



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

VehicleSense: Transportation Mode Detection Using Sound Data with an Accelerometer-based Trigger System

Sungyong Lee

Department of Computer Science and Engineering

Graduate School of UNIST

VehicleSense: Transportation Mode Detection
Using Sound Data with an Accelerometer-based
Trigger System

A dissertation
submitted to the Graduate School of UNIST
in partial fulfillment of the
requirements for the degree of
Master of Science

Sungyong Lee

14. 06. 2016

Approved by



Advisor

Kyunghan Lee

VehicleSense: Transportation Mode Detection
Using Sound Data with an Accelerometer-based
Trigger System

Sungyong Lee

This certifies that the dissertation of Sungyong Lee is approved.

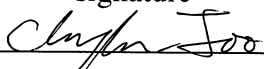
14. 06. 2016

signature



Advisor: Kyunghan Lee

signature



Prof. Changhee Joo

signature



Prof. Jaesik Choi

Abstract

A new transportation mode recognition system for smartphones, VehicleSense that is widely applicable to mobile context-aware services is proposed. VehicleSense aims at achieving three performance objectives: high accuracy, low latency, and low power consumption at once by exploiting sound characteristics captured while being on candidate transportations in a unique way. To attain the high energy efficiency, VehicleSense adopts hierarchical accelerometer-based triggers that minimize the activation of the built-in microphone of smartphones. Further, to attain the high accuracy and the low latency, VehicleSense manipulates the sampled sound with non-linear filters that are shown to lead to substantial performance improvement. Our 186-hour log of sound and accelerometer data collected by seven different Android smartphone models confirms that VehicleSense shows 98.2% of recognition accuracy with only 0.6 seconds of latency, while consuming only about 26.1 mW on average for all day monitoring.

Contents

1	Introduction	1
2	Related Work	4
2.1	Accelerometer only	4
2.2	Accelerometer and GPS	4
2.3	Accelerometer and Barometer	5
2.4	Accelerometer and Microphone	5
3	Motivation	6
3.1	Recognition accuracy	6
3.2	Recognition latency	6
3.3	Energy efficiency	7
4	Characteristics of Transportation Sound	8
4.1	Frequency-Domain Characteristics	8
4.2	Robustness of Sound Sensing	8
5	System Overview	13
5.1	Low Power Architecture	13
5.2	Accuracy Improvement	14
6	Triggers and Filters	17
6.1	Motion Trigger	17
6.2	Transportation Trigger	17
6.3	Sound Classifier	18
7	Evaluation	21
7.1	Data Collection	21
7.2	The Performance of Triggers	21
7.3	Performance of Sound Classifier	23
7.4	Performance Comparison with Existing Systems	25
8	Concluding Remarks	34

List of Figures

1	A sample output from an ideal transportation mode recognition system is compared with that from available systems that mostly have performance issues in recognition accuracy and latency.	2
2	Spectrogram of three transportations: bus, taxi, and subway. The red color indicates high spectrum intensity.	10
3	The unique sound patterns in the perspective of power spectral density are persistently observed over time for (a) bus, (b) taxi, and (c) subway.	11
4	The comparison of power spectral density over frequencies of an announcement made at a bus and the sample sound captured in a bus, a taxi, and a subway.	12
5	The block diagram of VehicleSense that consists of two hierarchical accelerometer-based triggers and the sound classifier.	16
6	Illustration of the seven different non-linear filters and their transformed scales ($M(f)$) over the standard herz scale (f).	19
7	CDFs of the Mag_a^{MA} values from the accelerometers under stationary (either being inside or outside a vehicle) and walking motions. We also depicted two thresholds, T_w and T_s	22
8	$p(W \rightarrow S)$, $p(V S)$, and $p(W_{cont})$ which are evaluated individually from the 110 user traces measured in the wild.	29
9	Precision and recall by smartphone devices.	33
10	$\mathbb{E}[\mathcal{P}]$ of VehicleSense and Hemminki's which are evaluated individually from the 110 user traces measured in the wild.	33

List of Tables

1	Experiment time(minute) of each transportation by smartphone model	27
2	Precision, recall, and score of our transportation trigger.	28
3	The power consumption of MT, TT, and SC modules implemented on Samsung Galaxy S6 running Android 5.1.1 Lollipop. Their activated durations for each call are also presented in seconds.	28
4	Precision, recall and F_1 score by the seven non-linear filters.	30
5	Precision, recall and F_1 score by the sampling durations. One result from the privacy protection mode (<i>PP</i>) is also included in the bottom.	31
6	The system-wide precision, recall and F_1 score of VehicleSense and Hemminki's system.	32

1 Introduction

We live in the world of smartphones where people interact with their smartphones all day. A recent report [20] reveals that Americans spend 3 hours on average with their smartphones per day and this is on a growing trend. As people spend more time on their smartphones, they expect higher intelligence from the smartphones. In order to meet this expectation, researchers have come up with the concept of recognition systems that bring intelligence to smartphones, with which the surrounding contexts are comprehended to provide timely services to the smartphone users. There already exist diverse recognition systems such as activity recognition [12], exercises recognition [9], transportation mode recognition [7], and touch-based identity recognition systems [31].

Among them, we give our special attention to the transportation mode recognition system that aims to figure out when a user gets on and off a vehicle and further figure out what type of the vehicle it is. Although the activity recognition is more general in terms of human life, we focus on recognizing the transportation mode due to its immediate applicability to a wide range of networking services. For instance, if a person wishes to let her smartphone save battery while streaming a video on YouTube, she should let the smartphone know the future wireless channel condition. This is for the smartphone to determine the right amount of video to prefetch at each moment since the energy consumption for prefetching the same amount of video is highly dependent on the channel condition. According to [11], the energy consumption in bad channel quality can be about two times higher than that in good channel quality when downloading the same amount of data through 4G LTE networks. This gap can be surely aggravated when the channel becomes almost unusable. The transportation recognition system can tell how the channel will vary in the future and can even tell when to avoid using the cellular network once it identifies the smartphone user is on a vehicle, especially either a bus or a subway whose route is predetermined. A similar advantage can be obtained when people on the same subway carriage wish to form a group and aggregate their cellular network connections to build the reliable tethering channel [1, 5] that may substantially reduce network outages. The transportation mode recognition system that knows when a person usually gets off can tell how to find the right person to invite to form a group. A wireless carrier can also benefit from the transportation mode recognition system if they wish to differentiate the video data pricing for the users either on a subway or on a bus, given that most of the mobile videos are shown to be consumed while people commute by public transportation [3].

Through a series of studies, there exist a number of transportation mode recognition systems such as [17, 6, 8]. However, those are not viable to run the aforementioned services because their recognition performance is not fully reliable in the wild. For instance, suppose there is a usual commute scenario where a person first takes a subway and changes to a taxi, and then walks to the office as in Figure 1.

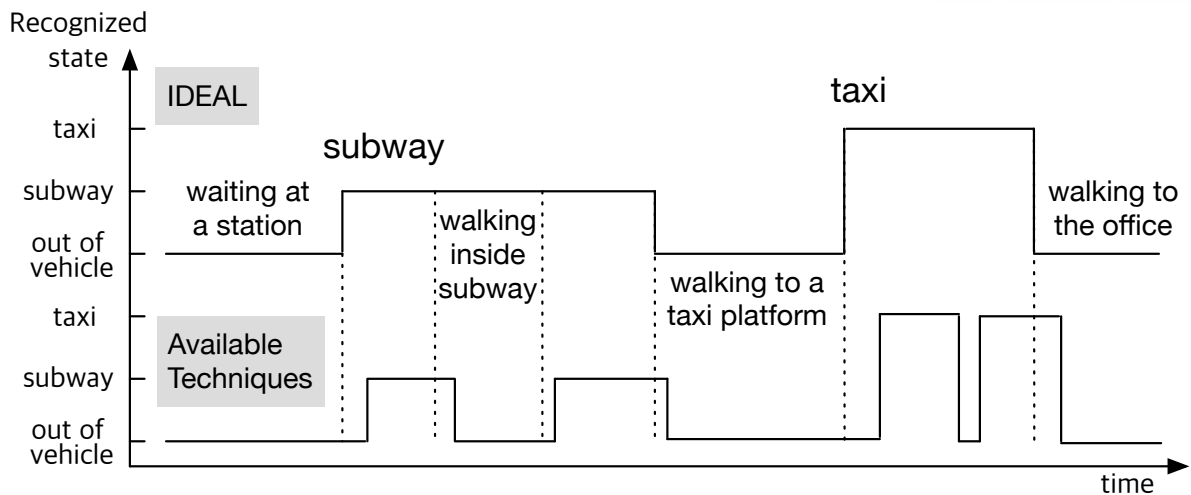


Figure 1: A sample output from an ideal transportation mode recognition system is compared with that from available systems that mostly have performance issues in recognition accuracy and latency.

The figure illustrates the outputs from both an ideal recognition system and available systems for the given scenario, which clarify three major performance issues of available techniques. The first issue is delayed recognition. Whenever the person makes a ride or ends a ride, most available systems experience non-negligible amount of recognition delay that easily spans to a few minutes. Such a long delay can cause critical misbehavior of services that rely on the transportation mode recognition system, such as unnecessary repeated scans to reform the tethering group although the group is no longer available for the person who got off a vehicle. The second issue is inaccurate recognition. As available systems show their best accuracy at around 90%, misjudgement happens frequently as illustrated in the taxi ride of Figure 1. If the recognition system misjudges, the services on top get confused. Confusions always lower the service quality but some confusions can be even more critical such as a false data charge. The third is incomplete recognition. As depicted in the subway ride of Figure 1, if the person makes a move inside a subway, e.g., to find a better seat, most available systems conclude that the person is getting off the subway. If the person is again determined to be inside a subway after that happening, the person is considered to take another ride. This type of misjudgement comes from the incomplete design of recognition systems that fail to cover all possibilities. This can cause severe performance drop at some services such as the wireless channel prediction.

In order to have a reliable transportation mode recognition system that overcomes all those performance issues, we propose VehicleSense, a sound-based transportation mode recognition system which is accurate, fast, and low-powered. VehicleSense makes a recognition based on its learning about the

characteristic spectral sound features of popular transportation modes involving subways, buses, and taxis. With our judicious design of the system that takes hierarchical accelerometer based triggers as well as sound pre-processing modules into consideration, we find that the implementation of VehicleSense shows highly reliable performance. Our 186-hour log of sound and accelerometer data collected by seven different Android smartphone models confirms that VehicleSense achieves 98.2% of recognition accuracy with only 0.6 seconds of latency, while persistently consuming only about 26.1 mW on average for all day monitoring. It also deals with the privacy concern that may arise from using the microphone by suggesting a privacy protection sampling mode that completely precludes VehicleSense from extracting a single word from the sampled sound.

2 Related Work

In accordance with our work, we brief previous transportation mode recognition systems classified by the sensors they mainly use as follows:

2.1 Accelerometer only

The 3-axis accelerometer equipped in every recent smartphone is the most popular sensor for the activity recognition. The popularity is based on its low power consumption compared to other sensors. Even in the area of transportation mode detection, there has been a number of work that only utilizes the accelerometer and tries to capture subtle characteristics observed while being on a specific vehicle. Wang et al. [19] introduced a transportation mode recognition technique based on Bayesian learning and identified a practical concern. That is the device orientation problem which largely confuses the per-axis Bayesian learning. In order to overcome the problem, they proposed to use orientation-independent features such as combined magnitude and demonstrated that the accuracy of distinguishing subway, car and bus can be as high as 70%. Hemminki et al. [17] tackled another important issue in accelerometer based recognition, which is so-called gravity factor elimination. In a nutshell, it is the problem of subtracting the gravity value (i.e., 9.8 m/s^2) which is dynamically distributed to axes while moving. They proposed and implemented the concept of peaks of accelerometers characterizing acceleration and deceleration patterns of motorized transportation and achieved about 85% of accuracy in distinguishing transportation modes of bus, car, subway, train and tram. But the computational complexity involved in the proposed gravity factor elimination is demonstrated to increase the power consumption as high as 85 mW.

2.2 Accelerometer and GPS

Reddy et al. [18] created a device position independent activity recognition system exploiting the built-in GPS receiver along with the accelerometers. Thanks to the GPS support, the proposed system identifies whether the user is stationary, walking, running, biking, or being in a motorized transport with a relatively high accuracy, 93.6%. However, the system is incapable of specifying which type of motorized vehicles is in use. More recent work [16] with the GPS support constructs a database of transit route information from the data published by various transportation agencies and detects the type of vehicle in use by performing a geographical pattern matching. [16] is shown to obtain about 90% of accuracy in distinguishing route-determined public transportations such as bus and subway, but accompanies severe battery drain from continuous location information acquisition.

2.3 Accelerometer and Barometer

Unlike others, [6] paid attention to a low-powered transportation mode detection using an embedded barometer in combination with the accelerometers. It is shown to consume 32 mW less compared to an accelerometer-only approach in detecting idle, walking, or being in a vehicle. However, the answers on how to continuously calibrate the barometer and how to improve the detection accuracy which is reduced down to 81% by the impreciseness of the embedded barometer are left as future work.

2.4 Accelerometer and Microphone

There exists a recent study [13] that exploits the microphone to detect when a person gets on and off a bus by letting a smartphone listen to the beeping sound from IC card readers of buses. However, because the original goal of the system is to improve the prediction quality on the bus arrival time from crowd-sourcing, its generalization to other transportations has not been considered. A more relevant work [8] utilizes the microphone of a smartphone along with the assistance of the accelerometers to recognize the type of the vehicle in use. The proposed system shows relatively good accuracy, 92%, but it is incapable of detecting when the user gets off from a vehicle. Also, distinguishing whether a movement is made inside or outside a vehicle is out of its scope.

3 Motivation

In order for various context-aware services to best utilize the transportation mode recognition system, the system should satisfy the followings: 1) *high accuracy*, 2) *instant recognition*, and 3) *low power consumption*. In each of these three perspectives, we explain why the existing systems have difficulties in meeting the requirements.

3.1 Recognition accuracy

Most activity recognition systems rely on 3-axis accelerometers to make an inference on the kinetic motions of the device owner. If it can be assumed that the kinetic movement of the owner is rooted from the target activity to recognize, accelerometers will do the right job as many of activity recognition systems work well in detecting human motions. However, such an assumption is not always true especially in the transportation mode recognition because kinetic motions sensed by accelerometers are affected by both human and vehicle motions. Thus, recognizing the vehicle in use becomes much more difficult while a person is walking or budging compared to sitting or standing still. In practical systems, this causes a recognition accuracy drop as people frequently make movements even inside a vehicle. A quick detour to this accuracy drop problem is to monitor the situation with accelerometers for a longer duration and make a delayed decision. However, this detour leads to wrong decisions when the duration of a ride is very short or when a person makes continuous movement. More fundamentally, it gets always confused when the smartphone is in use and gets persistent wiggling from handshaking and touch actions.

The barometer, which tells whether a smartphone is located below or above the ground level based on the air pressure difference, gives a little help to the accelerometers. But this sometimes brings more confusion since a subway route makes its run in the ground level and even a bus can go through an underground tunnel. More importantly, the barometer-based mechanism is prone to produce a lot of confusions in a city with many ups and downs as its reading is dependent on the absolute altitude of the measurement location as well as the weather condition at the moment.

Because of the aforementioned reasons, it is very hard to design a system that achieves over 90% of accuracy with the accelerometer and the barometer [6]. GPS can largely assist those sensors conceptually but its weak reception in downtown areas and no reception in underground makes its practicality restricted.

3.2 Recognition latency

Recognition latency is a key performance metric of a recognition system especially designed for applications that require timely services. Given that a delayed service can be often regarded as an unnecessary service, the recognition latency can even cause degradation of the perceived accuracy of the recognition

system. If we define a timely recognition as a decision made within 3 seconds from the moment at which some action happened (e.g., getting on/off a vehicle), there is no existing system that satisfies this tight requirement. The typical latencies of accelerometer-based recognition systems range from 100 to 200 seconds [36, 17]. There had been several proposals that use only a few seconds of accelerometer sampling, however they turned out to suffer from poor accuracy, and thus are no longer popular. If a recognition system is designed to detect abrupt deceleration and acceleration from breaking and driving [17], its recognition latency could reach as high as several minutes. GPS-based recognition also needs at least several tens of seconds and sometimes needs much longer given that a vehicle makes no move at the red light during one minute or two.

3.3 Energy efficiency

Energy efficiency is another key performance metric for a recognition system as the system needs to run in the background all the time. Given this necessity of continuous sensing, GPS which is known to consume huge power is certainly a bad choice. Using the Monsoon power monitor [10], we measure that GPS in Samsung Galaxy S6 [26] consumes about 317 mW. Thus, GPS can be regarded useful only when it is intermittently used to support decisions made by low power sensors such as the accelerometers and the barometer that consume only about 5.23 mW and 2.3 mW when sampled at 50 Hz in Samsung Galaxy S6. However, although the accelerometers and barometer are known as low-powered, recognition systems that perform continuous processing over sampled data tend to consume high power by keeping the mobile AP (application processor) awakened. This is unavoidable unless a trade-off is made between the power consumption and the decision making interval that affects both latency and accuracy.

As explained, most of the sensors used in the literature have their own sets of limitations. Therefore, instead of combining such sensors, we opt to exploit the built-in microphone readily available in every smartphone. The microphone collects sound information at the sampling rate of 16kHz in most smartphones and is able to uniquely recognize the characteristic sound patterns of various motorized vehicles such as bus, taxi, and subway. Considering that a person is highly likely to recognize what the transportation mode is while closing her eyes in a vehicle before any acceleration or deceleration happens, the microphone can be the most informative sensor that achieves high recognition accuracy along with nearly immediate recognition. Also, the power consumption of the microphone is manageable as it takes 174.51 mW in Samsung Galaxy S6, which is substantially lower than that of GPS and can be further optimized by the help of accelerometers.

In the following sections, we explain in detail how and why our proposed system relying on the microphone can achieve high accuracy, low latency, and low power consumption all at once.

4 Characteristics of Transportation Sound

4.1 Frequency-Domain Characteristics

How can a person sense if she is in either of bus, taxi, or subway only by the sound? What are the characteristics of the unique sound of each transportation? We try to answer these questions by performing the spectrogram analysis [34] which is a graphical representation of STFT (short time Fourier transform). As will be explained later in Section 7, we collected the empirical sound data of 186 hours captured from bus, taxi, and subway in South Korea and this data is used for the analysis. Figure 2 shows sample spectrograms of 20 minutes from a bus, a taxi and a subway. As the figures clarify, transportations have different spectral patterns. Moreover, Figure 3 confirms that the unique patterns are persistently observed over time, for instance, 400 minutes each in the figure. We discuss about the patterns below.

As shown in Figure 2 (a), a bus has high spectrum density in the range of 250~400 Hz while moving. Whenever it stops at a red light or at a bus station, the dispersion of spectrum density over frequency domains narrows down and gets its peak at around 300 Hz. Our experimental sound recording from various spots of a bus reveals that this characteristic low frequency sound comes from the engine of the bus. In South Korea, almost all city buses are equipped with a variety of 11,000cc (11 liter) CNG (compressed natural gas) engines with about 110 $kg \cdot m$ torque from several manufacturers, which satisfy the CO_2 emission regulation of South Korea. Our random rides on buses show that the concentration of spectrum density is observed at 250~400 Hz as in Figure 2 (a). This pattern is persistently observed as in Figure 3 (a).

Most taxis in South Korea are with LPG (liquefied petroleum gas) based engines and produce relatively low noise. As expected, we observe spectrum density dispersed widely over a larger frequency range with minor peaks as shown in Figure 2 (b). Low frequency components at 200~300 Hz are higher than other components, but they are modest. This pattern is persistently observed over time as in Figure 3 (b).

According to [30], the subway trains powered by distributed electric motors is deployed more than 55 countries around the world. The subway trains of this type that also runs in South Korea produce unique electric noise patterns at about 100 Hz and 200 Hz as shown in Figure 2 (c). When the subway train stops at a station, the pattern gets weaker but is still there. Figure 3 (c) confirms the persistency of this observation.

4.2 Robustness of Sound Sensing

One concern about using sound information to distinguish transportation modes arises at its robustness over vocal noises such as conversations, radio broadcasting, and station announcement. It is known that

human voices range widely from 20 to 5000 Hz, but are mostly concentrated at 100 ~ 200 Hz [33]. Figure 4 that compares the power spectral density over frequencies of an announcement made at a bus and the sound samples from transportations confirm that the dispersion of vocal noises largely deviate from the patterns of transportations. Thanks to this observation, we later confirm that our proposed system does not suffer from vocal noises in achieving high accuracy by its evaluation with 186 hours of sound data collected in the wild where various vocal noises prevail.

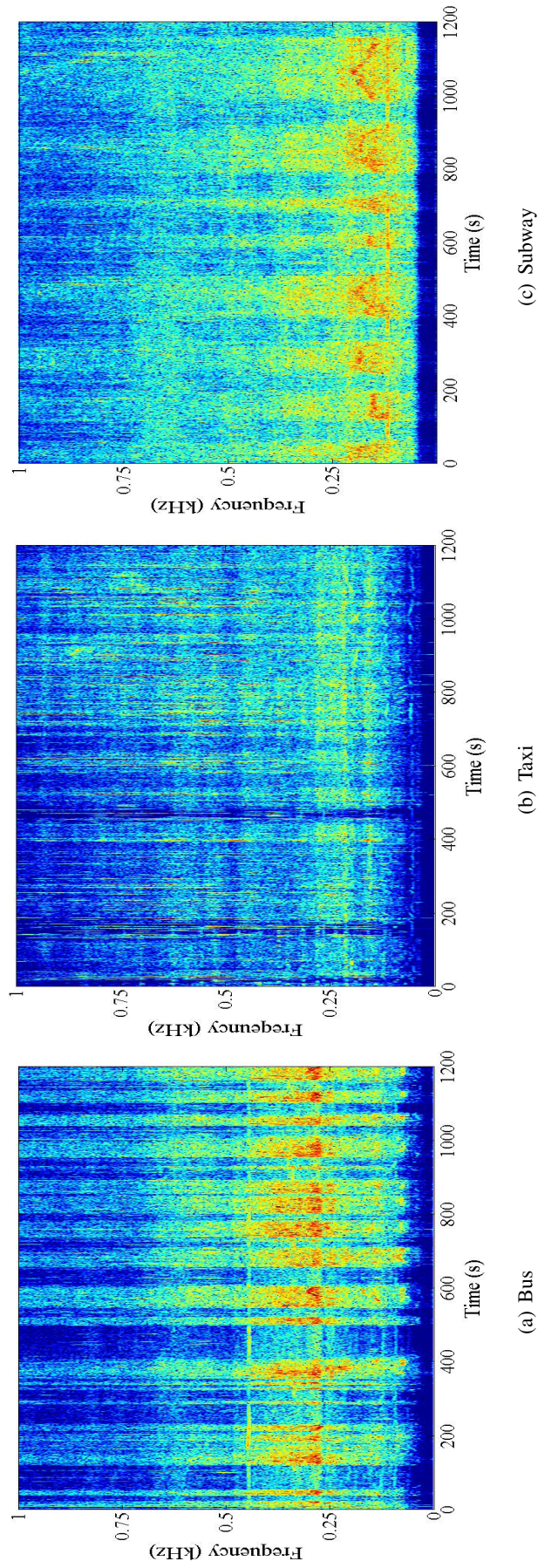


Figure 2: Spectrogram of three transportations: bus, taxi, and subway. The red color indicates high spectrum intensity.

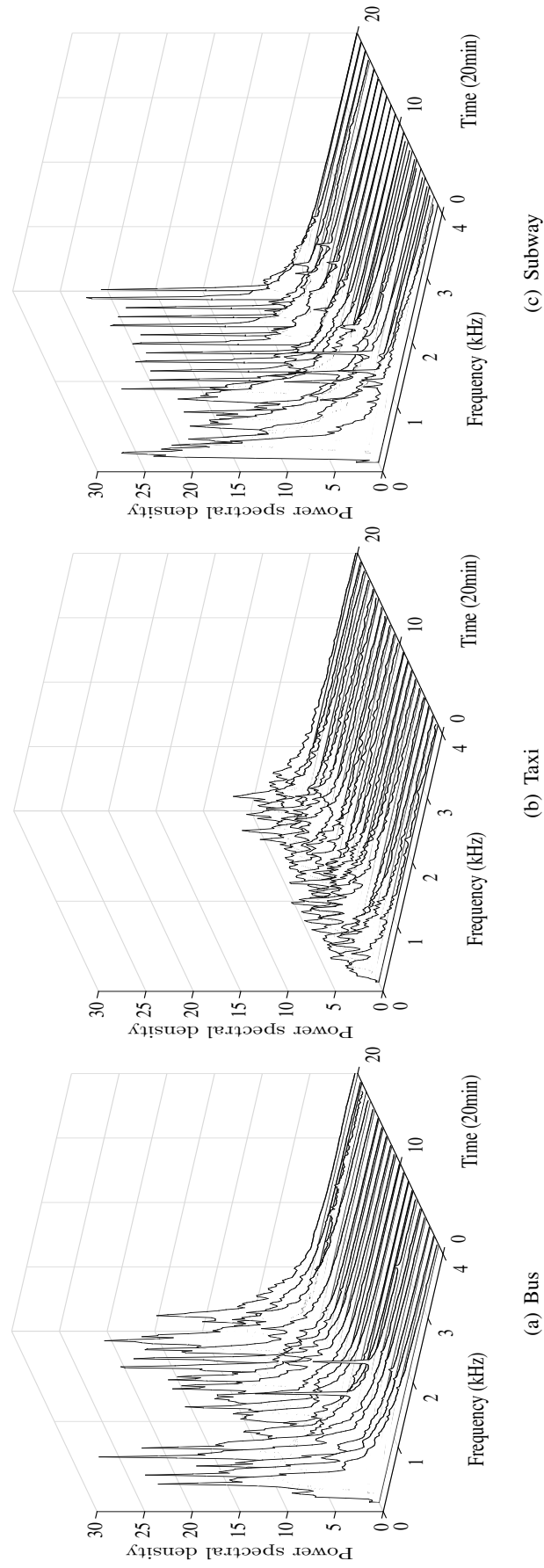


Figure 3: The unique sound patterns in the perspective of power spectral density are persistently observed over time for (a) bus, (b) taxi, and (c) subway.

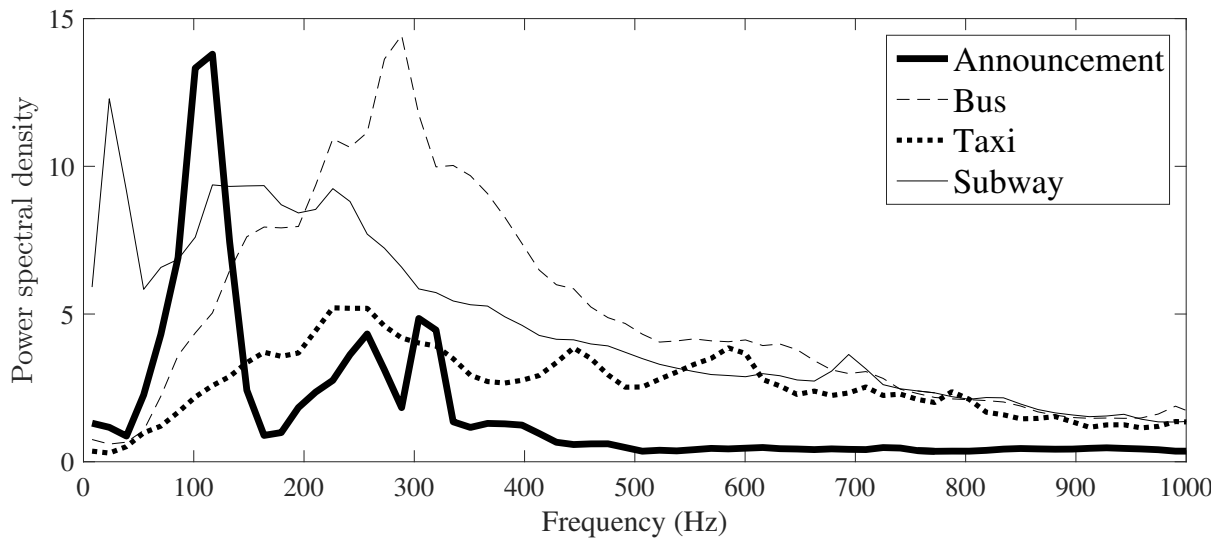


Figure 4: The comparison of power spectral density over frequencies of an announcement made at a bus and the sample sound captured in a bus, a taxi, and a subway.

5 System Overview

In order to achieve all three objectives in the motivation section at once, we design a new system, VehicleSense that immediately and precisely recognizes the type of vehicle on which the owner of a smartphone is riding based on the sound characteristics of candidate vehicles, with only minimal power consumption. Given that the microphone consumes non-negligible amount of power VehicleSense is designed to be such that it turns on the microphone during a very short period of time (e.g., 0.5 second) only when the sound information is needed. By doing so, we eliminate the need for continuous sound sensing that results in the realization of an extremely low power recognition system. Also, given that existing transportation mode recognition systems showed about 90% as their maximum accuracy. VehicleSense is designed to apply filtering techniques over the short-duration sound captured by the microphone for dramatic accuracy improvement. We explain the details of the system design in two aspects: 1) low power architecture and 2) accuracy improvement.

5.1 Low Power Architecture

VehicleSense aims at minimally utilizing the microphone for energy efficiency, while not losing any small detail of the transitional behaviors related to transportation modes. In order to do so, we find the exact moments of capturing the sound and also find time duration of no interest. The moments for capturing the sound can be exemplified by the following scenarios: 1) A person is walking into a vehicle and then stopped, and 2) a person was determined to be inside a vehicle and now keeps walking to somewhere. The first example explains the situation where she is getting on a vehicle and then sits or stands at a place (e.g., in a bus or in a subway). The second example visualizes the situation where she starts to walk out from the vehicle or starts to relocate inside the vehicle. We ensure that these two cases are the only moments to capture the sound and the time duration between these moments are of no interest. This makes sense considering that it is of no use to continuously capture the sound while she keeps staying inside a vehicle or outside a vehicle. To realize this selective and timely sound capture, we design hierarchical triggers as shown in Figure 5, which monitor the accelerometers and signal the microphone whether or not to capture the sound. The hierarchical triggers have two levels whose first level trigger recognizes whether or not a person has strong kinetic energy representing that the person is walking/running or being stationary. The first level trigger works with a simple thresholding method over the norm-2 magnitude from the multi-axis accelerometers. The threshold can be learned from activity data, for instance, our data logs and it is known by [14, 32] that the thresholding is precise enough to distinguish the kinetic states of a person. The computational complexity of the first trigger is minimal, thus it is even possible to run this trigger in any low power co-processors such as Qualcomm Hexagon 680 with a single core running at 1GHz, which

is equipped in Samsung Galaxy S7 series[27]. When the first level trigger detects that the kinetic energy of a person has changed from high to low which we call by *Go-and-Stop*, the second level trigger kicks in and recognizes if the person is on a transportation or not, irrespective of what the transportation is. It is known by [17] that the accelerometers experience small perturbation while being at a transportation and this subtle but persistent fluctuation in the accelerometer readings can distinguish whether the person is being stationary on a vehicle or at a static place like an office or home. If it is indeed possible to detect if the person is on a vehicle by accelerometers, it is surely the right moment to turn on and capture the sound for the detailed recognition of what the type of the vehicle is. This is how VehicleSense mainly utilizes the hierarchical triggers. The only exception happens when the person was determined to be on a vehicle and is now walking. The reason why this can be of an exception is that although the second level trigger is not activated by the movement, VehicleSense should be continuously determining by the microphone whether the person is moving inside the vehicle or moving out from the vehicle. Thus, we let the first level trigger signal the microphone upon detecting continuous walking over a certain time period (e.g., 5 seconds) and let the system make decisions repeatedly until the person is determined to be out from the vehicle or until the person becomes stationary again inside the vehicle. Note that this exception does not incur high power consumption as a person walks inside a vehicle only for a short time and a long walk toward outside a vehicle will soon be detected as being outside. All these operational cases are depicted in Figure 5.

5.2 Accuracy Improvement

In the design of VehicleSense, we choose to use the short-duration sound information whose length is less than 1 second which will be explained in Section 7 with the study of the sampling duration over the recognition accuracy. For maximizing the accuracy for a given sampling duration, we carefully design the pre-processing block, the feature extraction block, and the decision block of the sound classifier shown in Figure 5. First, the decision block has been tested with various machine learning algorithms such as Naive Bayesian, Multi-dimensional Bayesian, HMM (hidden Markov model), and SVM (support vector machine). The SVM has been also tested with a number of popular kernel functions such as RBF (radial basis function), polynomial, and sigmoid. Unlike [8] in which HMM was adopted for sound classification, we find that SVM with RBF kernel persistently outperforms available systems throughout extensive folding tests¹ over our sound logs. Based on this observation, we opt to use SVM for the decision block. It is known by [4] that once support vectors for the SVM is learned, every decision is made with little computation, which is nothing but the summation over matrix multiplications. Thus,

¹A folding test means a test in which the learning period and the decision making period are separated in the data, and shuffled for more reliable evaluation.

using SVM can still satisfy the low power requirement. Second, the feature extraction block has been tested with diverse sets of popular sound features such as signal power, variance, peak interval, spectral centroid, spectral flatness and so on. We have also tested the interaction between the feature extraction block and the pre-processing block by applying various pre-processing techniques such as LPF (low-pass filter), BPF (band-pass filter), and mel-scale frequency division. From our extensive combinatorial tests, we find that combining vectorized per-frequency features that are pre-processed by a specific form of non-linear pre-processing filter maximizes the accuracy even with the short-duration sound. VehicleSense is our crystallized system made up of these judiciously designed components.

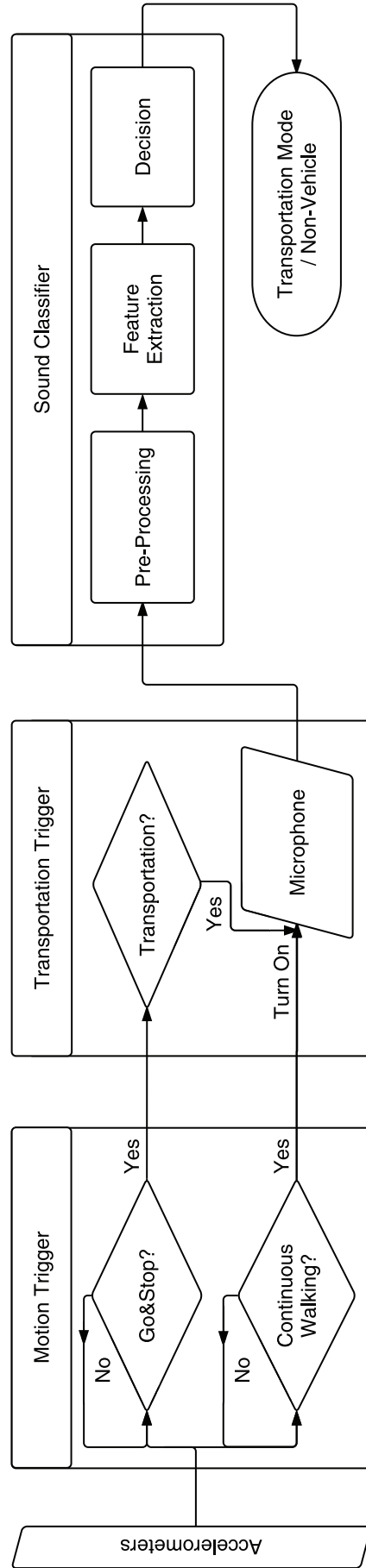


Figure 5: The block diagram of VehicleSense that consists of two hierarchical accelerometer-based triggers and the sound classifier.

6 Triggers and Filters

In this section, we explain algorithmic details of triggers and sound filters and also provide justifications on why such algorithms are adopted.

6.1 Motion Trigger

The first level trigger, **motion trigger**, is used to identify either of 1) the moments for the users changing their motions from walking to being stationary (i.e., sitting or standing), or 2) the moments for the users who have walked for longer than a preset duration of time (t_{cont}), for instance 5 seconds. For this, we let accelerometers perform continuous sensing over the magnitude that is defined as a square sum of chronologically measured three axial readings with the gravity factor (G) subtracted. As we do not focus on a specific axial data of the accelerometers, ruling out the gravity factor does not require any coordinate transformation and becomes as simple as a subtraction operation. Accelerometer readings are sometime unreliable mostly due to sudden motions and glitches of the chipset. To mitigate the reliability concern, we take the weighted moving average of the magnitude calculated from the average of 10 readings that take 200 ms for the accelerometers of 50 Hz sampling rates. Most smartphones of today such as iPhone 6s plus, LG V10, and Galaxy S6 can operate their accelerometers in this sampling rate. Galaxy S6 can even operate its accelerometers at 200 Hz, but we find that sampling rate beyond 50 Hz does not improve the motion detection accuracy as the time scale of changing motions for a human is no faster than 10 ms. Motion trigger recomputes the moving averaged magnitude at every 200 ms from the continuous sensing and compares the magnitude with the personalized preset thresholds for walking (T_w) and stationary motion (T_s) and resets the flags, $flag_w$ and $flag_s$, to be either of 0 or 1. The reason why we use two thresholds is to virtually eliminate the case where the trigger misses out an event that is possibly the moment of our interest. For this, the thresholds should satisfy $T_w < T_s$. These personalized preset thresholds are learned from our supervised user activity traces which will be explained more in Section 7. If the motion trigger detects that both of $flag_w$ and $flag_s$ are up, it calls the second level trigger, **transportation trigger**. If it detects $flag_w$ is up for more than t_{cont} seconds, it invokes the sound-based decision engine, **sound classifier** as described in Algorithm 1.

6.2 Transportation Trigger

The transportation trigger examines if a person is on a vehicle at every moment when Go-and-Stop event is observed indicating that there exists a possibility that a person gets on a vehicle and finds a place to stay. This trigger is grounded on the fact that transportations have vibration patterns that are not observable at a fixed place. Such patterns are detected by SVM over three features extracted from accelerometer

Algorithm 1 Motion trigger

```

1: while 1 do
2:    $Mag_a(t) := a_x(t)^2 + a_y(t)^2 + a_z(t)^2 - G^2$ 
3:   for every  $\Delta = 200\text{ms}$  do
4:      $\overline{Mag}_a(t) := \text{average}(Mag_a(t))$  for 200ms
5:      $Mag_a^{MA}(t) = 0.5 \cdot \overline{Mag}_a(t - \Delta) + 0.5 \cdot \overline{Mag}_a(t)$ 
6:      $flag_w(t) = (Mag_a^{MA}(t) > T_w) ? 1 : 0$ 
7:      $flag_s(t) = (Mag_a^{MA}(t) < T_s) ? 1 : 0$ 
8:     if  $flag_s(t) == 1 \ \&\& \ flag_w(t - \Delta) == 1$  then
9:       Go to Transportation Trigger
10:    else if  $flag_w(t) == 1$  remains for  $t_{cont}$  seconds then
11:      Go to Sound Classifier

```

readings during a preset time window (t_{acc}): 1) accumulated magnitude (AM), mean variance (MV), and several frequency components in the FFT (fast Fourier transform) of the accelerometer readings (SF). We use t_{acc} as 2 seconds, which is empirically shown effective from our traces. Note that t_{acc} does not delay the recognition as the accelerometer readings are continuously made by the motion trigger and the readings store can be used for the transportation trigger.

AM quantifies the extent of the accelerometer readings caused by large motions. This is done by accumulating the magnitude values of accelerometers, which are larger than a threshold denoted by T_{AM} , thus eliminating all minor movements during t_{acc} . By its design, AM can catch sudden motions such as standing up or sitting down. We use $96 (m/s^2)^2$ for the T_{AM} , which is also empirically chosen from our traces. MV measures the variation of accelerometer readings from the mean value during the time window, which is obtained by summing all per-axis variances over t_{acc} .

SF that analyzes spectral characteristics of accelerometer readings understands regular vibrations dispersed over various frequency components. We let SF focus especially on the spectrum range from 1 to 20 Hz where the most vibration patterns of transportations are concentrated.

We train these three features with SVM and let the SVM output one of the following states: being stationary at a fixed place (F), walking (W), being stationary in a vehicle (V). Thus, when V is provided, we invoke the sound classifier.

6.3 Sound Classifier

The sound classifier consists of three blocks: 1) pre-processing, 2) feature extraction, and 3) decision. We focus on explaining about our pre-processing technique as other blocks mostly involve standard SVM-based machine learning techniques. As soon as a trigger wakes up the microphone, the sound classifier samples a short duration sound data and start pre-processing. Our pre-processing aims at

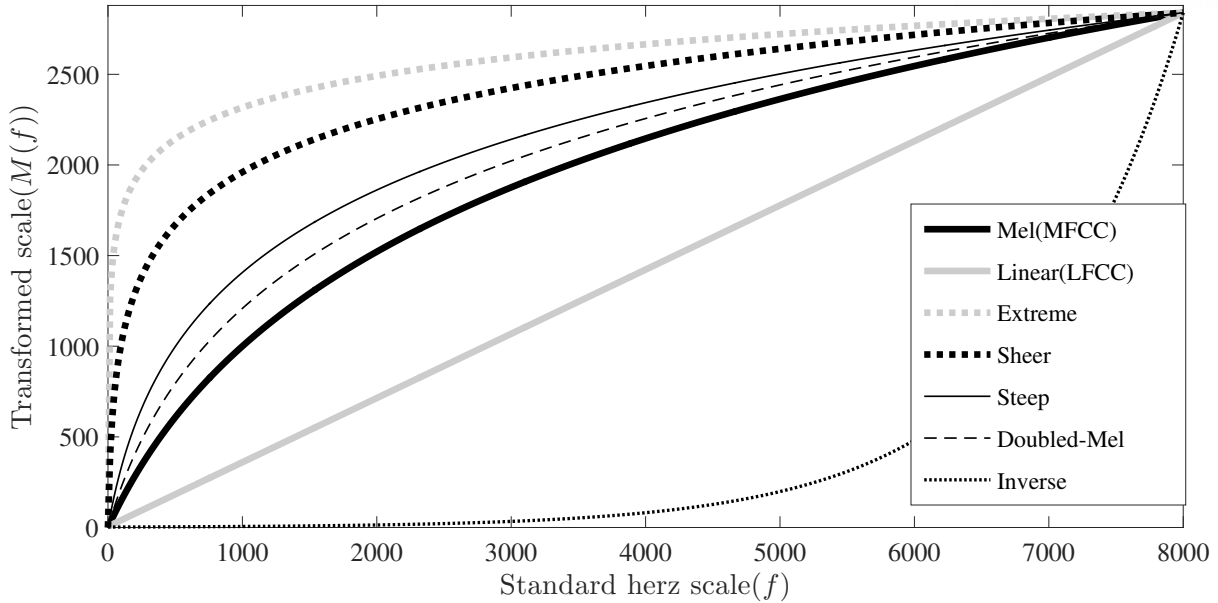


Figure 6: Illustration of the seven different non-linear filters and their transformed scales ($M(f)$) over the standard herz scale (f).

maximizing the differences between the sound characteristics of transportations in the feature space. Typically twisting the feature space to widen the distance between the data points to classify is done by a kernel function in the SVM, but this has a fundamental difference to the pre-processing. That is because once the vector coordinates of the data points are quantized, the amount of information included in the coordinates are unchanged even with a coordinate transformation. However, the pre-processing can affect the amount of information included in the data coordinates. MFCC (mel frequency cepstral coefficient) [2] is a popular signal processing filter which is widely used in the speech recognition field. The main idea of MFCC is to transform the sound signal in the frequency domain from hertz (Hz) scale to mel scale that is known to emphasize the frequency bands in which the most information of human voice is concentrated.

Inspired by MFCC, we try to design our own scales with diverse non-linear filters that may best highlight the characteristic sounds from transportations. In Figure 6, we depict our original filters, **Steep**, **Extreme**, **Inverse**, **Doubled-mel**, **Sheer** compared to existing filters, **Mel** (MFCC) and **Linear** (LFCC: linear frequency central coefficient) [35], whose transform formulas ($M(f)$) from the herz scale (f) are listed below. All these scales are designed to have the same start and end points and are linearly binned to be vectorized by the feature extraction block.

Our test results over the empirical sound data in different scales are quite interesting. Given that the maximum recognition accuracy we can obtain from **Mel** is 94.9%, **Linear** and **Inverse** show 92.7%

and 86.0% of the accuracy, which are disappointing. However to our surprise, **Extreme**, **Doubled-Mel**, **Sheer**, and **Steep** give about 97.1%, 98.25%, 99.2% and 99.3% as their accuracies, respectively. This result implies that somewhere in-between **Sheer** and **Steep**, there may exist the best pre-processing filter for the transportation mode recognition. Understanding the reason why this shape of frequency filter is beneficial to differentiating the transportation sounds is of an intriguing question. However, demystifying the reason is beyond the scope of this work so we leave it as our future work. We just take **Steep** as our default pre-processing filter. We show more detailed results in Section 7.

$$\mathbf{Mel(MFCC)} : M(f) = 2595 \cdot \log_{10}(1 + f/700)$$

$$\mathbf{Linear(LFCC)} : M(f) = 0.3550f$$

$$\mathbf{Extreme} : M(f) = 686 \cdot \log_{10}(1 + 10f)$$

$$\mathbf{Sheer} : M(f) = 1730 \cdot \log_{10}(1 + f/10)$$

$$\mathbf{Steep} : M(f) = 1818 \cdot \log_{10}(1 + f/175)$$

$$\mathbf{Doubled-Mel} : M(f) = 2141 \cdot \log_{10}(1 + f/350)$$

$$\mathbf{Inverse} : M(f) = 20 \cdot \exp(f/2595)$$

7 Evaluation

In this section, we first explain the data we have collected to verify the performance of VehicleSense and then extensively evaluate VehicleSense through the data in various perspectives. Lastly, we compare the performance of VehicleSense with that of an accelerometer-based recognition system.

7.1 Data Collection

For our study, we have collected the microphone and the accelerometer data from seven different smartphone models whose total length is 183 hours, experimented by 8 testers. For the microphone, we set the sampling rate as 16 kHz and the encoding method as 16 bit PCM (pulse-code modulation). For the accelerometers, we set the sampling rate as 50 Hz. While we collect the data, we have also acquired the activity labels provided by the testers at every moment of changing their motions. Each of the activity labels we have in the dataset is one of the following five motions: being stationary at a fixed place, walking, in a bus, in a taxi, and in a subway. Note that we minimized the labeling effort by the testers by asking them to keep each motion sustained at least for 20 minutes. Table 1 summarizes the per-device data length in hours for the candidate activities. In addition to this dataset, we have collected another dataset from the wild, which only recorded the accelerometer readings from 110 randomly selected volunteers of various age bands, jobs, and cities recruited from an Internet community. The data collection lasted for 8 days and the microphone data was excluded for their privacy concern. We exploit this data only for the evaluation of accelerometer based triggers.

7.2 The Performance of Triggers

The accuracy of triggers

We first evaluate the accuracy of the motion trigger. Figure 7 shows the CDFs of the magnitude values extracted under two motions: stationary (either being inside or outside a vehicle) and walking. We set the stationary and walking thresholds to be $25.7(m/s^2)^2$ within and $5.1(m/s^2)^2$ with which 99.65% of walking and 99.95% of stationary motions are captured. These stationary and walking thresholds produce false positives of 25.5% and 26.8%, but this choice is made by our intention that aims at minimizing the false negatives in the hope of missing out virtually no transportation related events such as getting on and off. With this design, we find that the missing rate of the riding events goes below 0.68%.

We then evaluate the performance of the transportation trigger which is learned by SVM with RBF kernel to distinguish being stationary at a fixed place from being stationary at a transportation with our accelerometer dataset. The detailed classification accuracy values measured by three popular metrics [15],

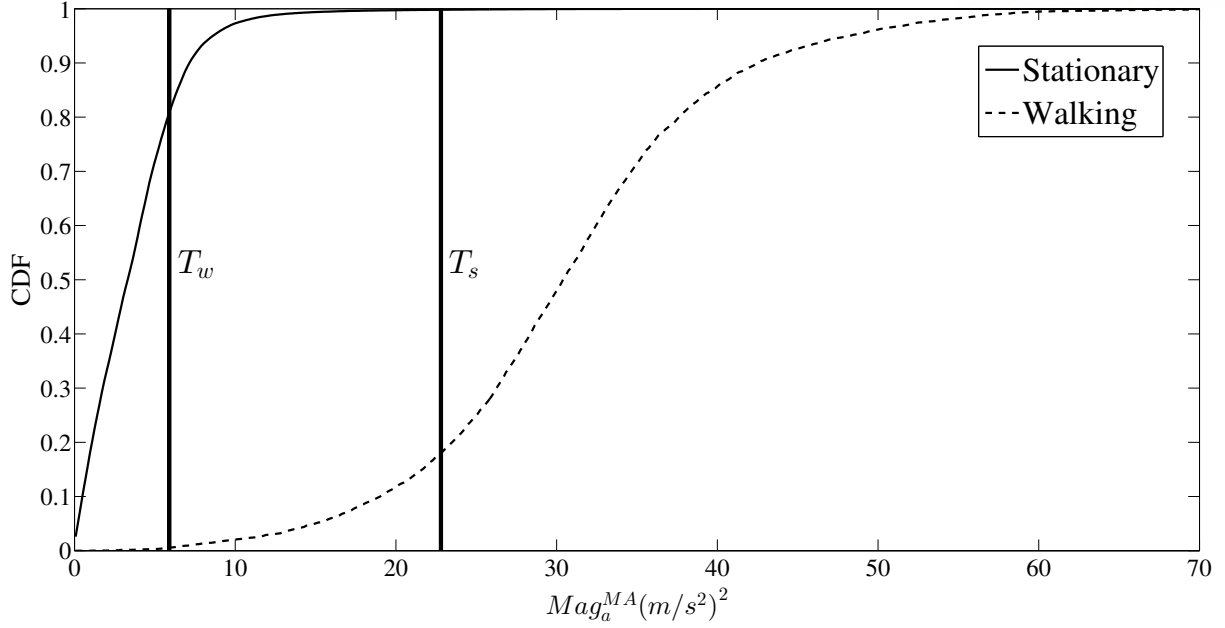


Figure 7: CDFs of the Mag_a^{MA} values from the accelerometers under stationary (either being inside or outside a vehicle) and walking motions. We also depicted two thresholds, T_w and T_s .

recall, precision, and F_1 score (the harmonic mean of precision and recall) are shown in Table 2. The most important performance metric as a trigger is again the missing rate which is denoted by FNR (false negative rate). FNR is nothing but $1 - \text{recall}$. Table 2 confirms that FNR is as low as 0.4%. Note that false positives do not incur critical problems as they only lead to a little more power consumption of the system.

Energy efficiency of triggers

We here evaluate how much the proposed triggers help in reducing power consumption of VehicleSense in comparison with existing continuous sensing methods. In order to understand the expected power consumption ($\mathbb{E}[\mathcal{P}]$) from VehicleSense for the realization of all day monitoring, we characterize it by estimating the probabilities of seeing specific events in a day. When we denote the instantaneous power consumption for the motion trigger (MT), the transportation trigger (TT), and the sound classifier (SC) by \mathcal{P}_{MT} , \mathcal{P}_{TT} , and \mathcal{P}_{SC} , the following equation holds:

$$\mathbb{E}[\mathcal{P}] = \mathcal{P}_{MT} + p(W_{cont}) \mathcal{P}_{SC} t_{SC} + p(W \rightarrow S) \{ \mathcal{P}_{TT} t_{TT} + p(V|S) \mathcal{P}_{SC} t_{SC} \},$$

where $p(W \rightarrow S)$, $p(V|S)$, and $p(W_{cont})$ denote the probability of observing a transition from walking to being stationary, the probability of being on a vehicle given that a person is being stationary, and the probability of observing the motion of continuous walking for $t_{cont} = 5$ seconds. Note that all these

probabilities are calculated from the discretized events assessed at every second throughout a day. Also note that t_{TT} and t_{SC} are the activated time duration in seconds, of the motion trigger and the sound classifier for each call. Notice that If we have no trigger system, the power consumption of VehicleSense simply becomes $\mathbb{E}[\mathcal{P}_{no-trigger}] = \mathcal{P}_{SC}$. If we only exploit the transportation trigger, $\mathbb{E}[\mathcal{P}_{1-trigger}] = \mathcal{P}_{TT} + p(V)\mathcal{P}_{SC}$ holds. Given the baseline statistics of MT, TT, and SC provided in Table 3 and the average event probabilities from our 110 user dataset, $p(W \rightarrow S) = 0.064$, $p(V|S) = 0.569$, and $p(W_{cont}) = 0.032$, the power consumption of VehicleSense becomes $\mathbb{E}[\mathcal{P}] = 26.11$ mW that outperforms $\mathbb{E}[\mathcal{P}_{no-trigger}] = 270.92$ mW and $\mathbb{E}[\mathcal{P}_{1-trigger}] = 183.62$ mW. Note that for the measurement of baseline statistics, we implemented VehicleSense on Samsung Galaxy S6 running Android 5.1.1 Lollipop. Figure 8 (a), (b), and (c) confirm that the $p(W \rightarrow S)$, $p(V|S)$, and $p(W_{cont})$ obtained from 110 users individually are different but have certain coherence. We can estimate the worst case power consumption from these statistics, which becomes $\mathbb{E}[\mathcal{P}] = 66.21$ mW. This is still far less than no-trigger and 1-trigger systems.

7.3 Performance of Sound Classifier

We here evaluate the performance of the proposed sound classifier in various perspectives. We first evaluate how much the non-linear filters affect the recognition accuracy and suggest the best non-linear filter. With the selected filter, we further evaluate the robustness of VehicleSense by varying the sampling duration of sound data and find the minimal sampling duration that barely loses the accuracy performance. With the selected time duration, we then assess the reliability of VehicleSense over various smartphone models by performing trace-drive simulations on the data traces collected by each model.

The recognition accuracy by non-linear filters

In Section 6, we have proposed non-linear filters that pre-process the sound data in the hope that we get the most out of the processed data. Table 4 summarizes the recognition accuracy from non-linear filters in three performance metrics, precision, recall, and F_1 score. As the Table clarifies, we can immediately notice that **Inverse** which is the only exponential shape filter shows the worst performance compared to others in a logarithmic shape. Considering that the important frequency features are concentrated in the frequency range under 1 kHz, this result is of no surprise. More interesting observation is on the pattern of the accuracy change over the logarithmic shape filters that become more convex in the order of **Linear**, **Mel**, **Doubled-Mel**, **Steep**, **Sheer**, and **Extreme**. As Table 4 dictates, the accuracy gradually increases and peaks at **Steep** by showing overall 99.4%, and then again gradually decreases. The well-known pre-processing technique, **MFCC** shows relatively poor performance which is about 6% lower than that from **Steep**. Note that we let the sampling duration of the sound data be 0.5 seconds in obtaining Table 4,

but we confirmed that this trend remains the same for different sampling durations.

The recognition accuracy by sampling durations

We have discussed in Section 3 that the recognition latency is one of the most important performance measures in the transportation mode recognition systems. In order to make VehicleSense conclude as quickly as possible, we test the trade-off between the sampling duration and the recognition accuracy as shown in Table 5. The tested durations range from 0.1 seconds to 9 seconds. Table 5 confirms that a longer sampling duration leads to higher accuracy, but at the same time the improvement in the accuracy is marginal beyond 0.3 seconds. A more practically valuable observation from Table 5 is that the accuracy is kept very high even at 0.3 seconds of sampling duration, which is much smaller than the sampling duration of 3 seconds suggested in a previous study [8]. By taking both the accuracy and the time into consideration, we opt to adopt 0.5 seconds as the default setting.

The privacy protection mode for VehicleSense

We briefly describe how we can mitigate the privacy concern from using the microphone. The gist is that we let the microphone alternatively sample sound and put a mute at each 0.1 seconds so that the sampled sound becomes no longer comprehensible. We call this *privacy protection (PP)* mode. For instance, if we use 0.5 seconds of sampling in *PP* mode, say 0.5 (*PP*), each sound sample has three 0.1 seconds of data and two 0.1 seconds of silence. We apply majority voting on three data pieces. To our surprise, as shown in Table 5, it gives comparable performance with that of 0.5 seconds of normal sampling. We confirm that we hear nothing from 0.5 (*PP*).

The recognition accuracy by smartphone models

Smartphones of different generations or from different manufacturers may have different microphone chipsets that may have slightly different frequency responses. We test how reliable VehicleSense is over different smartphone models with the aforementioned selections of the pre-processing filter and the sampling duration. Figure 9 summarizes the average precision, recall, and F_1 score with 95% confidence intervals, which are evaluated over seven different smartphone models through the trace-driven simulations. For the evaluation of each model, we only used its own data traces in Table 1. VehicleSense on Samsung Galaxy Note 2 and LG Optimus G Pro outperform the implementations on other models, but the gap is admittedly small. Also, regardless of the smartphone models, the average precision, recall and F_1 score achieve more than 98.5%. This confirms that unless the microphone of a smartphone gets a complete overhaul, VehicleSense is virtually universally applicable and there is almost

no need for per-device adaptation.

7.4 Performance Comparison with Existing Systems

We now compare the system performance of VehicleSense with that of the state-of-the-art accelerometer based method proposed in [17] in which peaks of accelerometers from acceleration and deceleration are carefully processed and characterized to recognize the type of a vehicle. We implemented the system of [17] in MATLAB as a trace-driven simulator for the accuracy comparison and also implemented it on an Android platform, Samsung Galaxy S6 running Android 5.1.1 Lollipop. for the comparison of the latency and the power consumption.

The recognition accuracy

We summarize the recognition accuracy results in Table 6, which are from VehicleSense and the Hemminki's system of two criteria, 4 and 10 accelerometer peaks. Note that using more peaks leads to a longer buffering of the accelerometer data and it is known to give better accuracy but with longer latency. As expected, the system with 10 peaks, showing 78.55% of average accuracy, outperforms that with 4 peaks in all accuracy measures. However even with 10 peaks, it is evident that it is outperformed by VehicleSense whose system-wide accuracy reaches to 98.2%.

The recognition latency

VehicleSense typically recognizes if a person is on a vehicle by going through the sound classifier followed by the two triggers. The Hemminki's system also has a trigger that senses the level of kinetic energy and invokes the main accelerometer processing module when the energy level is detected high. Given these hierarchical designs with different numbers of triggers, we measure the actual recognition latency from both systems running on the same Android platform. Our measurement on the recognition latency is done over simple scenarios of getting on and getting off a bus. Our measurement reveals that the total latency of VehicleSense keeps staying in the level of 0.6 seconds while that of [17] ranges from 80 to 140 seconds with 4 peaks, which is much slower than VehicleSense. Note that this system-wide latency evaluation is made right after the moment that a person becomes stationary in a vehicle for the fair comparison. About 2 seconds on average needs to be added in the latency to take the short roaming after getting on a vehicle into consideration.

The power consumption

Using the Android implementations of both systems, we measure the power consumptions of the triggers and the main classifiers. Then, we apply these results to 110 user traces to quantify how much of sensing events will occur during a day, and thus to estimate the average power consumption of both systems. Figure 10 summarizes the individual daily power consumption of both systems. On average, VehicleSense that takes 26.11 mW is far more energy efficient than the Hemminki's system taking 86.51 mW on average. This result makes sense given the extremely short sampling duration of sound compared to the monitoring of accelerometers for a much longer duration.

	LG Optimus G ProS [21]	LG Vu3 [22]	LG G2 [23]	SAMSUNG Galaxy S3 [24]	SAMSUNG Galaxy Note2 [25]	PANTECH Vega No.6 [28]	PANTECH Vega LTE-A [29]
stationary	140	120	120	160	160	140	160
walking	440	420	150	380	380	360	300
bus	440	300	140	960	460	220	180
taxi	260	400	180	620	340	300	220
subway	400	400	90	780	320	260	280

Table 1: Experiment time(minute) of each transportation by smartphone model

	Precision (%)	Recall (%)	F_1
On a vehicle	89.03	99.37	93.91

Table 2: Precision, recall, and score of our transportation trigger.

	MT	TT	SC
Power consumption (mW)	5.80	178.30	270.92
Activated duration (s)	Persistent	1.02	0.54

Table 3: The power consumption of MT, TT, and SC modules implemented on Samsung Galaxy S6 running Android 5.1.1 Lollipop. Their activated durations for each call are also presented in seconds.

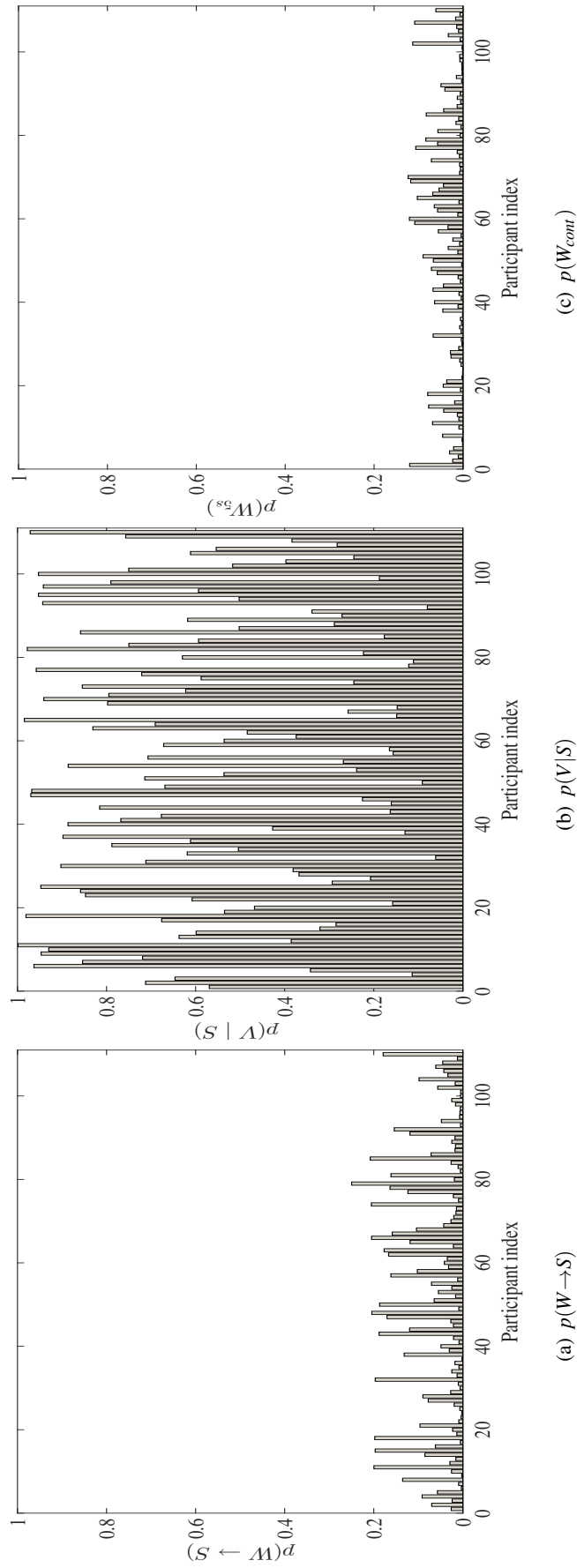


Figure 8: $p(W \rightarrow S)$, $p(V|S)$, and $p(W_{cont})$ which are evaluated individually from the 110 user traces measured in the wild.

Filters	Precision (%)				Recall (%)				F_1			
	Bus	Taxi	Subway	Non-vehicle	Bus	Taxi	Subway	Non-vehicle	Bus	Taxi	Subway	Non-vehicle
Inverse	86.29	85.24	89.24	98.61	86.40	92.21	78.58	97.03	86.34	89.04	82.69	97.81
Linear	92.49	93.35	93.93	99.03	93.22	93.55	90.04	97.57	92.85	93.45	91.95	98.29
Mel	92.44	95.40	96.91	99.55	92.19	95.72	96.75	99.9	92.32	95.56	96.83	99.77
Doubled-Mel	98.29	98.32	98.42	99.85	97.81	98.53	98.18	99.81	98.05	98.42	98.30	99.83
Steep	99.42	99.61	99.63	99.77	98.67	99.27	99.24	99.35	99.04	99.44	99.43	99.56
Sheer	99.31	99.47	99.53	99.87	98.64	99.13	99.27	99.97	98.97	99.30	99.40	99.89
Extreme	96.47	97.09	97.80	99.08	96.61	99.11	98.31	99.93	96.03	97.59	97.56	99.51

Table 4: Precision, recall and F_1 score by the seven non-linear filters.

Time(s)	Precision (%)				Recall (%)				F_1			
	Bus	Taxi	Subway	Non-vehicle	Bus	Taxi	Subway	Non-vehicle	Bus	Taxi	Subway	Non-vehicle
	0.1	97.63	97.47	98.04	99.30	97.97	96.72	98.26	99.99	97.80	97.09	98.15
0.3	99.38	99.42	99.54	99.77	98.16	99.18	98.65	99.24	98.77	99.30	99.09	99.50
0.5	99.42	99.61	99.63	99.77	98.67	99.27	99.24	99.35	99.04	99.44	99.43	99.56
1	99.70	99.58	99.74	99.82	98.69	99.42	99.26	99.37	99.19	99.50	99.50	99.59
5	99.37	99.93	99.93	99.82	99.27	99.05	99.64	99.36	99.32	99.49	99.78	99.59
0.5 (PP)	99.22	99.62	99.59	99.60	99.87	99.12	99.73	99.90	99.41	93.37	99.66	99.94

Table 5: Precision, recall and F_1 score by the sampling durations. One result from the privacy protection mode (PP) is also included in the bottom.

Systems	Precision (%)				Recall (%)				F_1			
	Bus	Taxi	Subway	Mean	Bus	Taxi	Subway	Mean	Bus	Taxi	Subway	Mean
VehicleSense	98.13	98.20	98.49	98.20	97.71	98.71	98.44	98.28	97.96	98.46	98.43	98.28
Hemminki's (4peaks)	79.10	75.32	73.55	75.99	82.04	78.26	79.80	80.03	78.04	84.39	57.52	73.31
Hemminki's (10peaks)	81.83	86.63	63.51	77.32	78.56	79.59	64.55	74.24	81.93	82.23	70.73	78.30

Table 6: The system-wide precision, recall and F_1 score of VehicleSense and Hemminki's system.

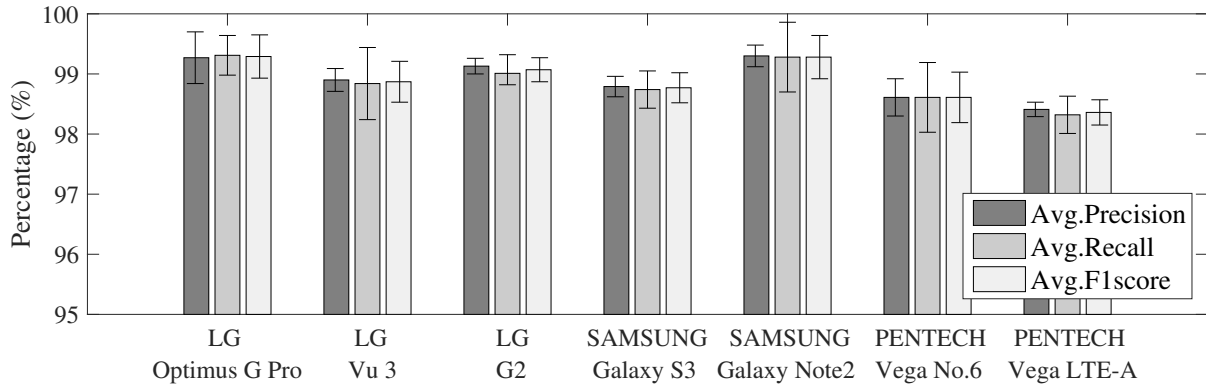


Figure 9: Precision and recall by smartphone devices.

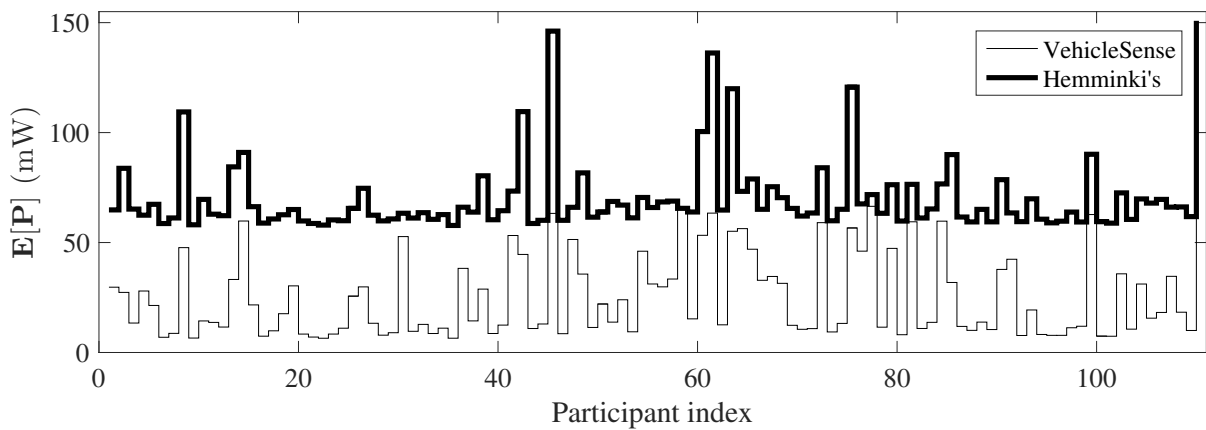


Figure 10: $\mathbb{E}[\mathcal{P}]$ of VehicleSense and Hemminki's which are evaluated individually from the 110 user traces measured in the wild.

8 Concluding Remarks

In this work, we suggested and implemented a reliable sound-based transportation mode recognition system, VehicleSense and provided extensive analyses on its performance over 183 hours of real smartphone log data. We find that VehicleSense achieves over 98.2% of recognition accuracy with about 0.6 seconds of system latency and consumes only 26.1 mW on average. We verified that this promising performance is reliably achievable in various smartphone models. We also demonstrated that VehicleSense can be privacy safe when it works with a privacy protection mode that prevents VehicleSense from capturing any conversational information while achieving almost the same level performance.

References

- [1] ASHISH SHARMA, VISHNU NAVDA, R. R. V. P. E. B. Cool-Tether: Energy efficient on-the-fly WiFi hot-spots using mobile phones. In *Proc. of ACM CoNext* (Dec 2009).
- [2] F. ZHENG, G. Z., AND SONG, Z. Comparison of different implementations of mfcc. *Journal of Computer Science and Technology* 16 (2001), 582–589.
- [3] HURWITZ, S. Say goodbye to the couch potato: Work and the commute are new prime viewing locations. Rovi Corporation, June 2016.
- [4] JOHN SHAWE-TAYLOR, N. C. *Support Vector Machines*. Cambridge University Press, 2000.
- [5] JOOHYUN LEE, KYUNGHAN LEE, Y. K. S. C. PhonePool: On energy-efficient mobile network collaboration with provider aggregation. In *Proc. of IEEE SECON* (June 2014).
- [6] KARTIK SANKARAN, MINHUI ZHU, X. F. G. A. L. A. M. C. C. L.-S. P. Using mobile phone barometer for low-power transportation context detection. In *Proc. of ACM Sensys* (November 2014).
- [7] LEON STENNETH, OURI WOLFSON, P. S. Y. B. X. Transportation mode detection using mobile phones and gis information. In *Proc. of ACM SIGSPATIAL GIS* (2011).
- [8] MANHYUNG HAN, LA THE VINH, Y.-K. L. S. L. Comprehensive context recognizer based on multimodal sensors in a smartphone. *Sensors* 12 (Sep 2012), 12588–12605.
- [9] MATHISAS SUNDHOLM, JINGYUAN CHENG, B. Z.-A. S. P. L. Smart-mat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix. In *Proc. of UbiComp* (September 2014).
- [10] Monsoon power monitor. <https://www.msoon.com/LabEquipment/PowerMonitor/>.
- [11] N. DING, D. WAGNER, X. C.-Y. C. H. A. Characterizing and modeling the impact of wireless signal strength on smartphone battery drain. In *Proc. of ACM SIGMETRICS* (2013).
- [12] OZLEM DURMAZ INCEL, MUSTAFA KOSE, C. E. A review and taxonomy of activity recognition on mobile phones. *BioNanoScience* 3 (May 2013).
- [13] PENGFEI ZHOU, YUANQING ZHENG, M. L. How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing. In *Proc. of ACM MobiSys* (June 2012).

- [14] PETER H. VELTINK, MEMBER, I. H. B. J. B. W. D. V. W. L. J. M. R. C. V. L. Detection of static and dynamic activities using uniaxial accelerometers. *IEEE TRANSACTIONS ON REHABILITATION ENGINEERING* 4, 4 (Dec 1996).
- [15] POWERS, D. M. W. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies* 2 (2011), 37–63.
- [16] RAHUL C. SHAH, CHIEH-YIH WAN, H. L. L. N. Classifying the mode of transportation on mobile phones using GIS information. In *Proc. of UbiComp* (September 2014).
- [17] SAMULI HEMMINKI, PETTERI NURMI, S. T. Accelerometer-based transportation mode detection on smartphones. In *Proc. of ACM SenSys* (2013).
- [18] SASANK REDDY, MIN MUN, J. B. D. E. M. H. M. S. Using mobile phones to determine transportation modes. *ACM Transportations on Sensor Networks* 6, 2 (Feb 2010), 526–537.
- [19] SHUANGQUAN WANG, CANFENG CHEN, J. M. Accelerometer based transportation mode recognition on mobile phones. In *Proc. of APWCS* (2010).
- [20] Average time spent per day with major media by US adults,2011-2017. eMarketer, Oct 2015.
- [21] Full phone specification of LG Optimus G Pro. http://www.gsmarena.com/lg_optimus_g_pro_e985-5254.php.
- [22] Full phone specification of LG Vu3. http://www.gsmarena.com/lg_vu_3_f3001-5723.php.
- [23] Full phone specification of LG G2. http://www.gsmarena.com/lg_g2-5543.php.
- [24] Full phone specification of SAMSUNG Galaxy S3. http://www.gsmarena.com/samsung_i9300_galaxy_s_iii-4238.php.
- [25] Full phone specification of SAMSUNG Galaxy Note 2. http://www.gsmarena.com/samsung_galaxy_note_ii_n7100-4854.php.
- [26] Full phone specification of SAMSUNG Galaxy S6. http://www.gsmarena.com/samsung_galaxy_s6-6849.php.
- [27] Full phone specification of SAMSUNG Galaxy S7. http://www.gsmarena.com/samsung_galaxy_s7-7821.php.
- [28] Full phone specification of PANTECH Vega No.6. http://www.gsmarena.com/pantech_vega_no_6-5268.php.

- [29] Full phone specification of PANTECH Vega LTE-A. http://pdadb.net/index.php?m=specs&id=4817&c=pantech_vega_lte-a_im-a880s_32gb.
- [30] Statistics brief world metro figures. Union Internationale des Transports Publics (UITP), Oct 2014.
- [31] TAO FENG, JUN YANG, Z. Y. E. M. T. W. S. Tips: Context-aware implicit user identification using touch screen in uncontrolled environments. In *Proc. of ACM HotMobile* (2014).
- [32] TROST SG1, LOPRINZI PD, M. R. P. K. Comparison of accelerometer cut points for predicting activity intensity in youth. *Medicine and Science in Sports and Exercise* 43, 7 (Jul 2011).
- [33] V. R. VIJAYKUMAR, P. T. VANATHI, P. K. Modified adaptive filtering algorithm for noise cancellation in speech signals. *Elektronika ir Elektrotechika* 74 (2007), 17–20.
- [34] Spectrogram in wikipedia. <https://en.wikipedia.org/wiki/Spectrogram>.
- [35] XINHUI ZHOU, DANIEL GARCIA-ROMERO, R. D. C. E.-W. S. S. Linear versus mel frequency cepstral coefficients for speaker recognition. In *Proc. of IEEE ASRU* (Dec 2011), pp. 559–564.
- [36] YOUNG-SEOL LEE, S.-B. C. Activity recognition using hierarchical hidden markov models on a smartphone with 3d accelerometer. In *Proc. of HAIS* (May 2011), vol. 6678.