Crowdsourcing Undersampled Vehicular Sensor Data for Pothole Detection

Andrew Fox, B.V.K. Vijaya Kumar
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA, USA, 15213
{afox1, kumar}@ece.cmu.edu

Jinzhu Chen, Fan Bai
General Motors Corporation
Warren, MI, USA, 48090
{jinzhu.chen, fan.bai}@gm.com

***Abstract*—The increased availability of embedded vehicle sensors allows for the detection of road features such as potholes. Despite being a promising approach, current vehicle embedded sensors operate at low frequencies and undersample sensor signals, thus degrading detection accuracy. One emerging solution is to crowdsource such undersampled sensor data from multiple vehicles to increase the detection accuracy. Aggregating sensor data from multiple vehicles, nonetheless, is a challenging task given the heterogeneity among vehicles, asynchronous sensor operation, GPS error, and sensor noise. Additionally, there may be bandwidth restrictions in vehicular networks which limit the amount of data available for aggregation. We investigate these issues by focusing on the problem of pothole detection. To quantify the detection accuracies and effects of real-world limitations, we design and evaluate three crowdsourcing pothole detection schemes involving vehicles and the Cloud. We also address the issue of lack of extensive model training data by demonstrating that a detection model applicable to real-world systems can be derived using simulated data. We validate our pothole detection methods using 38.1 km of real-world data collected from driving on roads in Warren, Michigan.**

I. Introduction

The increased use of embedded sensors and dedicated communications systems in consumer vehicles provides a resource for gathering current and wide-ranging road environmental information. This information can subsequently be used to generate road maps rich with detailed, vehicle-specific information. There is particular interest in detecting the presence or absence of road features such as potholes, and in this paper, we focus on pothole detection using vehicle accelerometer signals. Potholes damage wheels, suspension systems, vehicle frames, and potentially injure drivers and passengers. They are responsible for millions of dollars in insurance claims and roadway repairs, and drivers would benefit from knowledge of pothole locations in planning their routes.

In this work, we advocate utilizing vehicle embedded accelerometer sensors for pothole detection. Compared to smartphone sensors, embedded sensors are optimally located in a fixed position in the vehicle, and their orientation is calibrated by vehicle manufacturers. Raw sensor data can therefore be directly used without the necessary preprocessing to map data between the phone and vehicle reference frames. Resulting signals better represent the vehicle dynamics rather than the dynamics of a decoupled object within the vehicle. Embedded sensors provide a systematic basis to gather and report data by having uniform implementations across all vehicles. The sensors also more easily allow for a common infrastructure for sharing data as protocols can be universally implemented.

978-1-4673-7331-9/15/$31.00 © 2015 IEEE

Despite these advantages, using embedded vehicle sensors introduces a series of challenges. The sensors are designed for a low-power environment with limited buffer and storage capabilities. These practical constraints limit the embedded sensors to a low operating frequency and restrict the in-vehicle data processing capabilities. The low operating frequencies undersample the signals of interest, resulting in low pothole detection rates. For example, a vehicle traveling at 60 km/h making measurements at 1 Hz has only a 6% chance of taking a measurement within a 1 m long pothole. Detection using a single vehicle is therefore unreliable.

One solution to increase the detection probability is to crowdsource information by aggregating sensor data from multiple vehicles. This approach increases the effective sampling rate and makes detection less subject to noise and imperfections from a single vehicle. However, this crowdsourcing approach introduces new challenges.

First, communication is difficult as vehicular networks are inconsistent and constrained by limited connectivity time [1] and high packet loss rate [2]. Second, pothole detection is complicated by the differences in the mappings from the potholes to the measured accelerometer signals in different vehicles. Vehicles differ in size, weight, length, as well as the quality of suspension system. These differences affect how vehicle accelerometers respond to potholes. Similarly, not all potholes are identical. The position of the pothole on the road, the curvature of the road, and the length and depth of the pothole affect the accelerometer signals. Different vehicles driving over a single pothole would not produce identical signals in these varying situations.

To further complicate the problem, different vehicles also drive over potholes at different speeds, which not only affects the accelerometer response within a single vehicle, but also changes the spatial sampling frequency of measurements among vehicles. Since vehicle sensors operate asynchronously, the geographic distribution of sampling locations could therefore be undesirably scattered. Finally, GPS position error complicates the data aggregation since samples may be reordered.

In this work we demonstrate how such undersampled, distorted, and heterogeneous vehicle sensor signals can be crowdsourced and aggregated in vehicular networks (with limited bandwidth) to improve the pothole detection rate. To tackle varying practical scenarios, we aim to design a generic detection system architecture that performs well without tuning for specific scenarios. We therefore present and compare a number of candidate approaches using the following two variations of crowdsourced detection systems:

1) *Vehicle Binary Detection to Cloud Binary Detection* Individual vehicles each make binary decisions and these results are aggregated on the Cloud for final binary pothole detection.
2) *Vehicle Raw Data to Cloud Binary Detection* The raw signal data from all vehicles is aggregated on the Cloud for final binary pothole detection.

Another challenge in developing pothole detection systems is that designing these detection models requires a vast amount of training data encompassing the many different vehicles, driving patterns, and pothole situations that one may encounter on the road. This type of data is difficult to obtain empirically. We propose to use a simulator to generate training data so that training signals representing a variety of driving conditions and scenarios can be obtained. We demonstrate that detection methods designed from simulated data work reasonably well when directly applied to real-world data.

The key contributions of this work are the following:

1) We demonstrate how crowdsourced data from multiple vehicles can be used to increase the pothole detection accuracy in a generic framework, when using low-frequency embedded accelerometers operating asynchronously and with heterogeneous vehicle behavior.
2) We examine the tradeoff between the number of vehicles participating in crowdsourcing and detection accuracy, while analyzing how vehicle sensor restrictions and bandwidth constraints affect event detection. We further present a detection system to handle these constraints.
3) We show that pothole detection methods developed using simulated data can be applied successfully to real-world data.

The remainder of this paper is organized as follows: Related work is presented in Section II. Section III gives an overview of the simulated and real-world data. The pothole classifiers are discussed in Section IV. These classifiers are incorporated into three detection schemes in Section V, and Section VI presents results and analysis for those schemes. The derived models are directly applied to real-world data in Section VII. Conclusions are given in Section VIII.

II. Related Work

Crowdsourcing information has been investigated in prior works for multiple vehicular purposes. Examples include locating available road-side parking spots [3], estimating travel time [4], identifying anomalous traffic states from GPS trace data [5], or generating digital maps from vehicle GPS traces and distinguishing between one-way and two-way roads [6]. Due to the type of information considered, these scenarios lead to similar signals (excluding noise) from each vehicle for a single event instead of the heterogeneous scenarios considered in our work since the mapping from their observed signals to the detected feature is identical for all vehicles.

Methods of aggregating heterogeneous or conflicting road information have been investigated. The work in [7] uses a Time-Decay Sequential Hypothesis Testing algorithm to aggregate and disseminate road information to and from multiple vehicles. However at low sampling rates this relies on aggregating decisions from weak classifiers instead of using the collective raw data to strengthen the classifier.

Pothole detection has been the focus of some previous works, however the sensor device has generally been a smartphone instead of embedded vehicle sensors. Smartphones generally have significantly more computational capacity than what is available within the vehicle. This allows for a much higher sensor operating frequency (300+ Hz) than the embedded vehicle sensors, which enables a detection system to measure the full dynamic motion of the vehicle caused by the pothole. This is important since when we transition to using low frequency acceleration signals we lose important signal properties. Undersampling has been shown to be problematic for other applications such as vehicle speed estimation [8]. We use embedded sensors since they are standardized across vehicles types and are integrated within the controls and communications systems of the vehicle. This makes them a more logical resource as the detection system can be integrated into the existing framework of the vehicle and Cloud system.

The Pothole Patrol system [9] uses speed, high-pass, and vertical and lateral acceleration filters to identify potholes in signals from Boston taxis. Road bumps are detected in [10] by examining the peak vertical acceleration and the duration for which the acceleration dips below a heuristically defined threshold. Gaussian Mixture Models are used in [11] on aggregated data to determine potholes from 100 taxis in Shenzhen, China by examining z-scores of the features from [9]. A linear model for speed using 38 Hz sensors was constructed to try to eliminate the speed dependence of acceleration in [12], however the acceleration can deviate significantly from the linear model. Since these works use non-embedded high-frequency sensors, they rely on only one vehicle for detection. They therefore ignore problems resulting from GPS error.

Our proposed system relies on accelerometer data and is therefore dependent on a few initial vehicles being unable to avoid the potholes. The information obtained from those vehicles can be shared with subsequent vehicles for their benefit. Instead of using accelerometers, images from vehicle-mounted cameras could be used to pre-detect potholes, however the required angle of the camera and subsequent processing time may not provide the driver the opportunity to avoid the pothole. The work in [13] looks for pothole regions in images via segmentation methods employing shape-based thresholding, and examining geometric and texture based properties of suspect pothole regions. Similarly, large simulated potholes are found in images in [14] by looking for large circular

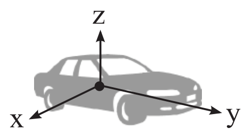

Fig. 1. Coordinate system in relation to the vehicle

objects with a predetermined brightness difference. Unlike accelerometer-based detection methods, image-based methods require specific lighting conditions to function properly and may be unusable at night or in poor weather conditions.

III. Pothole Data

To reflect the reality of heterogeneous signals and to create an extensive model capturing a variety of circumstances and changing variables, we require data from different vehicles driving over potholes in diverse scenarios along with ground-truth pothole locations. Obtaining such data through real-world driving is labor-intensive and very expensive. To address this problem, we propose using a simulator to synthesize training and testing data. With simulations, we are able to extend the experiments beyond the available equipment, constrain the environment to only the variables of relevance, and drive distances that would be too costly to drive manually. We train our detection models using the simulated data and then evaluate the performance of detectors with real-world data detailed in the following subsections. Variables are defined in reference to the vehicle coordinate system illustrated in Fig. 1.

A. Simulated Data

We use the CarSim® [15] program to simulate vehicles driving over potholes. CarSim® is a highly customizable vehicle simulation kit, which accurately simulates vehicle component (e.g. tires, suspension, steering, etc.) responses to given input environments and stimuli. It is the default tool for kinematic and controls simulation testing in the vehicle community [16]. With CarSim®, we simulated vehicles driving a total of 3800 km over 35478 potholes. These totals vastly exceed what can be collected manually and gives us a basis for creating models to subsequently apply to real-world data.

We simulated 4 different courses, each driven over at a constant speed from 5 km/h-150 km/h in 1 km/h increments to test the effects of a large range of vehicle velocities. This was repeated for standard Sedan, Minivan, and SUV car models to gauge responses from different vehicle types. A description of each road course is given in Table I. The general road roughness profile was constructed using the 3D Weierstrass-Mandelbrot (W-M) function [17] to represent a paved asphalt road. The W-M function has been shown to well approximate road roughness and its effect on vehicle dynamics [18], [19]. Potholes were simulated by adding a constant dip of the respective pothole size to the roughness profile on only one side of the vehicle for the length of the pothole. The non-pothole features of these courses were meant to induce vehicle acceleration responses unrelated to potholes, so the models could be trained to avoid false positives, for example from a large lateral acceleration while turning.

TABLE I
Description of simulated road courses.

Course	Road Style	Road Features (100 m spacing)
1	0.4 km long Straight, Flat	100 cm long, 8 cm deep pothole 40 cm long, 3.5 cm tall speedbump 120 cm long, 3.5 cm tall speedbump
2	3.1 km long Straight, Flat	30 potholes 2 cm - 10 cm deep 20 cm - 120 cm long
3	3.1 km long Straight, 5° incline	30 potholes 2 cm - 10 cm deep 20 cm - 120 cm long
4	3.1 km long Sinusoidally curving, Flat	30 potholes 2 cm - 10 cm deep 20 cm - 120 cm long

We recreated two road types by appropriately selecting runs from the simulated data set. The first road is a city road where the vehicles traveled at an average speed of 50 km/h with a 15 km/h standard deviation. The second road is a highway road where the vehicles traveled at an average speed of 100 km/h with a 20 km/h standard deviation.

The data for each simulation was output at 100 Hz. The signals were downsampled with random starting points to simulate data acquisition at lower measurement frequencies. We collected the following variables:

- Distance traveled, y
- Vehicle speed, v
- Lateral, longitudinal, and vertical accelerations, (a_x, a_y, a_z) respectively, at the vehicle center of mass.

We also simulated the presence of GPS position error by adding to each vehicle's position measurements additive white Gaussian noise of standard deviation of 4 m. This GPS position error is consistent with observations of real-world data [9].

B. Real-world Data

The real-world data set was collected by driving on the city roads around the General Motors campus in Warren, MI. We implemented a vehicle sensor recording system that logs the GPS position, vehicle speed, and the three-axis acceleration. The data set consists of 38.1 km of city roads containing 268 manually identified potholes from examining video from a recorder equipped on the testing vehicle. The vehicle was driven at speeds up to 75 km/h. The GPS position data was output at 1 Hz, speed at 10 Hz, magnitude of the vertical acceleration at 2 Hz, and lateral acceleration at 150 Hz. The locations used in the pothole detector are the reported GPS locations. GPS position error is accounted for by training the detector with the noisy simulated data previously described.

IV. Pothole Detector

Using the simulated data, we examine the effects that heterogeneous vehicles, potholes, and driving patterns have on the accelerometer signals. This is important for determining discriminating features for use in our detection models and to design the models to be applicable under general conditions without tuning for specific situations.

A. Accelerometer Signal Properties

We first analyze the effect that vehicle velocity has on the vertical acceleration measurements (excluding gravity). As an example, the 100 Hz vertical acceleration signals for the three different vehicle types driving over a 40 cm long, 4 cm deep pothole are shown in Fig. 2a and Fig. 2b for vehicles traveling at 25 km/h and 100 km/h, respectively. The vehicle's front and back tires each individually produce significant vertical acceleration peaks when driving over the pothole, with the distance between them dependent on the length of the vehicle. Since the temporal output frequency is the same between the runs, in a given distance there is less data at higher speeds since the traveling time is reduced. However, the response of the suspension system differs beyond just the spatial sampling rate, since the impulses corresponding to the entering and the exiting of the pothole are closer in time at higher speeds. Also, the aftershocks are larger at a further distance from the pothole at higher speeds due to the vertical acceleration decay being a function of time, not distance. This emphasizes the importance of including velocity as a component in the detection models as it affects the acceleration signal patterns.

The initial peaks from driving over the potholes are potential example candidate features for pothole detection. We therefore examine the upper and lower bounds of the vertical acceleration as a function of velocity. The results are shown in Fig. 3 for normal road and pothole regions. When driving over a pothole, the upper bound on the acceleration increases until about 30 km/h and then decreases. The lower bound is approximately constant at all velocities. The bounds on the vertical acceleration on the normal road however symmetrically increase as velocity increases. This opposing behavior between the normal road and pothole regions increases the difficulty of detecting potholes at high speeds, particularly when there is less available data. The bounds are also proportional to the size of the pothole and the vehicle center of mass height.

We perform a similar investigation with the lateral acceleration. A lateral acceleration response distinguishes potholes from features which span the width of the road, such as speedbumps or expansion joints. The upper and lower bounds on the lateral acceleration as a function of velocity behave similarly to the vertical acceleration when driving over a pothole. There is an increase in the magnitude of the bound over the pothole until about 30 km/h followed by a decrease.

The bounds and spread of the acceleration points, as well as their relation to velocity are therefore important discriminative features which could be used to detect the potholes. The algorithm for selecting which of these features has the greatest discriminative ability is presented in the following section.

B. Classifier Training

We design multiple detection schemes, described in Section V, that use aggregated data from multiple vehicles for pothole detection on the Cloud. Support Vector Machines (SVMs) [20] with radial basis kernel functions serve as our pothole classifiers, constructed using the LIBSVM [21] package. SVMs are discriminative classifiers that identify the

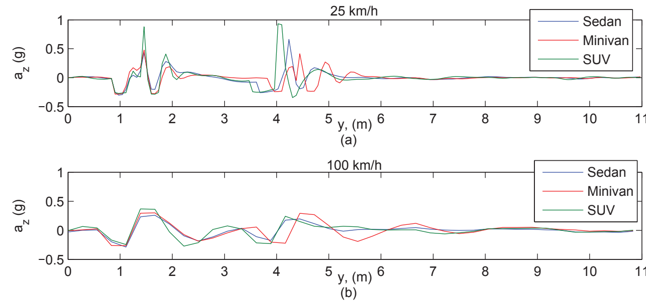

Fig. 2. Vertical acceleration signals from vehicles driving over a 40 cm long, 4 cm deep pothole at (a) 25 km/h and (b) 100 km/h.

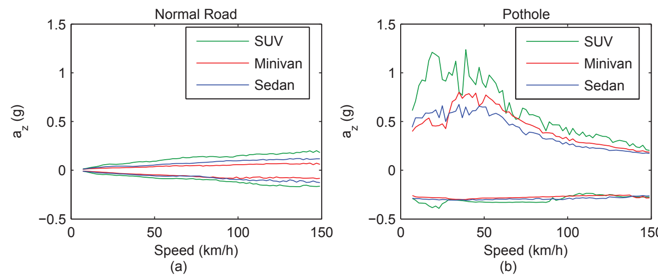

Fig. 3. Upper and lower bounds of the vehicle vertical acceleration as a function of speed on (a) normal road, and (b) 40 cm long, 4 cm deep pothole.

boundary between two classes of data by maximizing the gap between the two data classes. The SVMs discriminate between the two binary event classes (namely, pothole regions and non-pothole regions) based on features obtained from the raw signal data. We use radial basis kernel functions [20] to create nonlinear boundaries between the data classes, which is important given the nonlinear effects of velocity on the acceleration signals.

We use a sliding window scheme (10 m windows, sliding by 1 m) to group the data. The 10 m window size is chosen as it can envelop the length of the pothole region, vehicle length, and GPS position error. The sliding window structure counts data in multiple windows. Since a pothole location is unknown and needs to be estimated, this ensures that there exists some window that overlaps with the pothole.

For training the classifiers, we begin with a list of 120 candidate features for each window based on our observations of the accelerometer signal data properties. These features are based on the vertical and lateral accelerations as well as their ratios and products with the vehicle velocity. The maximum, mean, and standard deviation of the raw data and the absolute value of the raw data in each window, as well as correlation coefficients between the different acceleration components are taken as candidate features. These candidate features span a large space and provide a basis to select the most discriminating features. We also train a classifier to run on each individual vehicle for certain variations of our detection architecture. On a single vehicle we could also use the difference between consecutive measurement values as a candidate feature, however this is only useful at very low

Algorithm 1 Greedy forward feature selection algorithm for training SVM

```
i ← 0
score(0) ← −∞
selectedFeatures ← {}
candidateFeatures ← {allFeatures}
do
    i ← i + 1
    bestScore ← −∞
    bestScoreIndex ← 0
    for j : candidateFeatures do
        [β, α] ←trainAndTestSVM(selectedFeatures,
                                 candidateFeatures(j))
        s(j) ← β − α
        if s(j) > bestScore then
            bestScore ← s(j)
            bestScoreIndex ← j
        end if
    end for
    selectedFeatures.add(candidateFeatures(bestScoreIndex))
    candidateFeatures.remove(bestScoreIndex)
    score(i) ← s(bestScoreIndex)
while (score(i) − score(i − 1) > scoreThreshold)
selectedFeatures.remove(i)
finalSVM ←trainSVM(selectedFeatures)
```

speeds due to the embedded sensor low-sampling rate. For example, a vehicle traveling at 60 km/h with sensors operating at 1 Hz would expect only 0.6 data points on average in a 10 m window. The difference in consecutive values would therefore not always be meaningful.

Instead of manually selecting features as in previous works, we use a greedy forward feature selection algorithm [22] to determine which candidate features are best suited for classification. Pseudocode for this feature selection algorithm is shown as Algorithm 1. Using this process, an SVM is first created with each individual candidate feature. A score function is computed for each SVM based on the true detection rate, β, and the false alarm rate, α. If P is the number of windows that actually contain a pothole, N is the number of windows that do not actually contain a pothole, TP is the number of windows in which a pothole was correctly detected, and FP is the number of windows in which a pothole was mistakenly detected, then $\beta = \frac{TP}{P}$ and $\alpha = \frac{FP}{N}$. For each round, whichever candidate feature gives the highest score value, s, where

$$s = \beta - \alpha, \tag{1}$$

is then added as a selected feature. This process is then repeated by training new SVMs with the selected feature list in addition to each of the remaining candidate features individually. We stop adding candidate features to the selected feature list when the difference in score values for the test data between consecutive rounds falls below a threshold.

A pothole is considered to be detected if there exist positive detections in any of the windows that overlap with the pothole. Positive detections in windows following the pothole may be due to large aftershocks and are therefore not considered false alarms provided they follow a string of consecutive positive detections in windows that overlap with the pothole.

TABLE II

SELECTED SVM FEATURE LIST FOR SAMPLE VEHICLE NUMBERS ON AGGREGATED DATA AT AN AVERAGE VEHICLE SPEED OF 50 KM/H.

1 vehicle	10 vehicles	20 vehicles	75 vehicles	200 vehicles	500 vehicles
mean$(\frac{a_z}{v})$	std(a_z)	max$\lvert\frac{a_z}{v}\rvert$	std$(a_z v)$	mean$\lvert a_z\rvert$	mean$\lvert a_z\rvert$
max$\lvert a_z\rvert$	max$\lvert a_z\rvert$	max$\lvert a_z\rvert$	mean$\lvert a_z\rvert$	mean$\lvert\frac{a_z}{v}\rvert$	std$(a_z v)$
mean$\lvert a_z\rvert$	max$\lvert\frac{a_z}{v}\rvert$	std$\lvert a_z\rvert$	mean$\lvert\frac{a_z}{v}\rvert$	std$(a_z v)$	mean$\lvert\frac{a_z}{v}\rvert$
mean$\lvert\frac{a_z}{v}\rvert$	std$\lvert a_z\rvert$	max$\lvert a_x v\rvert$	std(a_z)	std(a_z)	max$\lvert a_x\rvert$
max$\lvert\frac{a_x}{v}\rvert$	std$(\frac{a_x}{v})$		max(a_z)	std(a_x)	std(a_x)
			max$\lvert a_x a_z\rvert$		

C. Aggregated Vehicle Data Variations

New SVMs are trained for different numbers of aggregated vehicles, as features have different discriminative qualities and classification boundaries depending on the model parameters. For example, it is highly probable that the maximum measured acceleration value in a window containing a pothole is close to the actual peak acceleration value when aggregating data from 1000 vehicles. However with only 10 vehicles uploading data, the measurement locations may be poorly clustered in the window and miss the peak location. A detection threshold for the maximum acceleration would therefore be lower for 10 aggregating vehicles than for 1000 since 10 vehicles are less likely to produce the peak value. The resulting feature list is shown in Table II for some example numbers of crowdsourced vehicles. The features are listed in the order they were selected by the greedy forward feature selection algorithm.

A function of the vertical acceleration is the most important feature for all tested numbers of aggregated vehicles, but both lateral and vertical acceleration features are always listed. The ratio of the acceleration to velocity is also highly ranked since, as was previously shown, the acceleration signals from both the pothole and normal road conditions are heavily influenced by velocity. The feature list and the associated boundaries therefore also change depending on the vehicle speeds.

V. DETECTION ARCHITECTURE

The detection objective is to use the raw sensor data collected from the vehicles to detect the binary presence or absence of potholes. Where the data is processed and how it is aggregated affects the detection capabilities and the computational and bandwidth requirements of the system. We investigate the following three architectures, shown in Fig. 4, categorized by the type of data being aggregated from the vehicles:

1) Vehicle Binary Detection to Cloud Binary Detection
    i) *Binary Voting Detection*
    With sensors operating at 1 Hz, classifiers within each vehicle estimate the binary presence or absence of a pothole. This binary data from multiple vehicles is aggregated in the Cloud to make the final detection decision.
2) Vehicle Raw Data to Cloud Binary Detection
    i) *Idealized Crowdsourced Detection*
    Operating the sensors at 1 Hz, all the raw data is

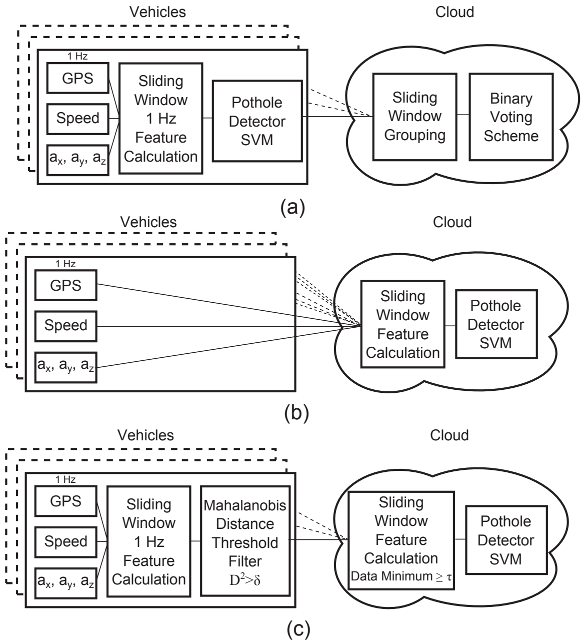

Fig. 4. Architecture of (a) Binary Voting Scheme, (b) Idealized Crowdsourced Detection, (c) Filtered Multi-stage Detection schemes

transmitted from each vehicle to the Cloud where potholes are detected using the aggregated raw data.

ii) *Filtered Multi-stage Detection*
The raw data is filtered in each vehicle to avoid transmitting data that corresponds to normal road conditions. This significantly reduces the required bandwidth as the vehicle hits potholes infrequently. Unlike the previous scheme, this requires some data processing in the vehicles.

The following subsections detail the differences between the different detection schemes and how the SVMs are applied.

A. Binary Voting Detection

The SVM for the 1 Hz sensor operating frequency derived for the single vehicle detection system is used directly in each individual vehicle. If there is a positive detection for a pothole in a given window, all data points inside that window are assigned the pothole binary value 1. The binary values with associated GPS locations are uploaded to the Cloud for aggregation.

In the Cloud, a second 10 m sliding window is used for voting. If within a window, at least a fraction, ϵ, of the aggregated data points vote that a pothole exists, then a pothole is determined to exist in that window. A threshold lower than majority vote must be used since the detection probability in each individual vehicle is low (<30%). The ϵ value should be chosen between zero and the average detection probability on each vehicle, depending on the desired tradeoff between the detection and false positive rates.

B. Idealized Crowdsourced Detection

The raw sensor data with associated GPS locations is directly uploaded from each vehicle to the Cloud for aggregation. SVMs trained for the aggregated data are then applied to a sliding window. This scheme requires greater bandwidth than the Binary Voting Detection scheme since all the data is uploaded, but also allows for a larger dynamic range of the data to be available to make a final detection decision.

C. Filtered Multi-stage Detection

The goal of the Filtered Multi-stage Detection system is to reduce the required bandwidth while maintaining the detection rates achieved by the previous schemes. The data collected in each window is filtered in each vehicle to determine its potential to discriminate potholes, and therefore if it should be transmitted to the Cloud for aggregation. Filtering is based on the Squared Mahalanobis distance, D^2, for each test vector $\mathbf{x}$ of the features from the given windows to the normal road data cluster, where,

$$D^2 = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}), \tag{2}$$

and $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance matrix of the cluster of the features for the normal road calculated for the simulated data. The $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ values are precomputed for normal roads and are initially shared with all the vehicles. When D^2 exceeds a threshold, δ, the features of the region are deemed significantly different from those of a normal road, implying that a pothole may have occurred, and the data is transmitted to the Cloud. Data in windows for which $D^2 \leq \delta$ are not uploaded to the Cloud, under the assumption that the data does not contain any discriminating information. This significantly reduces the required bandwidth as only data which deviates from normal road conditions are transmitted.

However, this system results in a large number of false alarms since in normal road regions, only noisy or corrupt measurements are transmitted to the Cloud. To mitigate these effects of missing baseline data from normal road regions, we disallow positive detections in windows that do not exceed a threshold, τ, for the total amount of aggregated data.

VI. Detection Results

A. Binary Voting Detection

The binary pothole decisions from the vehicles sampling at 1 Hz are aggregated from multiple vehicles in a voting scheme, as described in Section V-A. The detection rate and false alarm rate for the Binary Voting Detection scheme is shown in Fig. 5 for a varying number of transmitting vehicles for voting thresholds $\epsilon = 10\%$ and $\epsilon = 20\%$. For a single vehicle at 1 Hz, the signal is being significantly undersampled and the sparse set of samples does not sufficiently capture the vehicle dynamics, resulting in an unacceptably low detection rate. The Binary Voting Detection scheme is able to increase the detection rate to a satisfactory level for even just 10 aggregating vehicles at 50 km/h.

Unfortunately, with this many vehicles the false alarm rate spikes. For example, there are 0.22 false alarms per true pothole for 20 vehicles at 50 km/h with $\epsilon = 10\%$. This is because the classification boundary for the in-vehicle binary detector must be relatively close to the normal road data set

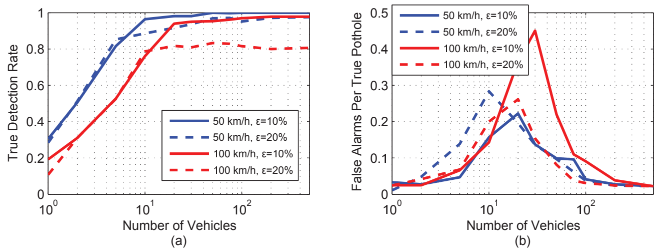

Fig. 5. (a) Detection rate, and (b) False alarm rate, for the Binary Voting Detection scheme for example ϵ thresholds

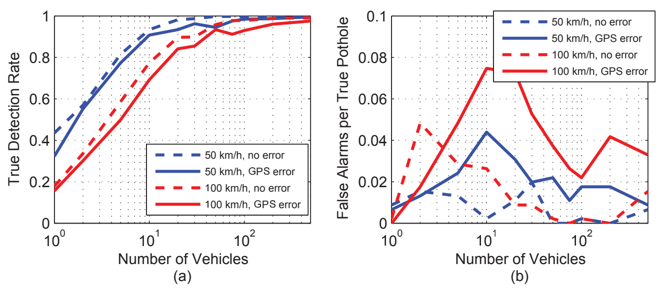

Fig. 6. Idealized Crowdsourced Detection System, (a) Detection rate, and (b) False alarm rate, with and without considering GPS error

since detection decisions must be made on windows containing very little data. This leads to excessive false alarms which could exceed the ϵ threshold when there are few vehicles involved in the voting. However as the number of vehicles continues to increase, a few noisy vehicles have less of an impact on the voting scheme and the number of false alarms per true pothole decreases. Increasing the ϵ voting threshold requires more vehicles to independently make a positive detection. This decreases the false alarm rate at the expense of also decreasing the true detection rate. With 500 vehicles we are able to achieve a detection rate of 1.00 and 0.022 false alarms per true pothole for the city road courses.

We avoid the false alarm problem from the weak in-vehicle classifiers by using the Idealized Crowdsourced Detection Scheme, where the classification boundary is pushed further from the normal road data cluster since more data is available in each window.

B. Idealized Crowdsourced Detection

The detection results for the Idealized Crowdsourced Detection scheme are shown in Fig. 6 for a varying number of aggregated vehicles, with and without including GPS error in the data set. At 50 km/h we are able to achieve 90% detection, even with GPS error, with only 10 vehicles. At 100 km/h we only need about 40 vehicles to achieve 90% detection. On typical roads it does not take long to collect data from 40 vehicles and the detection accuracy continues to increase with additional vehicles. The GPS error degrades the detection rate by up to 10%, decreasing as additional vehicles are added and the true detection rate approaches 1.

Similar to the Binary Voting Detection scheme, albeit at a much lower level, the false alarm rate increases from about 1 to 30 vehicles and then decreases when samples from

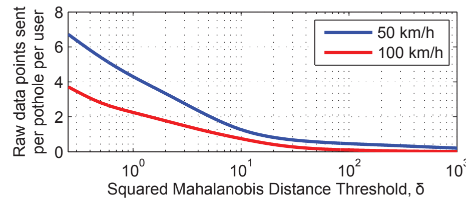

Fig. 7. Number of raw data points uploaded to the Cloud per vehicle per pothole as a function of the Squared Mahalanobis distance threshold from the normal road feature cluster.

additional vehicles are collected. With a small number of vehicles, there are very few pothole detections at all, true or false. As the number of vehicles increases, the detection rate and false alarm rate increase. However above 30 vehicles, we are more confident in the detection as more data is available, which results in the false alarm rate decrease. At 500 vehicles, the number of false alarms drops to 0.009 per true pothole. This outperforms the false alarm rate for the Binary Voting Detection scheme since that system relied on the weak detectors within each vehicle while the SVM for this Idealized Crowdsourced Detection scheme examines the full dynamic range of data collected from all the vehicles. GPS error is also responsible for false detections, especially with few vehicles as some of the potholes are detected, but in the wrong locations.

However this detection system requires excessive bandwidth. All the raw sensor data needs to be uploaded to the Cloud, when for the majority of the time, the vehicles are being driven over normal roads and producing non-discriminating data. The following section details the results from the Filtered Multi-stage Detection system designed to reduce the vehicular network bandwidth usage.

C. Filtered Multi-stage Detection

We use the two-stage system outlined in Section V-C. The Mahalanobis distance threshold in each vehicle determines how much of the raw data is sent to the Cloud, and therefore how much bandwidth is required. Fig. 7 shows the resulting number of data points for each vehicle per pothole which are uploaded as a function of the distance threshold, δ.

This two-stage process significantly decreases the amount of uploaded data from the previous Idealized Crowdsourced Detection system. As the threshold increases, first the normal road data is eliminated from transmission, followed by the less discriminating pothole data. These results are used to determine the minimum data amount threshold, τ, in the Cloud before pothole classification is considered. The largest benefit of adding this threshold, which we set to $\tau = 20\%$ of the expected data as determined from Fig. 7, is that the false alarm rate is significantly decreased since windows containing only a few noisy data points no longer result in positive detections.

The detection rate and false alarm rate for varying Mahalanobis thresholds, δ, and number of vehicles for the 50 km/h roads are shown in Fig. 8a and Fig. 8b respectively. There is a tradeoff in the detection rate from increasing the

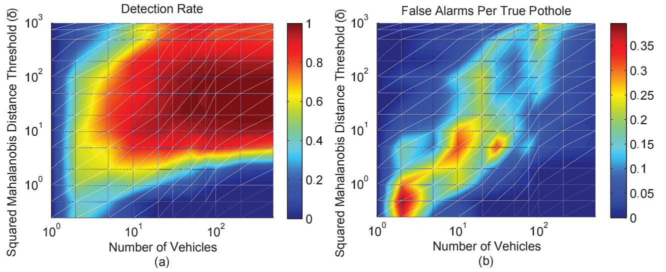

Fig. 8. (a) Pothole detection rate, and (b) False alarm rate per pothole, for the Filtered Multi-stage Detection scheme with vehicles traveling at 50 ± 15 km/h, avoiding detection in windows with less than $\tau = 20\%$ of the expected data per vehicle per pothole

distance threshold and therefore limiting the amount of data which is uploaded to the Cloud for aggregation. For low δ, the detection rate is limited due to τ being pushed artificially high based on the ratio of normal road regions to pothole regions. There is rarely enough data to initiate detection. For extremely high δ values, even the pothole data is excluded and the detection rate begins to fall again. There is an optimal threshold for which only the normal road data is excluded and the true detection rate is able to match that of the Idealized Crowdsourced Detection Scheme.

For the regions where the detection rate is at its highest, the false alarm rate is below the manageable 0.05 per true pothole. We observe similar result patterns for different vehicle speeds. These results show that the detection results of the previous crowdsourcing systems can be maintained while decreasing the bandwidth requirements of the vehicular network to only transmit pothole-related data.

D. Training Data

In order to create a generic pothole detection model applicable to diverse environments, we require an extensive data set for training. Generic models are important as tuning embedded sensors for different situations is extremely difficult and costly. To further demonstrate the importance of using extensive data, we trained new SVMs for the Idealized Crowdsourced Detection scheme using each of the road courses in Table I separately. These SVMs were used to test each of the other courses. The detection rate and false alarms per true pothole are shown in Table III and Table IV respectively for each training and test combination of the road courses with comparisons to training the SVM using all roads. These results are for vehicles traveling at an average of 50 km/h, using data aggregated from 100 vehicles.

Depending on how different the courses are, the classifiers trained on only one course are inaccurate when applied to other courses. The speedbumps on Course 1, for example, are continually detected as potholes by the other SVMs and the curves on Course 4 mask the potholes from the classifiers trained on the straight-line courses. The SVM boundaries are calculated to maximize the margin on the training course without any consideration as to their impact on features from other courses. The SVM trained on data from all the courses

TABLE III
TRUE DETECTION RATES FOR POTHOLE DETECTION ON DIFFERENT ROAD COURSES (SEE TABLE I) WHEN TRAINED ON INDIVIDUAL COURSES.

		Testing Course			
		1	2	3	4
Training Course	1	1.00	0.99	1.00	0.42
	2	1.00	0.99	0.97	0.35
	3	1.00	1.00	1.00	0.43
	4	1.00	0.85	0.87	0.95
	All	1.00	1.00	0.99	0.98

TABLE IV
FALSE ALARMS PER TRUE POTHOLE FOR POTHOLE DETECTION ON DIFFERENT ROAD COURSES (SEE TABLE I) WHEN TRAINED ON INDIVIDUAL COURSES.

		Testing Course			
		1	2	3	4
Training Course	1	0.20	0.01	0.03	0.23
	2	2.00	0.04	0.04	0.05
	3	2.60	0.14	0.05	0.04
	4	2.00	0.04	0.01	0.007
	All	1.00	0.05	0.01	0.03

does not always beat the SVM trained on individual courses, but overall is better able to handle diverse road conditions.

We performed a similar analysis by training with each of the three vehicle types individually and testing on the runs from the other vehicle types. Detection was similarly degraded when using classifiers trained for other vehicles. The SUV for example had higher peak accelerations than the Sedan for driving over the same potholes, causing the Sedan's accelerations to fall short of the boundary threshold trained for the SUV, resulting in undetected potholes.

These results reinforce the need to capture training data representing as many diverse scenarios as possible to create a general model. Simulations are used as the size and detail of the data set would be too costly to empirically collect.

VII. Real-World Data Results

To further validate our proposed framework, we conducted empirical experiments in real-world driving environments as outlined in Section III-B. To ensure that our established framework is generic enough to handle different types of data (e.g., simulated CarSim® data vs. empirical data collected from testing real vehicles), we directly apply the methods (i.e., 1 Hz SVM approach and the feature set) developed in Section V from our simulated data to our real-world measurement data set. We were able to detect 242 out of 268 marked potholes in the empirical data set, equal to a 90.3% detection rate. This encouraging result suggests that our road event detection framework developed with simulated CarSim® data works well on the real-world data.

Detection still performs well even when using an SVM designed for a lower operating frequency than what some of the sensors were using. The boundary for certain features is closer to the normal road data than for higher frequencies since the full range of data is not expected. For example, the maximum measured vertical acceleration for a vehicle driving over a pothole would be higher if 100 samples were available in a window rather than just 1. This closer boundary is problematic for the false alarm rate. For the real-world results, we initially registered 68 false alarms, equivalent to 0.25 false alarms per true pothole. However, many of those false alarms fell into well-defined classes:

- 9 false alarms following potholes, likely due to large aftershocks
- 14 false alarms from cracked or tiled roads.
- 30 false alarms from low speed travel ($<$ 30 km/h) and accelerating from or decelerating to a stop.

Since these low-speed scenarios occur frequently when driving, we heuristically added a filter to eliminate the anomaly caused by such low-speed scenarios. We avoid detecting potholes in any windows when the following conditions are satisfied: (1) $v < 30$ km/h, (2) $\frac{|a_y|}{v} > \frac{1\text{ g}}{50\text{ km/h}}$, and (3) $a_z < 0.26$ g. This reduced the total number of false alarms to 40 while reducing the detection rate only slightly to 88.9%. This filter is of a similar type to the features and boundaries already determined by the SVMs. This again emphasizes the importance of training with an extensive data set. Including stopping behavior in the simulations would allow us to integrate appropriate longitudinal acceleration properties into the SVMs and capture any nonlinear behavior. With the filter, if we further exclude the cracked roads in the false detections then the false alarm rate is reduced to a more manageable 0.10 false alarms per true pothole.

VIII. Conclusions

In this paper, we demonstrated how aggregating undersampled, heterogeneous, and distorted signals from embedded vehicle sensors in a crowdsourced fashion can be used for binary road feature detection, specifically as applied to potholes. We demonstrate multiple detection systems designed with bandwidth and computational constraints that differentiate how and where the raw signal data is processed to determine detections. We compare the systems in terms of their detection abilities and adherence to real-world constraints (low sampling rate of vehicle sensors and limited bandwidth for uploading data) for a varying number of participating vehicles. By using both CarSim® simulated data and real-world data collected from vehicle driving experiments, we show that our proposed framework provides a solid foundation for detecting potholes with high fidelity. In our simulated environment with 500 vehicles on a city road we could obtain a 99.6% detection rate with 0.01 false alarms per true pothole. Our empirical experiments show that our designed system could detect 88.9% of potholes on a 38.1 km stretch of road.

References

[1] W. Viriyasitavat, F. Bai, and O. Tonguz, "Dynamics of network connectivity in urban vehicular networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 3, pp. 515–533, March 2011.

[2] J. Fernandez, K. Borries, L. Cheng, B. Kumar, D. Stancil, and F. Bai, "Performance of the 802.11p physical layer in vehicle-to-vehicle environments," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 1, pp. 3–14, Jan 2012.

[3] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe, "Parknet: Drive-by sensing of road-side parking statistics," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10. New York, NY, USA: ACM, 2010, pp. 123–136.

[4] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "VTrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09. New York, NY, USA: ACM, 2009, pp. 85–98.

[5] J. Yoon, B. Noble, and M. Liu, "Surface street traffic estimation," in *MobiSys '07: 5th International conference on Mobile systems, applications and services*. New York, NY, USA: ACM, 2007, pp. 220–232.

[6] J. Davics, A. Beresford, and A. Hopper, "Scalable, distributed, real-time map generation," *Pervasive Computing, IEEE*, vol. 5, no. 4, pp. 47–54, Oct 2006.

[7] J. Ahn, Y. Wang, B. Yu, F. Bai, and B. Krishnamachari, "RISA: Distributed road information sharing architecture," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 1494–1502.

[8] H. Han, J. Yu, H. Zhu, Y. Chen, J. Yang, Y. Zhu, G. Xue, and M. Li, "SenSpeed: Sensing driving conditions to estimate vehicle speed in urban environments," in *INFOCOM, 2014 Proceedings IEEE*, April 2014, pp. 727–735.

[9] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The Pothole Patrol: Using a mobile sensor network for road surface monitoring," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*. New York, NY, USA: ACM, 2008, pp. 29–39.

[10] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08)*. New York, NY, USA: ACM, 2008, pp. 323–336.

[11] K. Chen, M. Lu, G. Tan, and J. Wu, "CRSM: Crowdsourcing based road surface monitoring," in *Proc. of the 11th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC)*, November 2013, pp. 2151–2158.

[12] M. Perttunen, O. Mazhelis, F. Cong, M. Kauppila, T. Leppänen, J. Kantola, J. Collin, S. Pirttikangas, J. Haverinen, T. Ristaniemi, and J. Riekki, "Distributed road surface condition monitoring using mobile phones," in *Proceedings of the 8th International Conference on Ubiquitous Intelligence and Computing*, UIC'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 64–78.

[13] C. Koch and I. Brilakis, "Pothole detection in asphalt pavement images," *Adv. Eng. Inform.*, vol. 25, no. 3, pp. 507–515, Aug. 2011.

[14] J. Karuppuswamy, V. Selvaraj, M. M. Ganesh, and E. L. Hall, "Detection and avoidance of simulated potholes in autonomous vehicle navigation in an unstructured environment," *Proc. SPIE*, vol. 4197, pp. 70–80, 2000.

[15] *CarSim: Quick Start Guide*, Mechanical Simulation Corporation, Ann Arbor, MI, 2013. [Online]. Available: http://www.carsim.com/

[16] J. Wilkinson, C. W. Mousseau, and T. Klingler, "Brake response time measurement for a HIL vehicle dynamics simulator," SAE Technical Paper, Tech. Rep., 2010.

[17] M. Ausloos and D. H. Berman, "A multivariate Weierstrass-Mandelbrot function," *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 400, no. 1819, pp. 331–350, 1985.

[18] J. J. Dawkins, "Terrain characterization and roughness estimation for simulation and control of unmanned ground vehicles," Ph.D. dissertation, Auburn University, Auburn, Alabama, December 2011.

[19] J. Y. Wong, *Theory of ground vehicles*. John Wiley & Sons, 2001.

[20] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[21] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–27, 2011, http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[22] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education, 2001.