutions to determine the current location of the mobile phones. Possible locations include: (1) in a backpack, (2) on a desk (display facing up or down), (3) in the user's hand, and (4) in the user's pocket. Moreover, we have tested different user contexts, including in an office setting, outdoors, and in public transportation.

The remainder of this paper is organized as follows: We first summarize existing work in Section II, before introducing our concept in Section III. We then detail our implementation in Section IV and present the results of our evaluation in Section V. We finally conclude this paper and discuss future work in Section VI.

## II. Related Work

A first category of solutions for user context detection is based on external sensors. This includes approaches using external microphones and accelerometers [3], multiple external accelerometers [4], external cameras [5], as well as integrated devices with multiple sensors [6] (accelerometer, barometer, thermometer, microphone, etc.). Other authors focus on using a single external sensor, often in the form of a microphone [7, 8, 9, 10]. While these approaches have the advantage of collecting less noisy data than on-board sensors and capturing more relevant features (like collecting data from multiple microphones placed in strategic locations), they also require external hardware. Thus, they incur extra

costs and, furthermore, external wearable sensors might cause user discomfort.

To overcome these shortcomings, approaches that used only the sensors integrated in smartphones were developed. Some applications focus on the information that can be obtained from one specific sensor, this being also the approach we take. For instance, GPS and WiFi traces are used to detect and predict user mobility [11], accelerometers to detect the user's physical activities [12], camera and microphone to characterize the user's environment [13, 14]. Reference [15] uses WiFi/Bluetooth traces to build user mobility patterns. For obtaining more complex information, the readings from multiple sensors are processed. For example, SurroundSense [16] attempts to detect also the type of location the user is in (library, restaurant, club, etc.) using WiFi, microphone, accelerometer, camera, and light sensor readings. CenceMe [17] records GPS, accelerometer, camera, and audio data to detect the user's activity and then shares it through the user's social networks, e.g. Facebook.

Similarly to our work, different solutions have been developed solely based on the built-in microphone. These solutions focus on the recognition of either human-based or environmental sounds. There are two general areas of interest, human-produced sounds and environment sounds. Approaches for human-emitted sounds include voice/speaker recognition [18] and emotion and stress detection [19]. A special part of this field is dedicated to medical purposes, like cough detection [20], physiological anomaly detection [21] or a heartbeat counter [22]. Environment sounds have multiple applications such as building maps and estimating noise pollution levels [14] and music/genre recognition [23]. Further aproaches include characterizing locations based on the ambient noise [16] and estimating the user's enerygy consumpion based on the sounds produced by various appliances [24]. Approaches like [25, 26, 9, 27, 28] classify a variety of sounds in different categories in order to detect the user's activity and characterize their environment, but do not address the localization of the phone itself.

The closest approach to the work presented in this paper is [29], which proposes a system to distinguish between various locations of a smartphone by recording the environment sounds. The solution is implemented only for two locations, inside and outside of the user's pocket. In comparison, we do not record environmental sound, but actively probe the phone's environment by playing back short pilot sequences. Due to the shortness of these sequences, the users are not disturbed. By doing so, we are not only able to determine if the phone is either within or out of the user's pocket as in [29], but can recognize multiple contexts.To the best of our knowledge, we are therefore the first to use active probing sequences to determine the environment conditions of mobile phones.

## III. Concept

As previous research has shown [28], sound samples alone can be used successfully to classify events from the user's environment, as long as they have quite distinct acoustic

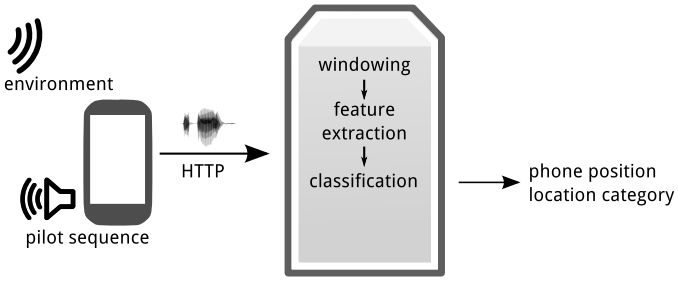



Fig. 1. System architecture

fingerprints. Furthermore, environment noises can harm the accuracy and the systems strongly depend on an appropriate positioning of the recording device. Such conditions include having the device directly exposed to the source, in a roughly similar position at all times (and, most importantly, similar to the one used for the initial training of the classifier).

The conditions required to obtain the same performances are however not compatible with a real-world deployment. For example, users may not constantly hold their mobile phones in their hands, but may carry them in backpacks or bags. Some users may also choose to wear their mobile phone on their belt or in a pocket. As, however, a majority of them are taking their phones out and walking with them in their hands, e.g., to check emails, changes in phone location are a common occurrence. Users might also simply leave their phone on a desk while at work and not take it along when going to a different place for a meeting or for lunch, which would render any context aware services useless and needlessly drain the phone battery.

To overcome this limitation, we probe the attenuation of a deterministic signal in the phone's current environment and use this information to determine the phone location. Our primary goal is thus to detect the position of the phone (pocket, backpack, desk, hand) regardless of the user's environment. However, we also analyze the data in order to correlate the position of the phone and the user's type of location (indoors, outdoors, public transportation).

Based on the outcome of our literature survey presented in the previous section, we propose to use a smartphone to play various pilot sequences and record them at the same time using its embedded microphone. We rely on the assumption that the sound is attenuated differently for the most common phone positions. While this might also be true for the environment noises (i.e., without emitting a known pilot signal), recordings of them alone are less clearly distinguishable and more prone to classification errors.

In our system, whose architecture is shown in Figure 1, the recordings are subsequently saved and transferred to a server, where they are processed. The processing pipeline include windowing, silence removal, feature extraction, and classification, which we explain in detail in the next section.

The only remaining user concern would be discomfort by annoying pilot sequences being played back continuously. As a result, the improvement in functionality (i.e., knowing the location of the phone) would be questionable if the perpetual

exposure to disturbing noises. This would be a legit problem if the pilot signals were long enough, but our classifier only requires samples of 10 ms duration and smartphones have the capacity of playing such short audio samples.

## IV. Implementation Details

### A. Sample Collection

We implemented an Android app which can play back wave audio files and collect sound recordings at the same time. The app collects recordings of a given duration at given time intervals. If the user is connected to a WiFi network, the app sends the files directly to the server. Otherwise it stores them locally and sends them to the server when the user connects to a WiFi network. The samples are then classified on the server.

When recoding the sound we had to take into account the technical capabilities of various phones, as only a few can support a 44.1 kHz sampling rate. For this reason, the app keeps attempting to record tracks, starting with 44.1 kHz and going down to 22.05, 11.05 and 8 kHz until a sampling rate supported by the phone is found. However, for the evaluation we only used phones that supported the 44.1 kHz rate. In addition, the volume of the generated pilot sequences is monitored and automatically adapted in order to avoid clipping, as well as having too faint and indistinguishable recordings.

### B. Pilot Sequences

We use a total of four pilot sequences. We started with Gaussian noise as a general way to probe our approach. The differences in spectrum for the phone positions are quite visible, as it can be seen in Figure 2.

We further selected a few sequences composed of prime numbers, such that harmonics ranging at multiples of the fundamental frequency will not impact our results). The audio data we intended to collect was to be sampled at 44.1 kHz, so due to the Nyquist limit, we selected only frequencies under 20 kHz for our probing sequences. We used two sequences of pseudo-logarithmically distributed primes, half of them distributed under 1 kHz and the other in the 1-20 kHz range. The first of the samples uses 40 primes, while the second uses 316, including all 158 primes under 1000. A further pilot sequence we used included 40 linearly distributed primes under 20 kHz.

The semilogarithmic sequences are visualized in Figure 3, together with the spectrum of samples collected in the main phone locations. One can notice that the spectral shapes are pronouncedly more distinct in this case than for the case using Gaussian noise. We should note that these samples were collected in silent environments; in noisy environments the spectrum becomes less distinguishable than for the Gaussian noise.

### C. Classification Process

To determine the location of the phone as well as the location of the user, we use a custom-tailored audio classification cascade. This cascade consists of four consecutive stages: windowing, Fourier transformation, feature extraction and classification. In the following paragraph, a brief description of these stages will be given.

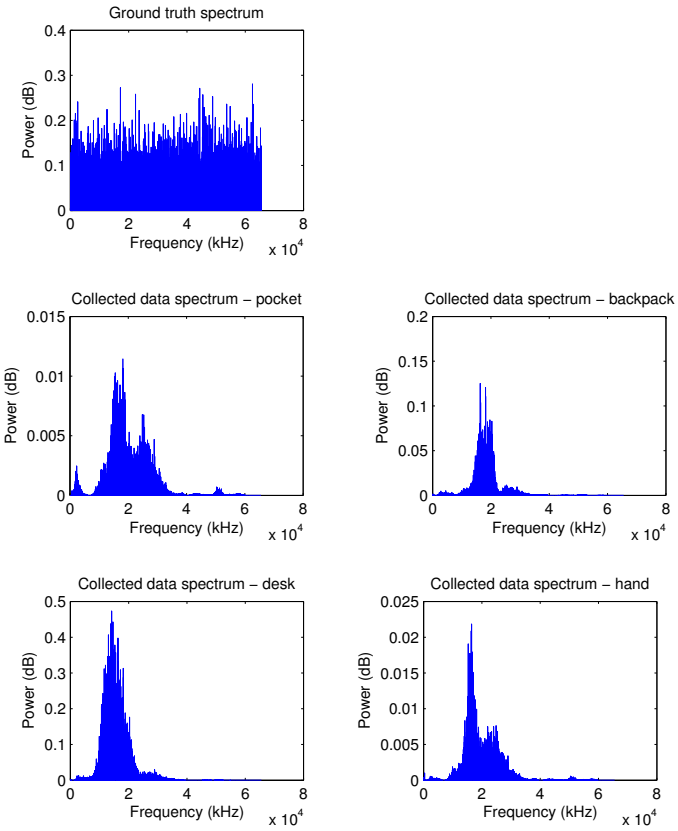



Fig. 2. The spectrum of the Gaussian noise sample, as well as of the samples collected with a phone located in the pocket, backpack, on the desk and in the user's hand

In the first stage, the sampled audio signal $s(t_n)$ is split into equally sized windows $w(t_n)$ of 4,096 samples per window. Given a sample rate 44,100 Hz, this causes an audio length of 10ms per window. Currently our implementation allows the usage of rectangular, Hamming or Hanning windows.

After windowing the data, each window is transformed to the frequency domain: $W(fn) = DFT(w(t_n))$. This representation directly shows the frequency components of the analyzed signal. Thus, selecting an appropriate window size is crucial for the performance of the overall system because the length of the window directly influences the frequency resolution of the Fourier transformation. Longer windows with more samples take longer to record but they also cause a higher frequency resolution. On the other hand, smaller window sizes results in a lower frequency resolution but cause a lower computation complexity. In our setup with a sample rate of 44,100 Hz and a window size of 4,096 samples this results in 4,096 spectral components ranging from 0Hz to 22,050 Hz with a frequency resolution of 10.77 Hz.

To further reduce the amout of data, the next stage extracts a feature vector for transformed window $I = W(f_n)$. Currently our implementation supports the extraction of Mel Frequency Cepstral Coefficients ($MFCC$), Delta Mel Frequency Cepstral

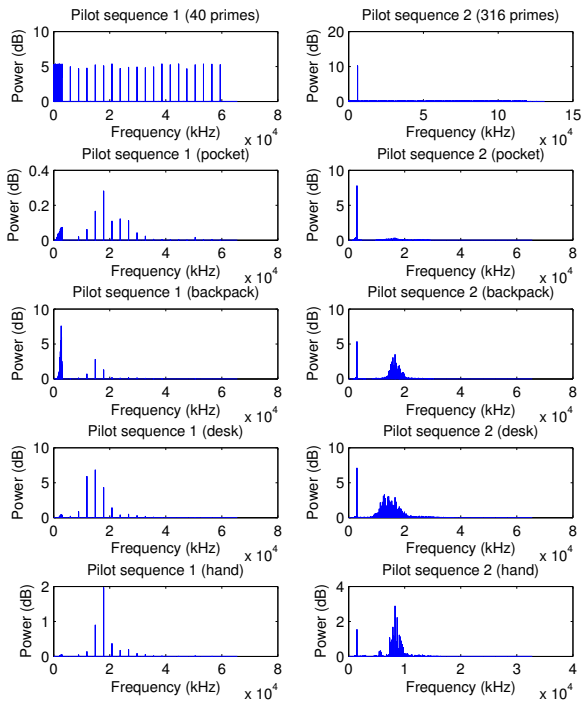



Fig. 3. The spectrum of the two sequences of pseudo-logarithmically distributed primes, as well as of the samples collected with a phone located in the pocket, backpack, on the desk, and in the user's hand

Coefficients ($DMFCC$) and Band Energy ($BE$) features, which are calculated by folding the window $W(f_n)$ with N triangular shaped filters with logarithmically distributed center frequencies. According to the findings of other researchers [30], we selected a value of N=13 to compute the first 13 cepstral coefficients and thus to create a feature vector with 13 elements. An in-depth description of the MFCC features including their relation to the human hearing and their calculation was contributed by [31]. The DMFCC features are the derivation of the MFCC features over time and clearly show transitions of different tones over time and the Band Energy features indicates the Band Energy for $M$ different frequency bands.

In the last stage, the previously extracted features are classified to determine the location of the user as well as the location of the phone. Although our framework would allow the usage of arbitrary classification alorithms, we decided to focus our research to tree-based and distance-based algorithms. We expect those algorithms to work best because all $N$ features which are forwarded from the feature extraction phase are numerical values which express the distribution (spectral shape) of the signal's frequency components. This distribution is directly influenced by the frequency selectivity of the transmission channel. As our work relies on the fact that this frequency selectivity is nearly constant and unique for a given environment, all feature vectors recorded in this particular environment will form clusters in the N-dimensional feature space.

Given the window size of 10 ms, it would be possible to collect up to $K$ classification results, then to fell a majority vote without increasing the recording time over $K * 10ms$. We decided to not follow this route because of two reasons: first, a variation of the schema is the core working principle of the ensemble-based machine learning algorithms we used and secondly such an algorithm would accumulate state in the classification cascade which complicates the evaluation. This is the case because the direct relation between input signal and classified output would be lost.

## V. Evaluation

### A. Evaluation Setup

For the evaluation, we recorded samples using a Samsung Galaxy Nexus and a Samsung Galaxy S3. The phones were chosen for to their audio recording quality and their support of a sampling rate of 44.1 kHz. For the Galaxy S3 we noticed the volume of the sequences we played had to be pronouncedly higher than for the Galaxy Nexus in order to achieve comparable results, presumably due to its noise reduction feature.

The general phone locations we considered were: desk, hand, pocket and backpack, as they cover the most common situations of a phone in use. One of our hypotheses was that the differences in the propagation of the probing sequences induced by the different environments would lead to very good accuracies for the classification.

As visible in Table I, we collected the samples both in silent (e.g., office) and noisy environments (e.g., outdoors, tram, bus etc.), the majority of these being in noisy environments. This structure of the data poses a problem to the classifiers, as each individual class is actually a combination of two distinct classes: the phone position and the user location.

We collected a total of 7862 samples while generating Gaussian noise, and a similar number without using any probing sequence. For each of the other pilot sequences, we collected 1640 recordings on average. The achievable classification accuracy of our prediction model has been evaluated by means of a 10-fold cross validation.

While recordings of 10 ms duration are sufficient for our system to work properly, for training and evaluation purposes we used longer recordings (1 to 10 seconds), and thus can ponder on the disturbance of the sequences themselves. The volume of the samples we played was less than half of the maximum supported by the phone in order to avoid clipping, so the noise was quite easy to ignore or go unnoticed in noisy environments, like outdoors or in a tram, and even in silent environments if the phone was in a backpack.

### B. Gaussian Noise

Let us first analyze the classification accuracy of the samples collected while generating Gaussian probing sequences. We used a variety of situations, noisy and silent, indoors and

TABLE I
PHONE AND USER LOCATIONS USED FOR THE CLASSIFICATION

| Code | Phone location | User location | State | # Windows |
|---|---|---|---|---|
| bbu | Backpack | Bus | Motion | 3975 |
| bof | Backpack | Office | Stationary | 11867 |
| bos | Backpack | Outdoors | Stationary | 4015 |
| bow | Backpack | Outdoors | Motion | 3699 |
| btm | Backpack | Tram | Motion | 3978 |
| ddf | Desk (down) | Office | Stationary | 11952 |
| ddt | Desk (down) | Outdoors | Stationary | 4028 |
| duf | Desk (up) | Office | Stationary | 11818 |
| dut | Desk (up) | Outdoors | Stationary | 3975 |
| hbu | Hand | Bus | Motion | 3780 |
| hof | Hand | Office | Stationary | 11685 |
| hos | Hand | Outdoors | Stationary | 4011 |
| how | Hand | Outdoors | Motion | 4024 |
| htm | Hand | Tram | Motion | 4016 |
| pbu | Pocket | Bus | Motion | 3926 |
| pfs | Pocket | Office | Stationary | 11960 |
| pfw | Pocket | Office | Motion | 11931 |
| pus | Pocket | Outdoors | Stationary | 3991 |
| puw | Pocket | Outdoors | Motion | 6336 |
| ptn | Pocket | Train | Motion | 12024 |
| ptm | Pocket | Tram | Motion | 11973 |
| | | | Total no. of windows | 148964 |

outdoors, stationary and in motion, as shown in Table I. When placing the phone on a table, we distinguish between the case when it is facing the table or the ceiling, thus having the microphone covered or fully exposed.

Next, we compare the accuracies of all combinations of feature extraction algorithms and classifiers for determining the phone and user type of location at the same time.

The features that we used were MFCC, DMFCC, and BE. For the classification model we used Gaussian Naive Bayes (GNN), Decision Trees (DT), K-Nearest Neighbors (KNN), Random Forest (RF), and Gaussian Mixture Model (GMM) classifiers.

Results can be seen in Figure 4, where one can notice that the best results, as far as features are concerned, are given by MFCC and Delta MFCC. This confirms the expectations that the two algorithms which take the coefficients of the spectrum into account result in best results (cf. Figure 2). MFCC has slightly better results than Delta MFCC, as the latter is calculated as the difference between MFCC feature values, and thus slightly uniforming the features.

K-Nearest Neighbors and Random Forest are the most accurate classifiers with a 97% and 96% accuracy respectively, as the extracted features form distinct clusters on two layers: phone position and user location. This also leads to Decision Trees having a lower accuracy than Random Forests, as the phone position, the user location, and their correlation are distinct enough to require at least separate trees.

The way data is clustered affects even more severely the Gaussian Mixture Model results, making the extracted features play an even more important role. For instance, while with

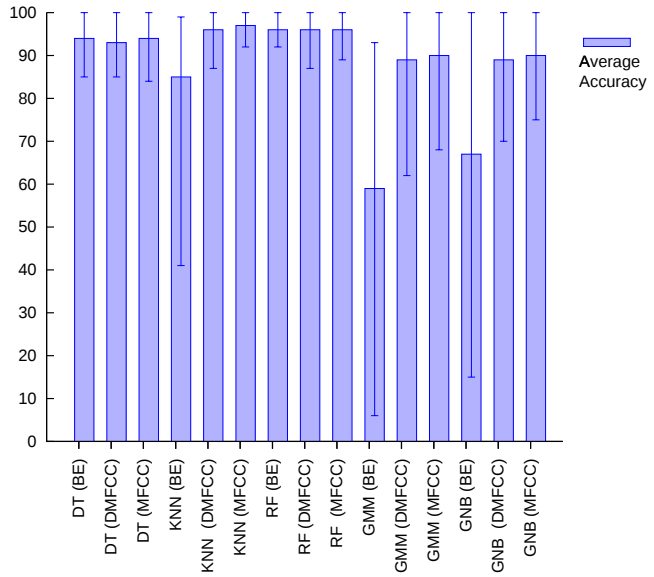



Fig. 4. Comparison of the classifier performances and feature types

MFCC and Delta MFCC the average accuracy is 90% and 89% respectively, for band energy it drops to 59%. It should also be noted that while for the classifiers with higher accuracy the standard deviation is quite low, it increases significantly for the lower accuracy classifiers.

### C. Gaussian Noise vs. No Pilot Sequence

Next, we will compare the results for the samples recorded using Gaussian noise and recordings of the environment sounds alone. As we have seen previously, MFCC together with K-Nearest Neighbors offer the best results for both types of recordings, therefore we are going to use them to compare their performance.

Figure 5 represents the confusion matrices for the aforementioned situations for the two approaches, considering all the situations from Table I. One can notice a clear improvement to the accuracy of the results due to the Gaussian noise, the recordings using Gaussian noise having a 97% accuracy, whereas the recordings of the environment sounds alone only reach a 71% accuracy.

Although these are quite good results for the recordings without any pilot sequences, this can mainly be attributed to the fact that environment sounds are still perceived in a different way: ideally when the phone is on a desk, muffled and attenuated when the phone is in the pocket. Of course, as shown by the figure, it is more difficult to distinguish between sounds muffled by a tight pocket or a big compartment of a backpack, or, even more so, between a phone lying on a desk facing the ceiling or the desk. Even if slightly more accurate, the recordings alone face another challenge when the phone has a similar location (e.g., desk) and have to distinguish between two different environments, or even a silent and a noisy one (e.g., indoors vs. outdoors).

When using the Gaussian noise the results are very good for classifying both the phone and the user location. While

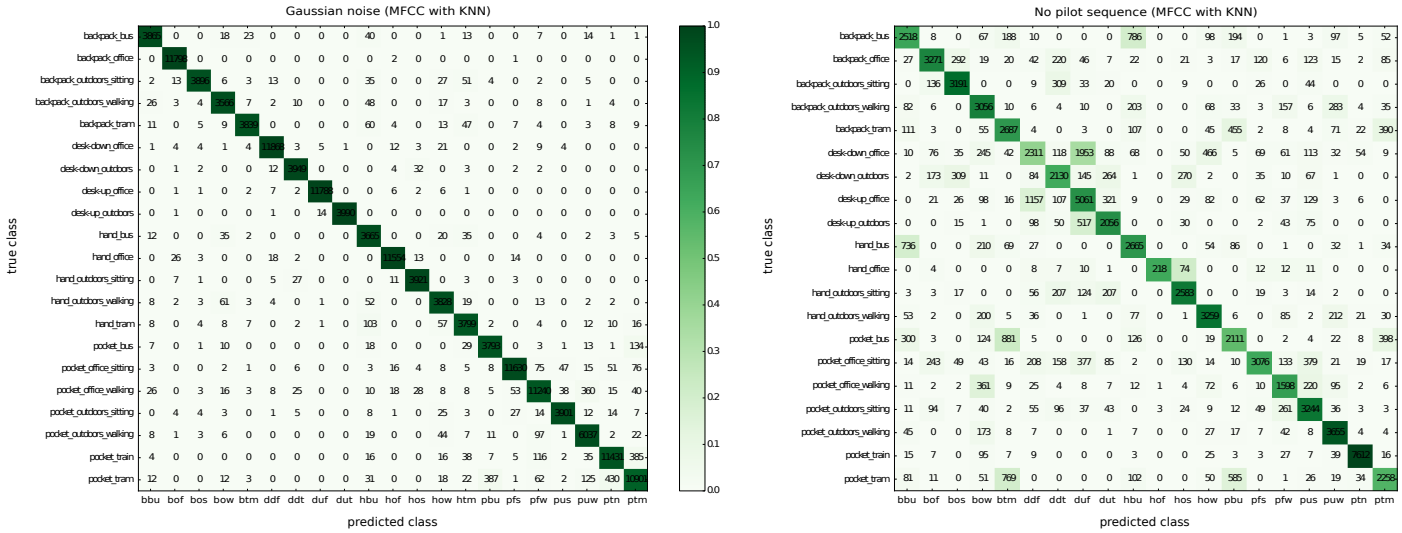



Fig. 5. Comparison of the classification results for recordings using Gaussian noise as pilot sequence vs. recordings of the environment sounds alone

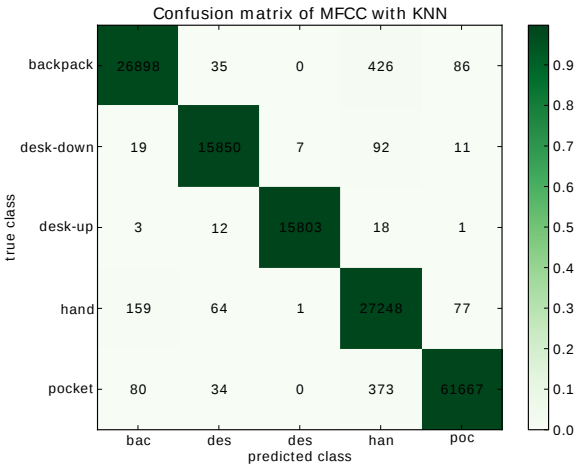
for the former, we attribute the performance to the pronounced propagation pattern of the Gaussian noise (much more obvious and pronounced than that of the environment sounds alone), for the latter the Gaussian noise helps cancel out some of the irrelevant environment noise. Here, the environment specific features are much more obvious as a difference between the frequencies of the recordings and those of the initial noise.

Next we will compare the overall accuracies for the general phone locations: hand, pocket, backpack, desk. We use the same datasets and just ignore the location of the user, thus lumping into the same class all recordings taken with the phone in the same position. Figures 6(a) and 6(b) present the confusion matrices for the two classifications, done using K-Nearest Neighbors and MFCC, like for the previous evaluation step.

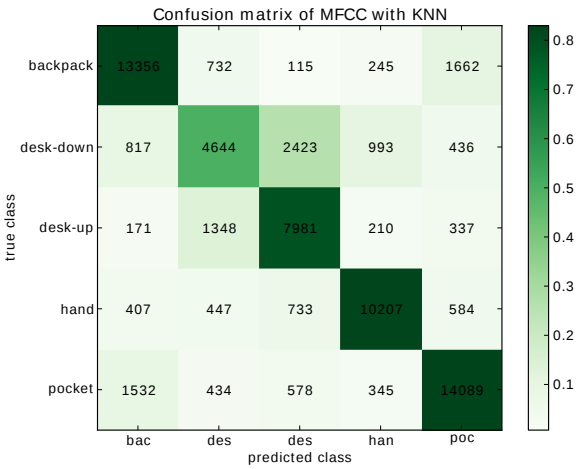
The already good accuracies see an improvement, from 97 to 99% using Gaussian noise, and from 71 to 77% for the environment sounds alone. The results for the samples using Gaussian noise are quite expected, given that we only care about the phone location, and thus the features of the propagated signal alone, while the type of features we used (MFCC) extracts a sequence of values of the cepstrum ("spectrum of the spectrum").

The 99% accuracy of the Gaussian noise recordings clearly argues for their suitability for such a task, given the variety of environments they were evaluated on. Even within the same environment, different samples had different noise levels and patterns. For instance, in a tram some recordings will have people talking, some will have just the tram noises, some will have the tram stopped in a station, and others will have the next station announced on the speakers. Therefore, it is important to note the robustness of the method under such conditions.

The accuracies for the recordings alone improve by about 10%, as the user's environment does not need to be detected. The classifiers relying on whether the particular sound is muffled and attenuated or not and extrapolating this pattern




(a) Gaussian noise




(b) No pilot sequence

Fig. 6. Classification results for the phone positions

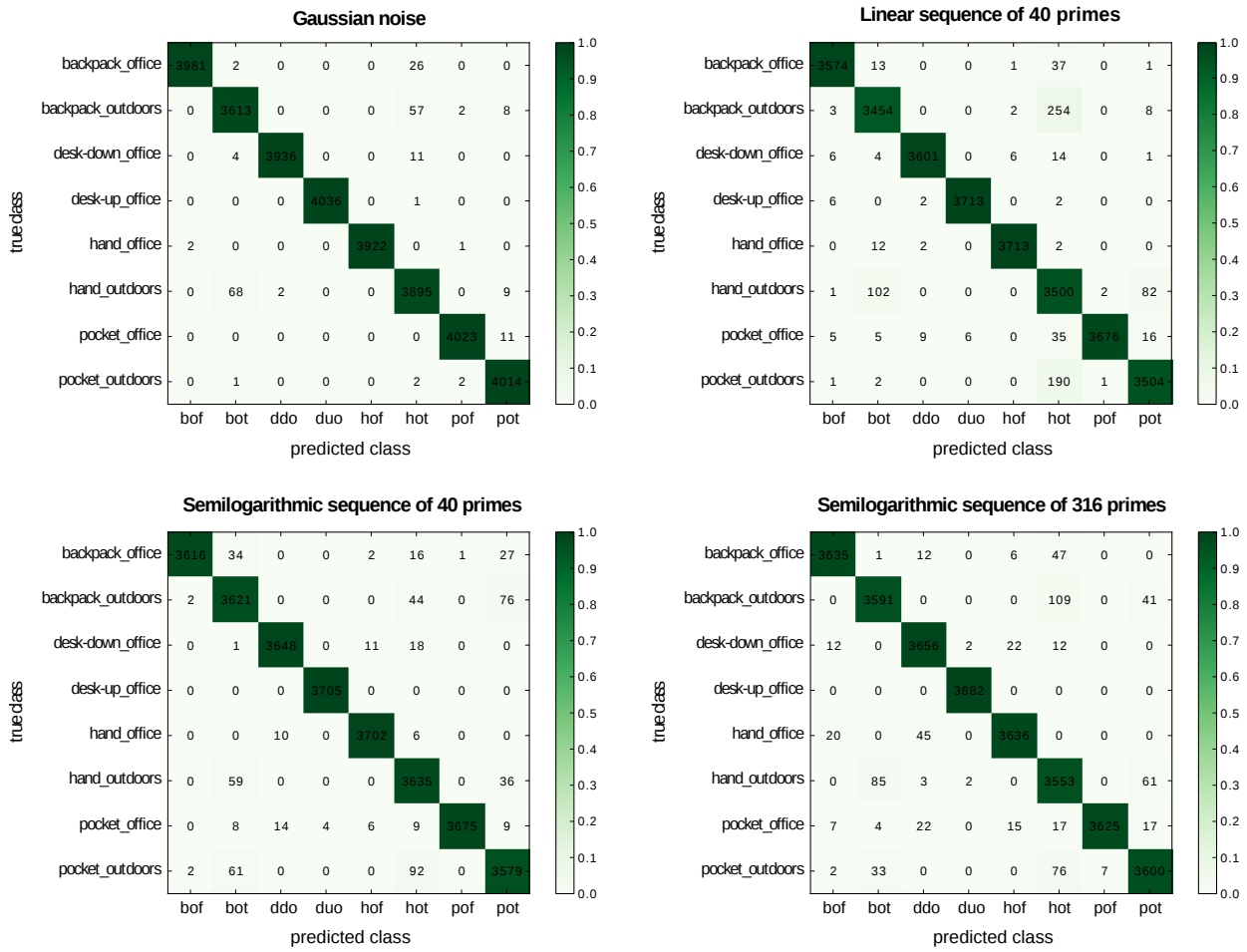



Fig. 7. Classification results for the various pilot sequences using Random Forest and Delta MFCC

alone is easier to achieve. However, distinguishing whether the phone is lying on its front or its back on a desk is still problematic, and accountable for the most misclassified samples.

The situation when the phone is lying on the desk facing the ceiling is also quite often confused with the user's hand, as the recording conditions are quite similar and the environment sounds are not changed or affected by any obstacles. This is also the reason for the high number of confusions between the pocket and the backpack: the sound is muffled to a certain degree, but it is quite difficult to distinguish between these degrees.

The bottom line for the recordings of environment sounds would be that they are pretty good at telling apart an open environment (hand, desk) from a closed one (pocket, bag). Also, as previously shown, the environment classification and its correlation with the "in" or "out" position of the phone is quite accurate. On the other hand, recordings using Gaussian noise have outstanding results both for determining the phone position and for correlating it with the user's location.

### *D. Comparison Between All Pilot Sequences*

Next, we compare the classification results using the four different pilot sequences, considering the main phone positions in silent and noisy environments (outdoors). For the classification we used Random Forests with Delta MFCC, as it was the second best classifier, its accuracy being very close to the top one (only 1% difference). Figure 7 presents the confusion matrices for all the cases.

As one can notice, the overall differences are quite small, with a 99% accuracy for Gaussian noise, 98% for both semilogarithmic sequences and 97% for the linear one. However, there are a few slight distinctions to be made. The Gaussian signal has the best overall performance, as it is less affected by noise than the other signals. One can notice that all four signals have a very similar number of correctly classified samples for the silent situations, coming very close to 100%.

Furthermore, in some of these silent situations, the sequences of primes have a better performance, as expected from the differences in the spectrum. For instance, for the case of a phone lying on a desk facing the ceiling in the office, the semilogarithmic sequence of 40 primes has the best accuracy, classifying it correctly in all cases for the given data set. The

instances that are misclassified by the sequences of primes due to the noise tend to be assigned to the correct user location, but to the wrong phone position.

### *E. Sequences of Primes*

In what follows, we will look closer at the capabilities of the sequences of primes previously presented to determine the overall phone positions. We use again the same datasets, but label them only based on the phone's position. The accuracies are 98% for the semilogarithmic sequence with 40 primes, and 97% for the other two signals. The confusion matrices are represented in Figures 8, 9 and 10.

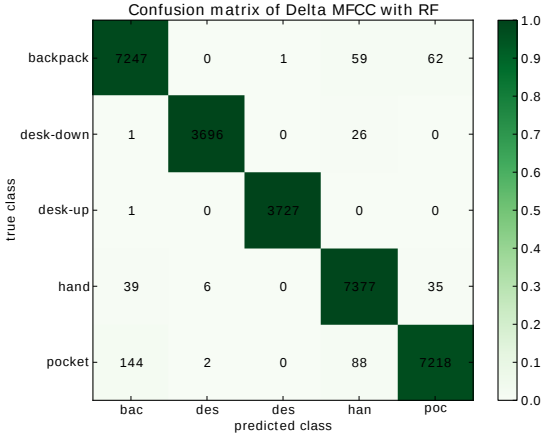



Fig. 8. Classification results for a pilot sequence composed of 40 semi-logarithmically distributed primes

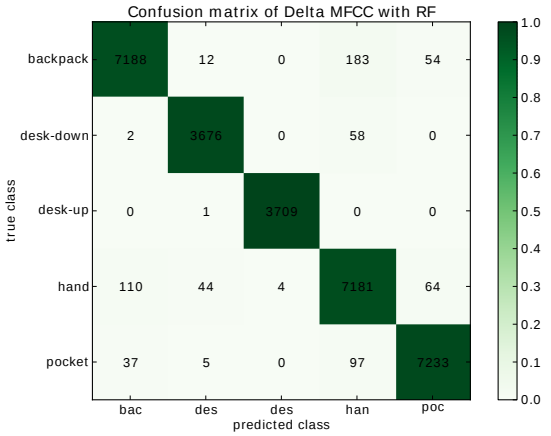



Fig. 9. Classification results for a pilot sequence composed of 316 semi-logarithmically distributed primes

For all the signals, it can be noted that, when a phone is lying on a desk, almost all instances are correctly classified. The semilogarithmic sequence of 40 primes has markedly the best results, with a slight confusion between the pocket and the backpack, generated most likely by the noisy conditions of some of the recordings. As has been previously shown, the results of all these three sequences have a lower quality for samples gathered in noisy environments. Therefore, one can

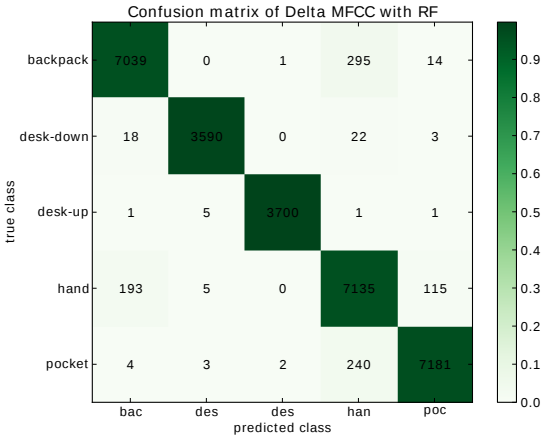



Fig. 10. Classification results for a pilot sequence composed of 40 linearly distributed primes

notice the confusions between the backpack and the pocket, as well as, to lesser extent, between the backpack and the user's hand for the other two signals as well

### *F. Evaluation Conclusion*

To sum up, environment noise recordings alone have decent results, but fail at determining more than whether the phone is in an open or closed environment. Gaussian noise has the best overall results for detecting the phone position, either on it own, or correlated with the user's type of environment. The sequences of primes have a higher accuracy than Gaussian noise for a good part of the data collected in silent environments, but are easily affected by noise and therefore do not measure up to the Gaussian signal in noisy environments.

Given the clustered structure of the classification task, with classes made up of two different elements, phone position and user type of location, K-Nearest Neighbors and Random Forest were the most successful classifiers. We relied on the differences in spectrum of the different classes of recordings, and MFCC and Delta MFCC were the features that led to the best results.

## VI. Conclusions and Future Work

Localization of mobile devices represents an important foundation for many user-centric services. Based on a smartphone's actual location within its user's environment, different services can be offered. We have thus presented a solution to identify the position of a smartphone by means of emitting short bursts of audio signals. By recording them simultaneously, the frequency-dependent signal attenuation of the material surrounding the mobile device can be determined.

We have analyzed different machine learning solutions with regard to their capability of modeling the spectral response to our pilot sequences and determined that a K-Nearest Neighbor classifier works best for the given data, achieving 97% of accuracy for 21 different tested positions of the phone. A supplementary analysis of further pilot sequences has shown that some pilot sequences (e.g., using 40 semi-logarithmically

distributed prime frequencies) are better suited to determine certain smartphone positions, like those confined to a office setting, but overall a Gaussian signal has been shown to perform best.

In the future, we plan to continue investigating other pilot sequences that specifically range in the frequency spectrum in which most discrepant attenuation levels could be observed. We also intend to research on possible further mechanisms to determine characteristic sequences in the time domain, e.g., by applying symbolic approximation mechanisms, like SAX [32]. The system could furthermore be improved by including readings from additional sensors, like light sensors. These could be used as a further filter to increase certainty of the already detected phone locations or to distinguish between borderline cases. Another aim would be to implement the classifier on the phone as an answer to the privacy concerns raised by centralized storage of the audio recordings.

## ACKNOWLEDGMENT

This work is supported by funds from the German Federal Ministry of Education and Research under the mark 01PF10005B and from the European Social Fund of the European Union (ESF). The responsibility for the contents of this publication lies with the authors.

## REFERENCES

[1] N. Maisonneuve, M. Stevens, M. E. Niessen, and L. Steels, "NoiseTube: Measuring and Mapping Noise Pollution with Mobile Phones," in *Proceedings of the 4th International Symposium on Information Technologies in Environmental Engineering*, 2009.

[2] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "The BikeNet Mobile Sensing System for Cyclist Experience Mapping," in *Proceedings of the 5th ACM International Conference on Embedded Networked Sensor Systems*, 2007.

[3] J. Ward, P. Lukowicz, G. Troster, and T. Starner, "Activity Recognition of Assembly Tasks Using Body-Worn Microphones and Accelerometers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[4] L. Bao and S. S. Intille, "Activity Recognition from User-Annotated Acceleration Data," in *Pervasive Computing*, 2004.

[5] B. U. Töreyin, Y. Dedeoğlu, and A. E. Çetin, "HMM Based Falling Person Detection Using Both Audio and Video," in *Computer Vision in Human-Computer Interaction*, 2005.

[6] T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. Legrand, A. Rahimi, A. Rea, G. Bordello, B. Hemingway, P. Klasnja, K. Koscher, J. Landay, J. Lester, D. Wyatt, and D. Haehnel, "The Mobile Sensing Platform: An Embedded Activity Recognition System," *Pervasive Computing*, 2008.

[7] J. Chen, A. H. Kam, J. Zhang, N. Liu, and L. Shue, "Bathroom Activity Monitoring Based on Sound," in *Pervasive Computing*, 2005.

[8] C. Couvreur, "Environmental Sound Recognition: A Statistical Approach," Ph.D. dissertation, Facult'e Polytechnique de Mons, 1997.

[9] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa, "Computational Auditory Scene Recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002.

[10] M. Stager, P. Lukowicz, and G. Troster, "Implementation and Evaluation of a Low-power Sound-based User Activity Recognition System," in *Eighth International Symposium on Wearable Computers*, 2004.

[11] Y. Kim, H. Shin, and H. Cha, "Smartphone-based Wi-Fi Pedestrian-tracking System Tolerating the RSS Variance Problem," in *Proceedings of the International Conference on Pervasive Computing and Communications*, 2012.

[12] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity Recognition Using Cell Phone Accelerometers," *ACM SIGKDD Explorations Newsletter*, 2011.

[13] R. Elias and A. Elnahas, "An Accurate Indoor Localization Technique Using Image Matching," in *Proceedings of the International Conference on Intelligent Environments*, 2007.

[14] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: an End-to-end Participatory Urban Noise Mapping System," in *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010.

[15] L. Vu, K. Nahrstedt, S. Retika, and I. Gupta, "Joint Bluetooth/Wifi Scanning Framework for Characterizing and Leveraging People Movement in University Campus," in *Proceedings of the 13th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, 2010.

[16] M. Azizyan, I. Constandache, and R. Roy Choudhury, "SurroundSense: Mobile Phone Localization via Ambience Fingerprinting," in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, 2009.

[17] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing Meets Mobile Social Networks: the Design, Implementation and Evaluation of the CenceMe Application," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, 2008.

[18] H. Lu, A. J. Bernheim Brush, B. Priyantha, A. K. Karlson, and J. Liu, "SpeakerSense: Energy Efficient Unobtrusive Speaker Identification on Mobile Phones," in *Pervasive Computing*, 2011.

[19] H. Lu, D. Frauendorfer, M. Rabbi, M. S. Mast, G. T. Chittaranjan, A. T. Campbell, D. Gatica-Perez, and T. Choudhury, "StressSense: Detecting Stress in Uconstrained Acoustic Environments Using Smartphones," in *Proceedings of the ACM Conference on Ubiquitous Computing*, 2012.

[20] E. C. Larson, T. Lee, S. Liu, M. Rosenfeld, and S. N. Patel, "Accurate and Privacy Preserving Cough Sensing Using a Low-cost Microphone," in *Proceedings of the 13th International Conference on Ubiquitous Computing*, 2011.

[21] D. Hong, S. Nirjon, J. A. Stankovic, D. J. Stone, and G. Shen, "Poster Abstract: a Mobile-Cloud Service for Physiological Anomaly Detection on Smartphones," in *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*, 2013.

[22] S. Nirjon, R. F. Dickerson, Q. Li, P. Asare, J. A. Stankovic, D. Hong, B. Zhang, X. Jiang, G. Shen, and F. Zhao, "MusicalHeart: a *Hearty* Way of Listening to Music," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, 2012.

[23] A. Wang, "An Industrial Strength Audio Search Algorithm," in *Proceedings of the 4th Symposium Conference on Music Information Retrieval*, 2003.

[24] F. Englert, I. Diaconita, A. Reinhardt, A. Alhamoud, R. Meister, L. Backert, and R. Steinmetz, "Reduce the Number of Sensors: Sensing Acoustic Emissions to Estimate Appliance Energy Usage," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, 2013.

[25] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic Event Detection in Real Life Recordings," in *18th European Signal Processing Conference*, 2010.

[26] M. Rossi, S. Feese, O. Amft, N. Braune, S. Martis, and G. Troster, "AmbientSense: A Real-Time Ambient Sound Recognition System for Smartphones," in *Proceedings of the International Conference on Pervasive Computing and Communications*, 2013.

[27] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, "SoundSense: Scalable Sound Sensing for People-centric Applications on Mobile Phones," in *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, 2009.

[28] S. Nirjon, R. F. Dickerson, P. Asare, Q. Li, D. Hong, J. A. Stankovic, P. Hu, G. Shen, and X. Jiang, "Auditeur: A Mobile-Cloud Service Platform for Acoustic Event Detection on Smartphones," in *Proceedings of the The 11th International Conference on Mobile Systems, Applications, and Services*, 2013.

[29] E. Miluzzo, M. Papandrea, N. D. Lane, H. Lu, and A. T. Campbell, "Pocket, Bag, Hand, etc.-Automatically Detecting Phone Context Through Discovery," in *Proceedings of the ACM International Workshop on Sensing Applications on Mobile Phones*, 2010.

[30] G. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals," *IEEE Transactions on Speech and Audio Processing*, 2002.

[31] Logan, Beth and others, "Mel Frequency Cepstral Coefficients for Music Modeling," in *Proceedings of the International Symposium on Music Information Retrieval*, 2000.

[32] E. Keogh, J. Lin, and A. Fu, "HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence," in *Fifth IEEE International Conference on Data Mining*, 2005.