

A Reliability-Aware Vehicular Crowdsensing System for Pothole Profiling

WEIDA ZHONG, SUNY Buffalo, USA

QIULING SUO, SUNY Buffalo, USA

FENGLONG MA, Pennsylvania State University, USA

YUNFEI HOU, California State University San Bernardino, USA

ABHISHEK GUPTA, SUNY Buffalo, USA

CHUNMING QIAO, SUNY Buffalo, USA

LU SU, SUNY Buffalo, USA

Accurately profiling potholes on road surfaces not only helps eliminate safety related concerns and improve commuting efficiency for drivers, but also reduces unnecessary maintenance cost for transportation agencies. In this paper, we propose a smartphone-based system that is capable of precisely estimating the length and depth of potholes, and introduce a holistic design on pothole data collection, profile aggregation and pothole warning and reporting. The proposed system relies on the built-in inertial sensors of vehicle-carried smartphones to estimate pothole profiles, and warn the driver about incoming potholes. Because of the difference in driving behaviors and vehicle suspension systems, a major challenge in building such system is how to aggregate conflicting sensory reports from multiple participating vehicles. To tackle this challenge, we propose a novel reliability-aware data aggregation algorithm called Reliability Adaptive Truth Discovery (RATD). It infers the reliability for each data source and aggregates pothole profiles in an unsupervised fashion. Our field test shows that the proposed system can effectively estimate pothole profiles, and the RATD algorithm significantly improves the profiling accuracy compared with popular data aggregation methods.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Networks** → *Sensor networks*.

Additional Key Words and Phrases: pothole profiling, crowd sensing, truth discovery

ACM Reference Format:

Weida Zhong, Qiuling Suo, Fenglong Ma, Yunfei Hou, Abhishek Gupta, Chunming Qiao, and Lu Su. 2019. A Reliability-Aware Vehicular Crowdsensing System for Pothole Profiling. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 4, Article 160 (December 2019), 26 pages. <https://doi.org/10.1145/3369815>

1 INTRODUCTION

One of the major distractions for safe and comfortable transportation is potholes on the road surface. A recent report by the American Automobile Association [2] shows that pothole-related damage costs American drivers

Authors' addresses: Weida Zhong, SUNY Buffalo, USA, weidazho@buffalo.edu; Qiuling Suo, SUNY Buffalo, USA, qiulings@buffalo.edu; Fenglong Ma, Pennsylvania State University, University Park, Pennsylvania, 16802, USA, fenglong@psu.edu; Yunfei Hou, California State University San Bernardino, San Bernardino, California, 92407, USA, yunfei.hou@csusb.edu; Abhishek Gupta, SUNY Buffalo, USA, agupta22@buffalo.edu; Chunming Qiao, SUNY Buffalo, USA, qiao@buffalo.edu; Lu Su, SUNY Buffalo, Buffalo, New York, 14260, USA, lushu@buffalo.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 Association for Computing Machinery.

2474-9567/2019/12-ART160 \$15.00

<https://doi.org/10.1145/3369815>

\$3 billion annually. In each year, about 3.2 million drivers in the U.S. have suffered pothole damage to their vehicles. The most severe damages are usually caused by large and deep potholes. Thus, it is extremely important for transportation agencies to be aware of the profile (e.g., length and depth) of potholes. Profiling potholes benefits both drivers and transportation agencies. Drivers should be noticed in advance to ensure driving safety, especially in bad weather when there is poor visibility or the pothole is covered by water or snow. Due to the limited resources of transportation agencies, prioritizing on which potholes need to be repaired first comes of substantial interest [20]. Therefore, designing effective systems to infer pothole profiles is in great need.

To infer the profile of potholes, traditional methods rely heavily on periodic road condition surveys, in which various dedicated devices are used, including ground penetrating radar [12], camera [13, 14, 16, 17, 27, 35], and so on. Though these dedicated devices can produce accurate results, their prohibitively high deployment costs and efforts make it impossible to achieve large-scale deployment, leading to limited road coverage and delayed information update.

To address this challenge, significant efforts are recently made to leverage the built-in sensors of smartphones (e.g., accelerometer, gyroscope and GPS) for the pothole profiling task. For example, [32] uses smartphone's accelerometer readings to get pothole profiles. However, due to the variety in vehicle conditions and the driving behaviors, profile measurements for the same pothole from different vehicles can be quite different from each other. Even the measurements of the same pothole collected by the same vehicle (at different time points) can be varying. In such case, directly averaging all the measurements of potholes (referred to as *claims* in this paper) collected from smartphones as the profiles is not accurate, since different vehicles may provide information of different quality (referred to as *reliability* in this paper).

Truth discovery methods [19, 21, 22, 25], due to their capability of capturing the source reliability, have been proved as effective ways to aggregate claims of multiple objects from multiple sources. In our problem settings, each vehicle equipped with a smartphone can be regarded as a *source*, and the *objects* are the potholes. The profiles of potholes are referred to as *truths*. Truth discovery methods are unsupervised and aim to simultaneously infer both truths of objects and reliability scores of sources. The basic idea of truth discovery approaches is that claims from more trustworthy sources are usually closer to truths than those from less trustworthy ones. A source will be assigned a higher reliability score if its claims are closer to the aggregated results, and its claims will be counted more in the aggregation procedure.

However, the existing truth discovery methods assume that each source makes at most one claim on each object, which is different from our scenario. In real life, a vehicle can pass through a certain road segment multiple times in a routine manner, which provides **more than one claims on the same object (i.e., pothole)**. As can be seen in Fig. 6, there may exist significant variance among a vehicle's claims on the same object. Intuitively, if we can capture and incorporate such variance into the aggregation process, we can improve the estimation on both source reliability and pothole profiles. Moreover, due to various reasons such as speed limit, traffic pattern, as well as the intrinsic characteristics of the potholes, it is highly possible that the same vehicle performs differently on different potholes. For example, different road segments may have different speed limits. Hitting a pothole at a higher speed could possibly yield claims with a higher variance. Thus, it is not desirable to assign a fixed reliability score to each vehicle. A fixed reliability score cannot describe the source accurately. It cannot determine the appropriate contribution for claims from this source in calculating the estimated profiles for different potholes. By capturing this inconsistency, we can infer more accurate aggregation results. The challenge here is how to accurately **characterize the dynamics of a vehicle's reliability** on different potholes.

In this paper, to tackle the aforementioned challenges, we build a system that has the ability to use smartphones' sensory data from moving vehicles for automatically inferring the profiles of potholes. We propose a new aggregation method called **R**eliability **A**daptive **T**ruth **D**iscovery (RATD) that is able to make full use of all claims of pothole profiles from all vehicles to infer more accurate results. In particular, we adaptively adjust a vehicle's reliability score based on not only the mean but also the variance of its claims on individual potholes. If a vehicle

provides claims with a higher variance on a certain pothole, it will be assigned a lower reliability score on that pothole. Based on the learned reliability, RATD then infers the estimated profile of that pothole. The aggregation procedure is formulated as an optimization problem, and we use block coordinate descent [3] to solve it. Finally, we evaluate the proposed method on the pothole profile database collected by our system, and show that our method outperforms state-of-the-art baselines. Our main contributions can be summarized as follows:

- We build a system that can collect motion sensor data of smartphones from moving vehicles, detect potholes on the road, infer the profiles of potholes including length and depth via aggregating reports from multiple vehicles, and eventually show the pothole profiles to end users.
- A new method called RATD has been proposed to make full use of all the repetitive claims from the same source on the same object. In this method, in addition to assigning an overall reliability score to each source, we also adjust a source's reliability score with respect to each individual object according to the quality and variance of its claims on that object.
- Extensive experiments have been carried out on the dataset that we have collected in real world. The evaluation results show that our proposed method RATD can reduce the MAE, RMSE and Error Rate for length inference by 35.3%, 30.43% and 35.08% respectively compared with state-of-the-art baselines and achieves the highest performance under all given circumstances, including the varying number of claims per source and the varying number of potholes that each car covers.

The rest of the paper is organized as follows. Section 2 introduces the architecture of the whole system. Section 3 goes through details about the data preprocessing and the pothole profiling from an individual claim. Section 4 describes our proposed algorithm. Section 5 presents the evaluation results. Discussion and related work are provided in Section 6 and Section 7. Section 8 concludes the paper.

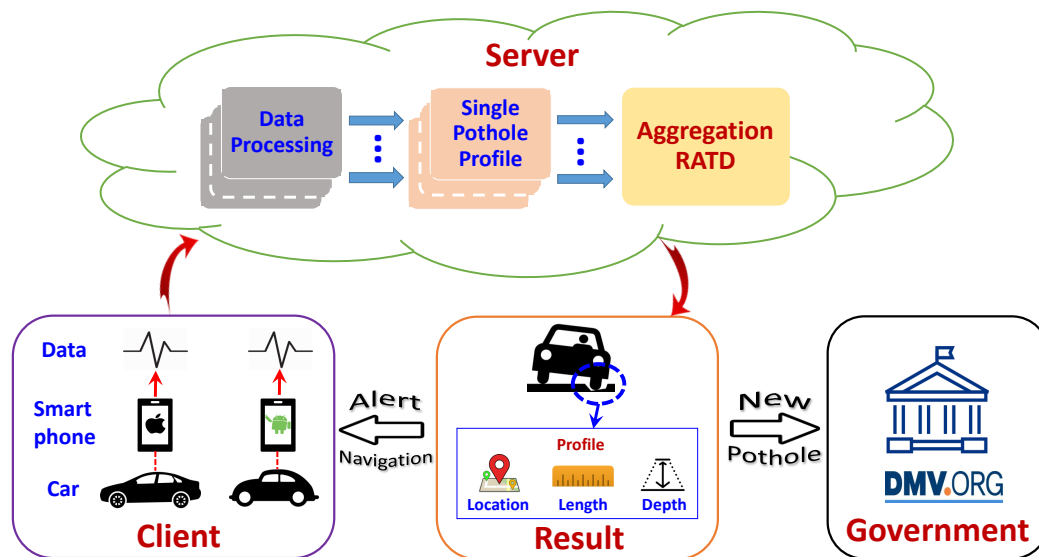


Fig. 1. System framework.

2 SYSTEM OVERVIEW

Fig. 1 shows the framework of our proposed system, which consists of three components: data collection from client vehicle to the server, pothole profiling on the server, and results displaying to end users as well as transportation agencies.

Data Collection: We develop an Android app to collect data via smartphone’s sensors including accelerometer, gyroscope and GPS, as well as an OBD (On-board diagnostics) II scanner that is connected to the smartphone via Bluetooth. The sampling rates and data fields of collected data are listed in Table 1. Specifically, we set the sampling rate of accelerometer to 200Hz. Although nowadays accelerometer can achieve a higher sampling rate (e.g., 400Hz) and yield a slightly better performance, it will drain more power and can even cause overheating problem. We use the OBD II scanner to collect the vehicle velocity which is usually more accurate than that estimated directly from GPS data. This is because OBD II retrieves the velocity data directly from the car system and also has a much higher sampling rate than GPS. The velocity from OBD II can also serve as the ground truth when we want to train a model to get more accurate velocity from GPS and accelerometer data. In addition, to reduce pressure on the internal storage, the network bandwidth and the battery, we only save the data related to potential pothole hitting events. If the vertical acceleration reading exceeds a certain threshold, then we keep only the data within 5 seconds before and after the event. This way, the app generates much smaller amount of data when the vehicle is running on roads with fewer potholes. The collected data is compressed and saved onto the phone in small chunks. Keeping each data file in a small size is very helpful for uploading, especially when the network connection is poor and a file might need to be re-uploaded in case of failure.

Table 1. Summary of Collected Data Information

Type	Sampling rate (Hz)	Collected fields
Accelerometer	200	timestamp, acceleration along x, y, and z axis
Gyroscope	200	timestamp, rotation rate around x, y, and z axis
GPS	1	timestamp, longitude, latitude, velocity
Vehicle velocity sensor	10	timestamp, velocity via OBD II

Pothole Profiling: The collected data from each client contains lots of noise, so we first pass it through a series of preprocessing steps, including filtering, GPS map matching, etc. Then the pothole profile, including both length and depth, is obtained from the data by modeling the vehicle as a single-degree-of-freedom system. Since there are chances that multiple vehicles hit the same pothole, and even the same vehicle may hit the same pothole multiple times, we need a method to aggregate all these pothole profiles from all vehicles on all potholes to produce more accurate estimations. Since existing data aggregation methods cannot handle this scenario effectively, we propose a reliability adaptive truth discovery method (RATD). This method can make full use of all repetitive claims and adjust each vehicle’s reliability across different potholes based on the variance in this vehicle’s claims on individual potholes. The data preprocessing steps and pothole profiling from an individual report are discussed in detail in Section 3, and our proposed data aggregation method is introduced in Section 4.

Results Displaying: The aggregated pothole profiles are saved into a Firebase Realtime Database¹, and updated with users by showing them as markers on the built-in Google Map². When a marker is clicked, the profile of the pothole will be displayed so that end users can know better about the road condition of their interested area. In addition to showing potholes on the map, we implement the basic navigation function. After

¹<https://firebase.google.com/docs/database/>

²<https://cloud.google.com/maps-platform/>

the user enters source and destination, a couple of candidate routes will be shown on the map. Beyond the expected travel time, the number of potholes will be displayed along with each route. Therefore, the user can choose the route with fewer potholes among ones of similar travel time, which could result in a better and safer driving experience. This navigation function is especially useful when a user is new to some area and unfamiliar with surrounding road conditions. When the app is switched to the navigation mode, the pothole markers will become unclickable so that the user won't get distracted. When a user is driving towards a pothole, the app will show a pop-up displaying the pothole profile along with a voice alert³. Besides, the estimated pothole profiles can be helpful to road maintenance authorities as they can prioritize potholes based on such information when the human and equipment resources are limited.

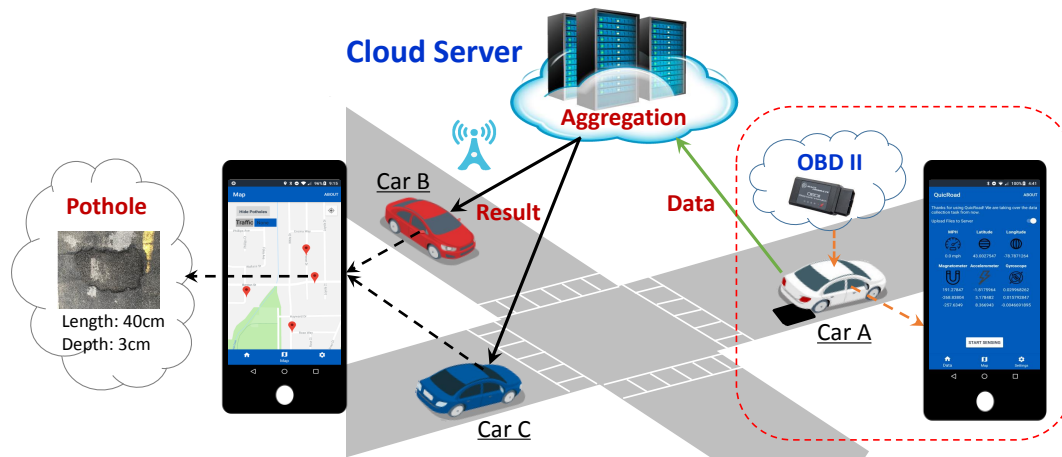


Fig. 2. A real world example of how the system works.

A Real World Example: Figure 2 illustrates how the system works in real life. Two screenshots of our Android app are shown. The screenshot on the right hand side pictures an interface displaying real time data from various sensors.

The left hand side screenshot shows the built-in Google Maps on which the detected potholes are marked. When car A (on the right hand side of the figure) hits a pothole, the collected sensory data will be sent back to the server for processing. The profiles including depth and length of the pothole are then inferred from the sensory data. After that, the estimated profiles on the server are aggregated using our proposed truth discovery algorithm. Our aggregation method assigns a reliability score to the claims from car A, indicating the degree of its information quality. With this reliability-aware aggregation method, an accurate estimation of pothole profiles can be obtained. The aggregated results are then saved into a cloud-based real-time database and pushed to all the clients in real time. The other two cars in the figure that are going to pass the road with the detected pothole will get an alert in advance, so that the drivers can be cautious when approaching the pothole or even take another route to avoid it.

3 POTHOLE PROFILING FROM INDIVIDUAL CLAIMS

In this section, we give the details about how a pothole is profiled from individual claims. In our design, each claim contains all the data listed in Table 1 that are collected by a vehicle when it hits a pothole. In most cases,

³This feature will be implemented in the future updates of the mobile app.

such data suffer from significant noises and coordinate misalignment. So we first take a series of preprocessing steps to remove noise and smooth the data, and then calculate the pothole profile according to each individual claim.

3.1 Data Preprocessing

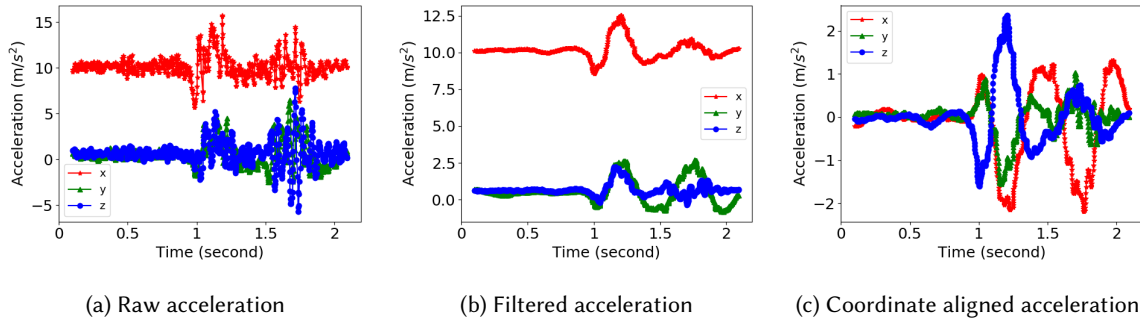


Fig. 3. Filtering and coordinate alignment for acceleration data.

Accelerometer and Gyroscope Data: Readings from accelerometer and gyroscope usually contain lots of noises. Fig. 3a shows an example of the collected raw acceleration data, where we can observe the impact that the noise has on the acceleration data. In the figure, x -axis represents the time, and y -axis is the acceleration along with 3 axes. The possible reasons for such noises include the mobility of the vehicle, uneven road surface, the vibration of the engine, and the inherent error of sensors. Thus, the first thing we need to do is to **remove noises**. In particular, we first apply a sliding window median filter, followed by a 2-order Butterworth filter to smooth the data. For the sliding window, we set the window size to be 15 sample points, which accounts for 0.0725 seconds in the 200Hz scenario and yields a reasonably good smoothing result. Setting the window size too small cannot achieve enough smoothing effect, whereas too large window sizes will make it hard to find the precise time points when the probing wheel enters and comes out of the pothole. The filtered acceleration data can be seen in Fig. 3b.

Another challenge in processing accelerometer and gyroscope data is that we need to use the vertical acceleration of the vehicle to calculate the pothole profile. However, depending on the posture of the smartphone in the vehicle, the coordinate system used by the smartphone sensors⁴ may not be well aligned with that of the car. So we follow a similar **coordinate alignment** method introduced in [30] but make substantial changes based on the features of our system to align the coordinate system of the phone with that of the vehicle by utilizing the data from the accelerometer, the gyroscope and the OBD II. In short, we need to get the rotation matrix

$$M = [I_i, I_j, I_k] = \begin{bmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ z_i & z_j & z_k \end{bmatrix}$$

to transfer acceleration readings $(x_p, y_p, z_p) \in R^{1 \times 3}$ from phone's coordinate system to corresponding ones $(x_c, y_c, z_c) \in R^{1 \times 3}$ in vehicle's coordinate system, i.e., $(x_c, y_c, z_c) = (x_p, y_p, z_p) \times M$. The three unit vectors in M are calculated as follows:

⁴https://developer.android.com/guide/topics/sensors/sensors_overview

- I_k : When the vehicle is stationary, the accelerometer only has gravity force applying on it. Although gravity can be read via an Android API directly, we find it is not accurate enough. Thus, we use exponential smoothing to retrieve the gravity component, which is then normalized to get unit vector I_k .
- I_j : After removing gravity from data, to get I_j , we detect the acceleration or deceleration phrase while the vehicle is moving in a straight line. Gyroscope readings are used to determine if the car is making a turn or not. We use velocity data to find an acceleration period, and retrieve the accelerometer readings during the same period. Out of the retrieved accelerometer readings, we select the one that is as orthogonal as possible to I_k and normalize it to get vector I_j .
- I_i : is the cross product of I_j and I_k .

Coordinate aligned acceleration data is shown in Fig. 3c. More specifically, Fig. 3a and Fig. 3b are using the smartphone's coordinates, but Fig. 3c is using the vehicle's coordinates. From Fig. 3c, we can tell that at around 1 second, the wheel enters a pothole since the z -axis' value drops sharply.

GPS Data: GPS data from smartphones usually suffers from the low accuracy problem due to the quality of the build-in GPS chips. The error could be big enough to make traces off the road. Occasionally GPS points can even jump between nearby roads especially when traveling around tall buildings. Therefore, we make use of the Snap to Roads API⁵ from Google Maps to do the **map matching**, which maps GPS points to the most reasonable road segments.

The original **velocity data** and the processed accelerometer, gyroscope and GPS data are interpolated to fill the gap in time points. This is because even if the accelerometer and the gyroscope have been set at the same sampling rate, timestamps of their readings are not necessarily the same, needless to say that the actual sampling rates cannot always keep up with the settings of the sampling rates in the app. So we interpolate the data to get estimated readings at evenly spaced time points.

After the aforementioned data processing, we can use these data to find out the pothole location. Similar method introduced in [9] has been used to determine if there is indeed a pothole at a certain location. The general idea is to design a series of filters to reject non-pothole events. A low velocity filter is used to remove non-pothole events, such as door slams, and a high-pass filter applied to get rid of the events introduced by turning, braking, etc. Three more filters including z -peak, xz -ratio and $velocity$ vs. z ratio are implemented. Parameters in these filters are learned to achieve the best locating accuracy.

3.2 Obtain Pothole Profile from Sensory Data

After the data goes through the above preprocessing procedure, it can be used to infer the profiles of potholes. We first calculate the depth and then the length. Consistent with daily driving experience, when a wheel hits a pothole, the vehicle will go through an underdamping vibration process. The vehicle can be treated as a single-degree-of-freedom (SDOF) system⁶ [31, 32]. A free vibration can be represented by $f(t) = kx(t) + c\dot{x}(t) + m\ddot{x}(t)$, where $f(t)$ is the external excitation force, k is the linear elastic stiffness coefficient of the spring, c is the linear viscous damping coefficient of the damper, m is the mass of the vehicle, $x(t)$ is the vertical displacement of vehicle body, $\dot{x}(t)$ is the derivative of $x(t)$, i.e., the vertical moving speed, and $\ddot{x}(t)$ is the second order derivative of $x(t)$, i.e., the vertical acceleration. Letting $y(t)$ stand for the vertical displacement of the probing wheel, we obtain $f(t) = ky(t) + c\dot{y}(t) + m_{car}\ddot{y}(t)$, where m_{car} is the weight of the car body. Thus we have $kx(t) + c\dot{x}(t) + m\ddot{x}(t) = ky(t) + c\dot{y}(t) + m_{car}\ddot{y}(t)$. By solving this equation, we can get $y(t)$. In the solving process, the $\ddot{y}(t)$ related term is omitted since the weight of the wheel is far less than the car body. In the end, we get the

⁵<https://developers.google.com/maps/documentation/roads/snap>

⁶<https://web.itu.edu.tr/gundes/sdof.pdf>

following expression,

$$y(t) = e^{-\frac{k}{c}t} \int \frac{f(t)}{c} e^{\frac{k}{c}t} dt = e^{-\frac{k}{c}t} \int \left(\frac{m}{c} \ddot{x}(t) + \dot{x}(t) + \frac{k}{c} x(t) \right) e^{\frac{k}{c}t} dt. \quad (1)$$

$y(t)$ is 0 at the time points before the wheel enters the pothole and after the wheel leaves the pothole. The smallest $y(t)$ between these two 0s can be treated as the depth of the pothole.

Let $x_p(t)$ stand for the vertical shift distance of the smartphone. Then $\ddot{x}_p(t)$ is the vertical acceleration of the smartphone, which can be retrieved from the accelerometer readings. The smartphone and the vehicle body undergo the same damping process with different amplitudes, and the relationship between $\ddot{x}_p(t)$ and $\ddot{x}(t)$ is determined by the position of the smartphone in the car. For example, in Fig. 4, l_1 is the distance between the phone and the left wheel, and l_2 is the distance between the phone and the right wheel. Then we can have $x_p(t) = \frac{l_1}{l_1+l_2}x(t)$. The same relationship also holds between $\dot{x}_p(t)$ and $\dot{x}(t)$, and between $\ddot{x}_p(t)$ and $\ddot{x}(t)$.

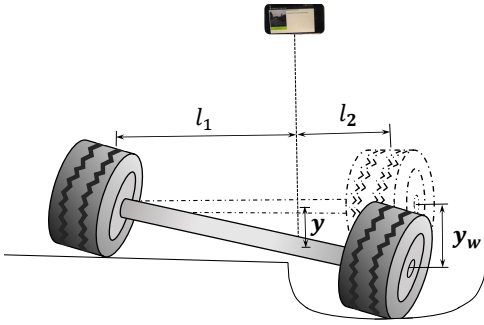


Fig. 4. Vertical displacement when hitting a pothole.

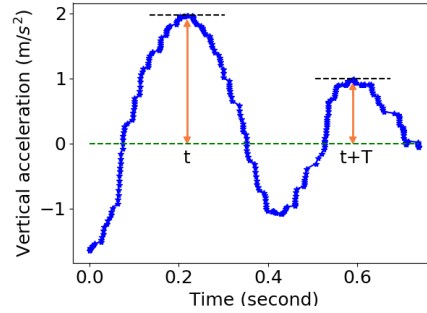


Fig. 5. Vertical acceleration $\ddot{x}_p(t)$ in underdamping process.

Therefore, in order to get $y(t)$ from Eq. (1), the vital task is to get the values of m , c , and k . Actually we don't need to know the exact values of these three parameters. Instead, we just need the values of two unique variables, i.e. $\frac{m}{c}$ and $\frac{k}{c}$. Next, we are going to talk about how to get these two variables.

When the probing wheel hits a pothole, the car is forced to vibrate, and the underdamping vibration starts. The car continues vibrating at its natural damped frequency with the amplitude gradually decreasing to 0. Since the underdamping vibration happens when the car is running on even road surface, $f(t)$ should be 0. So we have

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = 0 \rightarrow m\ddot{x}_p(t) + c\dot{x}_p(t) + kx_p(t) = 0. \quad (2)$$

The solving process of Eq. (2) can be found in the Appendix A.1. Its solution is,

$$x_p(t) = \bar{X}e^{-\sigma t} \cos(\omega t + \phi), \quad (3)$$

where $\sigma = \frac{c}{2m}$, $\omega = \sqrt{\frac{k}{m} - (\frac{c}{2m})^2}$, and \bar{X} and ϕ are the initial amplitude and phase of the underdamping vibration, respectively. The periodicity T of the underdamping vibration can be expressed as follows,

$$T = \frac{2\pi}{\omega} = \frac{2\pi}{\sqrt{\frac{k}{m} - (\frac{c}{2m})^2}} \approx \frac{2\pi}{\sqrt{\frac{k}{m}}} \rightarrow \frac{k}{m} = \left(\frac{2\pi}{T}\right)^2. \quad (4)$$

The damping ratio ζ , i.e. the ratio of vibration amplitudes at two successive peaks, is expressed as follows,

$$\zeta = \frac{x(t+T)}{x(t)} = \frac{\bar{X}e^{-\sigma(t+T)} \cos(\omega(t+T) + \phi)}{\bar{X}e^{-\sigma t} \cos(\omega t + \phi)} = e^{-\sigma T} \rightarrow \sigma = -\frac{\ln \zeta}{T} \rightarrow \frac{m}{c} = -\frac{T}{2 \ln \zeta}. \quad (5)$$

Since T and ζ can be read directly from $\hat{x}_p(t)$ which can be seen from Fig. 5, we can easily get the values of $\frac{m}{c}$ and $\frac{k}{c}$ by using Eq. (5) and Eq. 4. Then according to Eq. (1), we can get the vertical displacement of the probing wheel $y(t)$. And from $y(t)$, we can get two time points t_1 and t_2 around the pothole hitting event so that $y(t_1) = 0$ and $y(t_2) = 0$. t_1 and t_2 are the time points when the wheel enters and leaves the pothole, respectively. The absolute value of the lowest point of $y(t)$ between t_1 and t_2 is the depth of the pothole. Furthermore, let $\Delta t = t_2 - t_1$ and v be the mean velocity of vehicle during running over the pothole. Then we can calculate the length l of the pothole via $l = \Delta t * v$.

4 PROFILE AGGREGATION

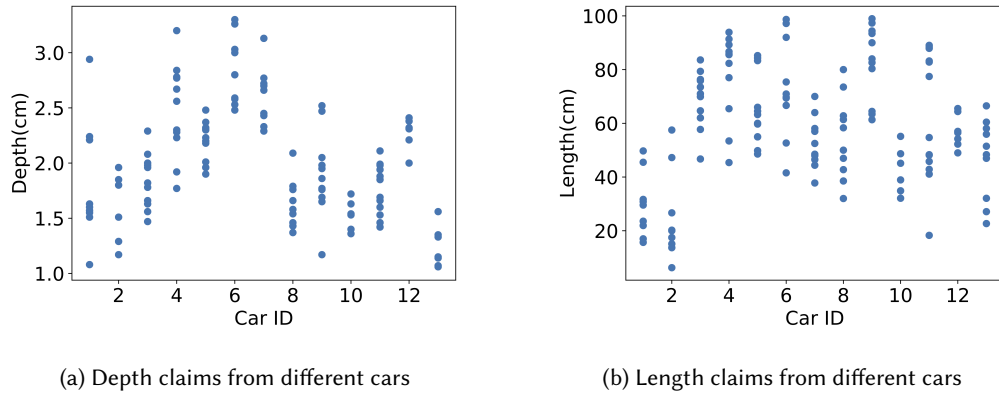


Fig. 6. Depth and length claims from different cars on one pothole.

In real practice, the same pothole can be hit by multiple vehicles. In some cases, the same pothole can even be run over by the same commuter for more than one time. Due to various reasons, such as the accuracy of locating the time points when the probing wheel enters and leaves the pothole and the velocity reading for the short pothole hitting period, different vehicles might report different profiles for the same pothole, and even the profiles obtained from multiple hits of the same vehicle can be different. Fig. 6 shows the depth and length claims from different cars on one pothole. In the figure, x -axis is the car ID, and y -axis is the value of claims about depth in Fig. 6a or length in Fig. 6b. We can see that on the same pothole, claims from different cars indeed fall into different ranges, and claims from the same car can vary vastly. In such cases, an individual profile claim reported from one pothole hitting event is highly likely to be inaccurate and even far from the true value. In order to obtain accurate pothole profile estimations, we propose a novel truth discovery method to effectively aggregate the claims.

4.1 Problem Definition

We first introduce the formal definition of the problem. The set of all cars and potholes are represented as \mathcal{S} and \mathcal{N} , respectively. $x_n^{s,i}$ is the content of the i_{th} profile report from car s on pothole n . Here x can be either length or depth. Then the problem is, given all profile reports $\{x_n^{s,i}\}_{n \in \mathcal{N}, s \in \mathcal{S}, i \in \mathcal{X}_n^s}$, to infer the estimated truths $\{\hat{x}_n\}_{n \in \mathcal{N}}$ for each pothole and figure out every car's reliability scores $\{w_s\}_{s \in \mathcal{S}}$. In this paper, we use weight and reliability score interchangeably. Table 2 lists the important notations used frequently in this paper.

Table 2. Frequently Used Notations

Symbol	Definition
\mathcal{S}	the set of sources (cars)
\mathcal{N}	the set of objects (potholes)
w_s	the weight (or reliability score) of source s
$x_n^{s,i}$	the i_{th} claim of source s on object n
\mathcal{X}_n^s	the set of claims on object n from source s
μ_n^s	the mean of claims on object n from source s
σ_n^{s2}	the variance of claims on object n from source s
$x_n^{(*)}$	the ground true value of object n
\hat{x}_n	the estimated truth of object n

4.2 Truth Discovery Recap

When there are claims about the same object from multiple sources, more often than not, there are conflicts among the claims from different sources. In this case, to infer the true properties of objects, one common way is to take the mean value of all claims for continuous data or use majority voting for categorical data as the estimated truth. In this way, all sources are treated equally and assigned the same weight for determining the final result. However, it ignores the difference of the source reliability, and may incorrectly take too much information from unreliable sources. To resolve the conflicts among multiple sources, truth discovery methods [19, 21, 22, 25] are proposed to capture the different reliability of each source in aggregating and therefore infer more accurate results. They assign each source a different weight based on the quality of their claims. Sources provide higher quality claims are assigned larger weights in inferring the final estimated truths. Usually the quality of a claim is measured by distance, i.e., how far away the claim is from the estimated truth since truth discovery methods are unsupervised. The basic format of the objective function in a truth discovery approach can be written as follows,

$$\begin{aligned} \min_{w_s, \hat{x}_n} \sum_{s \in \mathcal{S}} \sum_{n \in \mathcal{N}} w_s d_n^s, \\ \text{s.t. } \sum_{s \in \mathcal{S}} \exp(-w_s) = 1, \end{aligned} \quad (6)$$

where w_s is the weight of source s to be learned, and d_n^s represents the distance between the claim on object n from source s and the estimated truth. The constraint makes sure that the trivial solution where all weights $\{w_s\}_{s \in \mathcal{S}}$ are set to 0 is not valid.

4.3 The Reliability Adaptive Truth Discovery Algorithm

The existing truth discovery models can not be applied directly to our problem since they assume that one source makes at most one claim on each object. However, in our scenario, it is highly possible that one source provides more than one claims on the same object. For example, a certain pothole on the road with a single lane in each direction may be hit by the same commuter for many times. Moreover, previous methods assign each source s a single reliability score w_s , and each w_s remains unchanged on all objects. However, the reliability of an individual source may drift across different objects. Therefore, to overcome the aforementioned shortcomings, we propose a new algorithm called **R**eliability **A**daptive **T**ruth **D**iscovery (RATD), which can fully utilize the information of repetitive claims and adaptively adjust the reliability score of each source across objects.

4.3.1 Distance of Repetitive Claims. A simple solution to incorporate the repetitive claims is to use the mean value as the input to current truth discovery models. In this way, two cars with the same mean claim but different variances will be treated as having the same reliability. In fact, the variance of claims can also affect the source reliability. For example, a careful driver tends to keep consistent driving behavior and the claims that he/she provides have a low variance and are often closer to the truths. On the other hand, a driver with rough driving behavior would provide highly varying claims, which are often unreliable. Therefore, we need to consider both mean and variance when calculating the source reliability. To make full use of all the repetitive claims, we define the distance d_n^s between the claims \mathcal{X}_n^s from source s on object n and the object's estimated truth \hat{x}_n as follows,

$$d_n^s = \sum_{i=1}^{|\mathcal{X}_n^s|} \frac{(x_n^{s,i} - \hat{x}_n)^2}{|\mathcal{X}_n^s|}, \quad (7)$$

where $|\mathcal{X}_n^s|$ is the number of claims from source s on object n , and \hat{x}_n is the estimated truth for object n . The normalization term $|\mathcal{X}_n^s|$ ensures that the distances are not affected by the number of claims. Otherwise, the more claims one source makes on the same object, the larger distance value it may end up with, and the source will be incorrectly assigned a lower reliability score. The distance function can be further written as,

$$d_n^s = \sum_{i=1}^{|\mathcal{X}_n^s|} \frac{(x_n^{s,i} - \hat{x}_n)^2}{|\mathcal{X}_n^s|} = \sigma_n^{s^2} + (\mu_n^s - \hat{x}_n)^2, \quad (8)$$

where $\sigma_n^{s^2}$ and μ_n^s are the variance and the mean of claims from source s on object n , respectively. The proof of Eq. (8) can be found in the Appendix A.2. We can see that the distance d_n^s depends on not only the difference between the mean value and the estimated truth, but also the variance of the claims. This meets our assumption that the mean value itself does not contain all information. Under the condition of the same mean value, the larger variance yields a larger distance value. In other words, if there are two sources providing claims with the same mean value, then the source with the larger variance should be assigned a lower reliability score.

4.3.2 Adaptive Reliability. In reality, although the vehicle's condition does not change very often, the driver's driving behavior including velocity, acceleration rate, decelerate rate, etc., may change under certain circumstances. The changes of the driving behavior could have an impact on the source's reliability across different objects. Usually people tend to have similar driving behavior under similar circumstances, and once the environment changes, their driving behaviors often change accordingly. For example, one person may drive fast on straight road without taking brakes even if there are potholes, but decelerate sharply when there is a traffic jam. Hard brakes often cause the sensor readings to change sharply and lower the quality of the data. Thus for the same person, he/she could provide different quality of data under different conditions. Since the vehicle condition does not change across different potholes, we can assume that a source has its general overall reliability, but its reliability on specified object needs to be adjusted.

In the algorithm, instead of assigning a fixed weight to every source across all objects, we adjust a source's weight based on the variance in its claims on each object, i.e. $w_n^s = w_s g_n^s(\cdot)$, where w_n^s is the adjusted weight of source s on object n , and $g_n^s(\cdot)$ is a function related to source s 's claim variance on object n . Generally, $g_n^s(\cdot)$ should give more penalty to a source with higher variance in its claims. Therefore, we choose $g_n^s(\cdot)$ to be $\exp(\alpha \sigma_n^{s^2})$, where $\alpha > 0$ is a scale factor to control the magnitude of the function. If α is set too large, it will dominate the loss and all the sources will be assigned the same weight; whereas if it is too small, it can not add enough penalty to help adjust the sources' weights. In Section 5.8, we will discuss in detail about how α impacts the performance of the algorithm and how to choose an appropriate value of α .

4.3.3 *Optimization Framework.* After incorporating the object related weights, the objective function becomes,

$$\begin{aligned}
& \min_{w_n^s, \hat{x}_n} \sum_{s \in \mathcal{S}} \sum_{n \in \mathcal{N}} w_n^s d_n^s \\
& = \min_{w_s, \hat{x}_n} \sum_{s \in \mathcal{S}} \sum_{n \in \mathcal{N}} w_s \exp(\alpha \sigma_n^{s2}) d_n^s \\
& = \min_{w_s, \hat{x}_n} \sum_{s \in \mathcal{S}} w_s \left(\sum_{n \in \mathcal{N}} \exp(\alpha \sigma_n^{s2}) d_n^s \right), \\
& \text{s.t. } \sum_{s \in \mathcal{S}} \exp(-w_s) = 1.
\end{aligned} \tag{9}$$

where d_n^s can be obtained from Eq. (8). As σ_n^{s2} increases, $g_n^s(\cdot)$ increases and more penalty will be added to w_s , therefore, w_s tends to get a smaller value. The relationship between σ_n^{s2} and w_s can be seen later in Eq. (13).

There are two sets of parameters in Eq. (9), i.e. $\{w_s\}_{s \in \mathcal{S}}$ and $\{\hat{x}_n\}_{n \in \mathcal{N}}$, that need to be learned. For the optimization problems that involve two sets of variables, block coordinate descent [3] is a widely adopted approach. The general idea is to alternatively calculate one set of parameters while keeping the other fixed until the convergence criterion is satisfied. The convergence criterion is that the difference between the objective values from two consecutive iterations is smaller than a predefined threshold. In this paper, we empirically set it to be 10^{-6} which is pretty small. To minimize the objective value in Eq. (9), we iteratively conduct the following two steps.

Weights Update Step. In this step, the estimated truths $\{\hat{x}_n\}_{n \in \mathcal{N}}$ are considered to be known and fixed, and the weight of each source is updated. Applying the method of Lagrange multipliers, the objective function of Eq. (9) is transformed into the following formula,

$$\sum_{s \in \mathcal{S}} w_s \left(\sum_{n \in \mathcal{N}} \exp(\alpha \sigma_n^{s2}) d_n^s \right) + \lambda \left(\sum_{s \in \mathcal{S}} \exp(-w_s) - 1 \right). \tag{10}$$

Letting the partial derivative of Eq. (10) with respect to w_s be 0, we can get,

$$\lambda \exp(-w_s) = \sum_{n \in \mathcal{N}} \exp(\alpha \sigma_n^{s2}) d_n^s. \tag{11}$$

Combining Eq. (11) with the constraint in Eq. (9), we obtain,

$$\lambda = \sum_{s \in \mathcal{S}} \sum_{n \in \mathcal{N}} \exp(\alpha \sigma_n^{s2}) d_n^s. \tag{12}$$

After all the above steps, we can get the updating formula for weights $\{w_s\}_{s \in \mathcal{S}}$ as follows,

$$\begin{aligned}
w_s & = -\log \left(\frac{\sum_{n \in \mathcal{N}} \exp(\alpha \sigma_n^{s2}) d_n^s}{\sum_{s \in \mathcal{S}} \sum_{n \in \mathcal{N}} \exp(\alpha \sigma_n^{s2}) d_n^s} \right) \\
& = -\log \left(\frac{\sum_{n \in \mathcal{N}} \exp(\alpha \sigma_n^{s2}) (\sigma_n^{s2} + (\mu_n^s - \hat{x}_n)^2)}{\sum_{s \in \mathcal{S}} \sum_{n \in \mathcal{N}} \exp(\alpha \sigma_n^{s2}) (\sigma_n^{s2} + (\mu_n^s - \hat{x}_n)^2)} \right) \\
& = -\log \left(\frac{d_s}{\sum_{i \in \mathcal{S}} d_i} \right),
\end{aligned} \tag{13}$$

where,

$$d_s = \sum_{n \in \mathcal{N}} \exp(\alpha \sigma_n^{s2}) d_n^s, \tag{14}$$

is the adjusted distance between claims from source s and the estimated truths of all objects. The larger the adjusted distance d_s , the smaller the weight w_s , which keeps consistent with our assumption that the source with claims that are varying and far away from the truths should be assigned a smaller weight.

Truths Update Step. In this step, we assume that the weights $\{w_s\}_{s \in \mathcal{S}}$ of all sources are known, and we need to estimate the truths for all the objects. By applying the similar Lagrange multiplier method mentioned in the weights update step and taking partial derivative of Eq. (10) with respect to \hat{x}_n , we can get the update formula for the estimated truths $\{\hat{x}_n\}_{n \in \mathcal{N}}$ as follows,

$$\hat{x}_n = \frac{\sum_{s \in \mathcal{S}} w_s \exp(\alpha \sigma_n^{s2}) \mu_n^s}{\sum_{s \in \mathcal{S}} w_s \exp(\alpha \sigma_n^{s2})}. \quad (15)$$

From Eq. (15), we can see that the estimated truth is related to not only the mean values of claims from each source, but also the variances. The claims from the source with a larger variance tend to play less important roles in determining the estimated truth. This is indeed what we expect.

The aforementioned workflow is summarized in Algorithm 1.

Algorithm 1: Reliability Adaptive Truth Discovery.

Input: α , claims $\{x_n^{s,i}\}_{n \in \mathcal{N}, s \in \mathcal{S}, i \in \mathcal{X}_n^s}$

Output: Sources' weights $\{w_s\}_{s \in \mathcal{S}}$ and objects' estimated truths $\{\hat{x}_n\}_{n \in \mathcal{N}}$

- 1: Calculate the mean and variances $\{\mu_n^s, \sigma_n^{s2}\}_{s \in \mathcal{S}, n \in \mathcal{N}}$
 - 2: Initialize $\{\hat{x}_n\}_{n \in \mathcal{N}}$ using the mean claim for each object, i.e., $\{\mu_n\}_{n \in \mathcal{N}}$
 - 3: **repeat**
 - 4: Update sources' weights $\{w_s\}_{s \in \mathcal{S}}$ according to Eq. (13);
 - 5: Update objects' estimated truths $\{\hat{x}_n\}_{n \in \mathcal{N}}$ according to Eq. (15);
 - 6: **until** The convergence criterion is satisfied.
-

5 SYSTEM EVALUATION



Fig. 7. Example of potholes.



Fig. 8. Setup of data collection device.

In this section, we evaluate the performance of the proposed method RATD on the collected data using our system. We first give some details about the dataset. Then we introduce the state-of-the-art baseline methods and evaluation measures. In the evaluation results section, we compare the performance of all the methods and show that RATD outperforms all baselines. We conduct studies on how well RATD works in case of different number

of claims per pothole per car and different number of potholes each car covers. We also show the convergence of RATD. In the end, we analyze the impact that the scale factor α has upon the performance of RATD.

5.1 Evaluation Setup

We show examples of potholes in Fig. 7. During the data collection process, the smartphone is placed on the dashboard as shown in Fig. 8, and its distances to both the left and the right wheel are measured. Out of two weeks of driving data, we collect more than 1,150 pothole hitting reports coming from 13 different vehicles, including 2006 Toyota Corolla, 2015 Honda Civic, 2015 Volkswagen Touareg, 2016 Subaru Forester, 2017 Ford Focus, etc. In addition to the sensory data, we use the rear-view camera of the smartphone to record video during each trip which serves as the ground truth of pothole hitting events. We build an Apache HTTP server⁷ to receive and host all the collected data. During the data collection process, we measure the ground truths of all potholes several times across different days using a laser distance measurer. There is no noticeable change in the studied potholes' size during the relatively short span of data collection period.

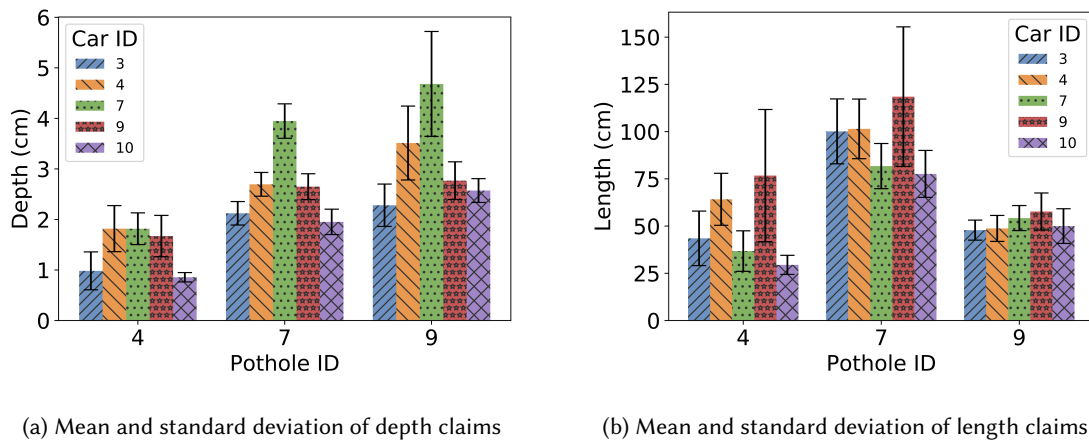


Fig. 9. Mean and standard deviation of claims on different potholes from different cars.

Some of the collected data are visualized in Fig. 9. Here we did not show the values of individual claims since we have shown in Eq. (8) and Eq. (9) that the aggregation results of the proposed RATD depend on the mean and variance of the claims instead of individual claims. Fig. 9a shows the mean and standard deviation of five cars' claims about the depth of three potholes, and Fig. 9b shows the mean and standard deviation of their lengths. The bars are the means and the error bars are the corresponding standard deviations. As we can see from these figures, for both depth and length, different cars have different claimed values for the same pothole, providing conflicting information about pothole profiles. Even the multiple claims from the same car on the same pothole may be different. For each individual car, the standard deviation of its claims also varies across different potholes. For example, from Fig. 9b, we can find that car 9 has larger variance on pothole 7 than that on pothole 9. These could result from a couple of reasons, e.g., the car's mechanical condition, the driver's driving behavior, and so on. We will study the possible factors in future work. Both the mean and the variance of repetitive claims can affect the aggregated results. For example, on length estimation, car 9 has larger standard deviations compared to

⁷<https://httpd.apache.org/>

other cars on the three potholes, which indicates that car 9 is less reliable. We will show in Section 5.6 that car 9 is indeed assigned a very small reliability score by our algorithm.

5.2 Baselines & Evaluation Measures

5.2.1 *Baselines.* We compare the performance of our proposed RATD with the following state-of-the-art baselines:

- **3-Estimates** [10]: uses accuracy to compute the quality of sources. During the calculation, it takes the difficulty of data records into consideration.
- **CATD** [18]: aims to detect truths from conflicting data with long-tail phenomenon. In addition to estimating the source reliability, it considers the confidence interval of the estimation.
- **CRH** [19]: formulates the conflicting resolving task as an optimization problem to minimize the overall weighted distance between the input and the estimated truths.
- **GTM** [36]: is based on Bayesian probabilistic models to solve conflict resolution problem for numerical data. It gets the maximum a posterior estimate for each object.
- **KDEm** [28]: introduces the concept of trustworthy opinion of an entity to address the uncertainty that we can only identify single or multiple reliable facts from opinions. It treats the trustworthy opinion as a random variable and uses its distribution to describe consistency and controversy.

Besides, **Mean** and **Median** are also used for comparison. For **Mean**, there are two ways to get the mean claim value for each object. One way is to take the average of all claims directly. The other way is that for each object, we first calculate the average of claims per source, and then take the mean of each source. In experiment, we find that these two methods yield very close results since the numbers of claims across different sources are not significantly different. For simplicity, we use the first method. The same is hold for **Median**.

For all other baseline methods, for each pothole, we use the mean value of claims from each vehicle as the vehicle's claim for that pothole. For all baseline methods, we implement and set parameters as suggested in the corresponding papers. In other words, we use their suggested tuning method/criteria/guidance to tune parameters instead of fixed values used in their work since optimal parameters may vary for different data sets.

5.2.2 *Evaluation Measures.* The following evaluation metrics have been used to compare the performances between our proposed RATD with baselines. For all three measures, the lower the score, the better the performance.

- (1) Mean Absolute Error (MAE): it is the average over the data points of the absolute differences between the estimation and the ground truth,

$$\text{MAE} = \frac{1}{N} \sum_{n=1}^N |x_n^{(*)} - \hat{x}_n|.$$

- (2) Root Mean Squared Error (RMSE): it measures the average distance between the estimation and the ground truth, and can be expressed as follows,

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^N (x_n^{(*)} - \hat{x}_n)^2}.$$

- (3) Error Rate: it is similar to MAE, but uses the relative instead of absolute difference,

$$\text{Error Rate} = \frac{1}{N} \sum_{n=1}^N \left| \frac{x_n^{(*)} - \hat{x}_n}{x_n^{(*)}} \right|.$$

Table 3. Performance Comparison.

Methods	Depth			Length		
	MAE	RMSE	Error Rate	MAE	RMSE	Error Rate
3-Estimates	0.7898	0.9319	0.3580	26.5853	28.8453	0.3623
CATD	1.0197	1.1852	0.2325	12.8877	14.2620	0.1779
CRH	0.6185	0.8625	0.2366	13.2034	14.5275	0.1826
GTM	0.7251	0.9667	0.2763	16.6239	20.0629	0.2267
KDEm	0.6398	0.9200	0.2574	21.0216	28.8521	0.2876
Mean	0.6320	0.8712	0.2427	13.7479	15.3119	0.1893
Median	0.6339	0.9104	0.2387	14.3939	16.2431	0.1988
RATD	0.5672	0.8254	0.2229	8.3389	9.9214	0.1155

5.3 Accuracy Analysis

Table 3 lists the performance of RATD along with all baselines. From the table we can find that RATD achieves the best results in terms of all the three performance metrics. Compared with the best performance among all baselines, the proposed RATD reduces MAE, RMSE and Error Rate by 8.29%, 4.3% and 4.13% respectively for depth, and 35.3%, 30.43% and 35.08% respectively for length. The performance of the baseline methods might suffer from different factors. For example, GTM assumes that the claims from all the sources for each object follows a Gaussian distribution which may not be true in our scenario. CATD is designed for data with long-tail phenomenon. Most of the truth discovery approaches, such as CRH, assign each vehicle with a fixed reliability score, and thus cannot fully capture the dynamics in the quality of the vehicle's claims across different potholes. For each object, all baselines can only make use of the mean claim from each source. All the baselines assume that each vehicle makes at most one claim on every pothole. If there are repetitive claims from the same vehicle on the same pothole, only the mean value is used as input. But the mean value does not contain all the useful information. In contrast, our proposed RATD model does not have any specific assumption about which particular distribution the data is following. It takes advantage of all the useful information from repetitive claims to infer the reliability of sources. Furthermore, RATD can adjust a source's reliability on different objects. In Section 5.6, we give more detailed discussions on source reliability learned by different models.

It is worth noting that for most of the examined methods, the error in length estimation is lower than that in depth estimation. This is due to the fact that the sensory measurements of pothole length is often more accurate than those of pothole depth. Compared to pothole depth, pothole length has a larger value scale, and are less affected by unexpected factors such as tire pressure. Since length estimation is generally more accurate than depth estimation, we focus on analyzing the methods on length estimation in the following sections.

5.4 Performance w.r.t. Number of Claims Per Car Per Pothole

In real life, the number of claims provided by vehicles may vary across potholes, and it could impact the accuracy of the aggregated results. Thus to show that our method works well under different settings, we randomly pick up from 2 to 14 claims per car per pothole as the input to evaluate the performance of all models. In the case that there is not enough number of claims for a certain pothole from a certain car, we use as many as they provide. We repeat the experiment with respect to each number of claims for 2,000 times, take the average performance as the final results, and plot the results in Fig. 10. We don't include all but the top four baselines which perform better than others on the length estimation in Table 3, so as to make the figures more clear.

In Fig. 10, we can see that for all methods, MAE, RMSE and Error Rate are all decreasing as the number of claims per pothole per car increases. This meets our expectation that more claims can provide more thorough

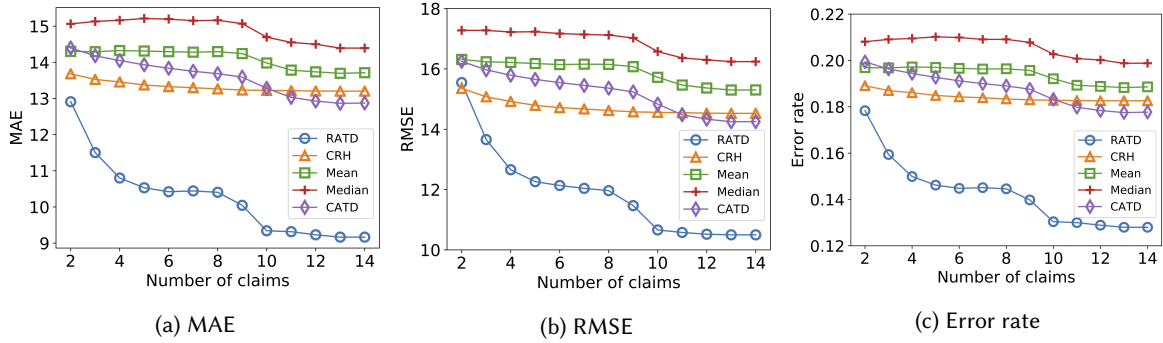


Fig. 10. Performance of different models with respect to the number of claims per car per pothole.

information about the pothole and reduce the impact of potential outliers. RATD benefits even more because more data on the same pothole can characterize the distribution, i.e., the mean and variance, of one car’s claims on one pothole more precisely so that the reliability of the car on different potholes can be learned more accurately. As a result, for each number of claims per car per pothole, RATD outperforms all baselines, and the gap of performance between RATD and the best baseline keeps enlarging as the number of claims increases. Especially, by using just 3 claims per car per pothole, RATD outperforms all baselines using 14 claims. This implies that our method is very effective even when the number of repetitive claims is very small. We also notice that with more than 10 claims per car per pothole, the performance of all methods stabilize. In other words, increasing the number of claims all the way up will not improve the performance infinitely. That’s because once the number reaches a certain threshold, we are already able to derive a good understanding of the distribution of the claims, and simply adding more claims cannot contribute more useful information. However, new claims are always needed since the pothole profiles might change over time. Therefore, to keep the algorithm running efficiently and keep the inferred results up to date, we can always replace old claims with the most recent ones from each vehicle.

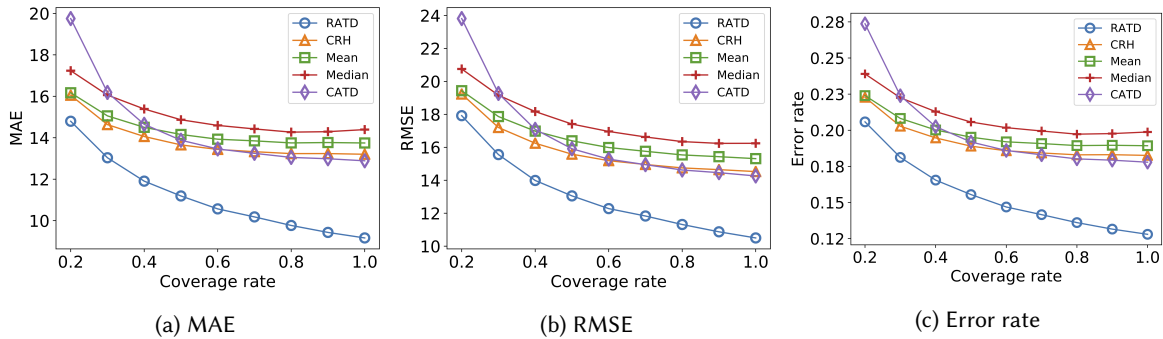


Fig. 11. Performance of different models with respect to car’s coverage rate.

5.5 Performance w.r.t Car’s Coverage Rate

Another scenario in real world is that most people travel within a relatively limited area for most of the time so that it is very likely that he or she will never run over some potholes while hitting other potholes over and

over again. A car's coverage rate is defined to be the ratio of the number of potholes that it covers with respect to the number of all the potholes that have been encountered by all cars. The coverage rate could impact the overall performance since lower coverage rate usually means that there are less number of claims on each pothole. Therefore, it is useful and necessary to study the performance of different methods under circumstances of different coverage rates. In the evaluation process, for each coverage rate ranging from 0.2 to 1.0 with the step size of 0.1, we randomly pick up corresponding number of potholes for each car, and feed associated claims as input into each method. The process is carried out for 2,000 times, and the average of all results are taken as the final results which are plotted in Fig. 11.

From Fig. 11 we can see that with the increase of coverage rate, the performance of all the methods are improving. This is because a higher coverage rate means more claims on each pothole and consequently some potential biases resulted from small number of claims get eliminated. What's more, for RATD, as the coverage rate gets higher for one car, we can have a better knowledge of the car's characteristics, and evaluate its reliability more precisely. Therefore, under each coverage rate, our method outperforms all the baselines and as the coverage rate increases, the performance gap becomes larger and larger. With less than 40 percent of coverage rate, RATD could outperform all the baselines having full coverage. This demonstrates that the proposed algorithm is more efficient and effective, and could take less data to get a better view of the pothole conditions.

5.6 Reliability Scores Learned by Different Methods

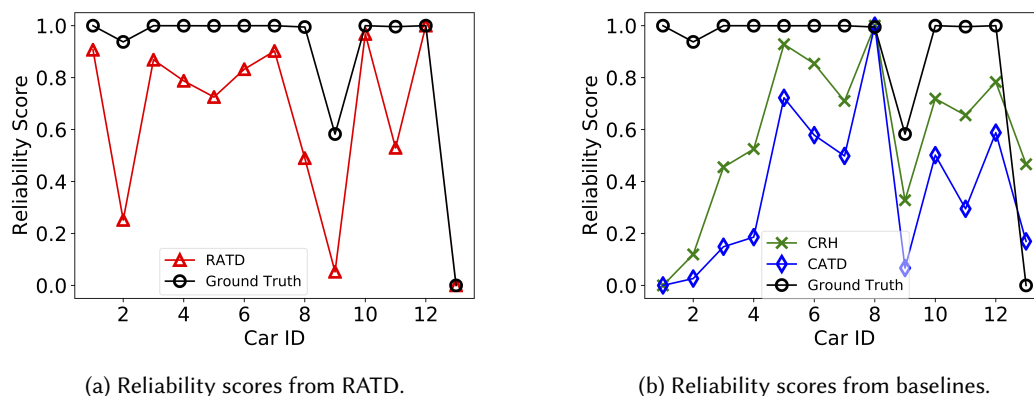


Fig. 12. Comparison of reliability scores.

In addition to estimating the profile for each pothole, truth discovery methods can also infer reliability scores $\{w_s\}_{s \in \mathcal{S}}$ of sources. Here we show the reliability scores learned by CRH, CATD and RATD. To demonstrate the accuracy of the learned reliability scores, the ground truth reliability score of each source is also used for comparison. In general, a source's ground truth reliability score is defined to be the overall similarity between its claims and the ground truths of objects. In this experiment, to get ground truth reliability scores, we first calculate the adjusted distance (defined in Eq. (14)) between each source's claimed and the ground truth pothole profiles. Then, these adjusted distances are normalized into the range of $[0, 1]$, and a source's ground truth reliability score is one minus its normalized distance. Thus, the ground truth values of the sources' reliability scores are in range $[0, 1]$. A source with a higher ground truth reliability score means its claims are more trustworthy. Please note that after normalization, a source with a reliability score equal to 0 is not absolutely untrustworthy, but has the relatively lowest reliability among all the sources. Similarly, the car with a reliability score equal to 1 indicates it

has the relatively highest reliability. For the purpose of comparison, the learned reliability scores of each method are normalized into the range of $[0, 1]$ as well. The comparison can be seen in Fig. 12. To make it clear, we show the reliability scores of our proposed RATD in Fig. 12a, and those of other truth discovery methods in Fig. 12b. The ground truths are plotted in both figures. From Fig. 12a we can see that there are a few cars with the ground truth reliability score close to 1. This is because the adjusted distances of car 13 is way larger than those of others. Thus after normalization, the reliability score of car 13 is 0, and a few others are close to 1. From the figures, we can see that RATD captures the sources' reliability scores more precisely than baseline methods. CRH and CATD especially struggle for estimating the reliability scores on car 1 to 7. Take car 1 and car 2 for example. The mean value of the claims from car 1 is further away from the ground truth than that of car 2, and both CRH and CATD determine a source's reliability based on the mean value, so car 1 is assigned a lower reliability score than car 2. But actually, the variance of car 1's claims is much smaller than that of car 2 so that the total distance for car 1 is smaller than that for car 2. That's why car 1 has been assigned a higher reliability score than car 2 by RATD. This demonstrates that RATD can achieve more accurate estimation of source reliability.

5.7 Convergence Analysis

In Fig. 13a, 13b and 13c, we show the convergence of MAE, RMSE and Error rate along with the number of iterations respectively. Since these three measures have different ranges, we plot them in separate figures. In these three figures, the x-axis denotes the number of iterations and y-axis the performance. For each measure, the value of the first iteration is the same as the performance of the mean method because the mean values are used as the initial input for RATD. It can be seen from these figures that RATD converges within as few as 4 iterations. The values of all three measures decline sharply in the first iteration which demonstrates the high running efficiency of our proposed model. This fast rate of convergence benefits from the exponential factor that we add in Eq. (9), which captures the difference between claims and estimated truths more quickly.



Fig. 13. Convergence of RATD

5.8 Impact of the Scale Factor α

Different choices of α in the objective function Eq. (9) could have different impact on the performance of the method. In this part, we study what influence its value has on the performance. The results have been shown in Fig. 14. Since these three measures have different ranges, we plot them in separate figures. We find that the best α for MAE, RMSE and Error Rate is 0.026, 0.021 and 0.026, respectively. This means that we cannot find a single α value to achieve the best MAE, RMSE, and Error Rate simultaneously. Intuitively, we find that the optimal value has the same order of magnitude as the reciprocal of the median of all variances $\{\sigma_n^{s^2}\}_{s \in S, n \in N}$. This ensures

$\exp(\alpha\sigma_n^{s^2})$ fall in a reasonable range compared with d_n^s . With α either decreasing or increasing from the optimal point, the performance declines. The reason behind it is that when α is approaching 0, Eq. (9) reduces to Eq. (6). Therefore, the performance gets close to that of CRH. Whereas when α is getting too large, the adjusted distance becomes more important than the reliability scores, so that all sources will be assigned similar reliability. Thus the performance is closer to the mean method.

In all aforementioned experimental results in Section 5.3– Section 5.7, we set $\alpha = 0.01$. Although it is not the optimal parameter, our method still outperforms all the baselines by a considerably large margin. This illustrates that our method can work in a relatively large range of parameter selection.

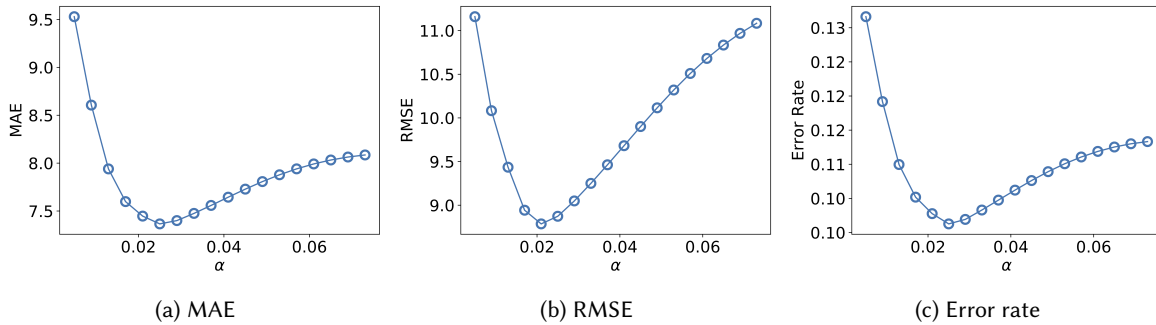


Fig. 14. Performance for length with respect to α .

5.9 Regression Methods for Single Pothole Profiling

The physical (SDOF) model used in Section 3.2 to infer a single pothole profile from the sensory data doesn't need any prior knowledge, nor any training process. We use this unsupervised model mainly because in real-world it is often difficult to collect enough data with ground truth. Here, we show how supervised machine learning methods perform in solving this problem. We first choose three popular regression models, i.e. linear regression, support vector regression (SVR) and CatBoost regressor [7], to carry out the single pothole profiling task. Then we use RATD to aggregate and get the final results.

In order to apply these regression algorithms, we first need to extract features from the sensory data. We cut 1 second of the vertical acceleration measurements as one data sample. Then we apply the fast Fourier transform (FFT) on these data samples which is one of the most common procedures to deal with time-series sensory data. After that, we extract the signal amplitudes, i.e., the root sum squared value of the real part and the imaginary part of the transformed data. Finally, we concatenate the amplitudes with the vehicle velocity as the input features. For all three regression methods, we split the data randomly into two sets, one with 60% of data as training set and the other with 40% as testing set. To demonstrate what impact the training data size can have upon the performance, we randomly choose 10% to 100% data from the training set to train the models. Then we apply the trained models to the testing set to get the regression results. We pass the regression results into RATD to obtain the aggregated estimations. For comparison, we also calculate the measures on the potholes in the testing data with the SDOF model. This data splitting, training, testing, and aggregation process has been carried out for 2,000 times, and the mean values are taken and depicted in Fig. 15. From the figure we can see that for all regression methods, with the training size being enlarged, the performance improves. The SDOF model achieves the best overall performance. Should there be more data, it is possible that the SDOF model would eventually be outperformed by SVR in terms of MAE, and by CatBoost in terms of RMSE. CatBoost performs poorly in MAE partially due to the loss function chosen to minimize RMSE.

Overall, this experiment shows that the SDOF physical model has a relatively better performance compared with three popular regression methods with a training set of reasonable size. There might exist other single pothole profiling methods, but due to the scope and space limitation of this paper, we leave that as a future work.

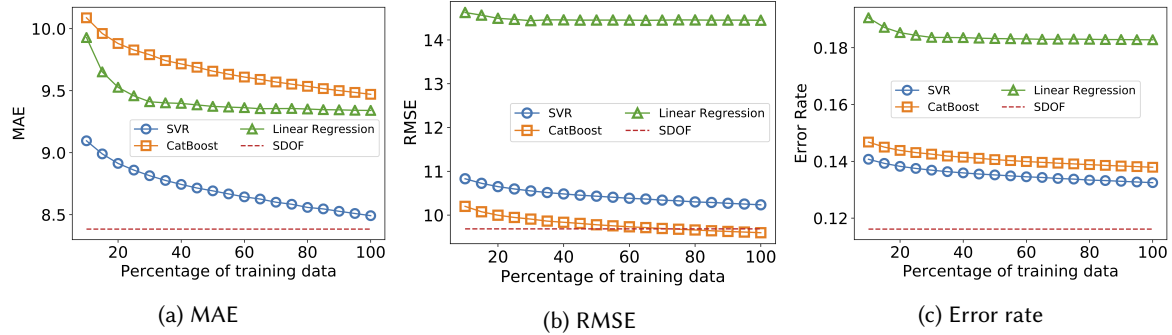


Fig. 15. Performance of regression methods with respect to the training size. We randomly choose 10%-100% data from the training set to train the regression models.

6 DISCUSSION

In this section, we discuss other large-scale deployment related issues and also comment on the other potential applications of the built system and the proposed aggregation algorithm.

Velocity From Other Sources: The velocity is an important factor in the calculation of the pothole length. In our system, we acquire the velocity using OBD II. Even though an off-the-shelf OBD II scanner is very cheap, it is still impractical to assume that majority of the drivers get it installed in their cars. However, since it is possible to obtain velocity information from GPS data [34], and/or from sensor data [11]. In a word, velocity will not be an obstacle to extend the system to a large scale. While collecting data, we also record the velocity from GPS data. We have compared the velocity obtained from the two sources and found out that the average difference between these two is within 5%. With the help of collected acceleration data, the accuracy of the velocity obtained from GPS can be further improved since the high sampling rate of the acceleration can help achieve better calibration and interpolation of GPS data.

Lane Level Pothole Detection and Profiling: With GPS data and appropriate clustering method, we can find the approximate location of the pothole on the road. If the pothole is on a road with multiple lanes, with the help of smartphone-based lane changing behavior recognition [5] and lane detection [1] techniques, it is possible to determine on which lane the pothole resides. Further, we can even know which side of the car hits the pothole by mining the readings from the gyroscope and/or the accelerometer. Afterwards we will be able to provide drivers with alerts containing more useful information. For example, we can inform the driver if he/she needs to keep to the left side or right side of current lane, or change to a different lane to avoid hitting a pothole ahead.

Power Consumption: Using the app all the time puts pressure on the battery. A couple of measures can be taken to alleviate low-battery anxiety. The first is to reduce the sampling rate. Since a lower sampling rate may harm the profiling accuracy, we need to find a trade off between sampling rate and performance. If a user is using the navigation, then turning the screen off is no longer an option. We could try to lower the power cost of the pothole profiling activity itself. One possible solution is to dynamically adjust the sampling rates based on the road surface roughness since it is less likely to encounter a pothole on a smooth road segment than on a bumpy one. And the road roughness condition can be learned from the collected sensory data [8]. Besides, improving the

accuracy of pothole hitting event detection can also reduce the amount of collected data, and hence less power will be used for data saving and uploading.

Privacy: Nowadays, privacy protection becomes a more and more important task as there are so many possible ways to track back to the actual person through data collected from different sources. In the system implementation, we take a lot of precaution to protect the privacy of end users. Since only data around potential pothole hitting events is uploaded to the server, it is very hard if not completely impossible to recover the whole driving trace. For the same reason, the server has no clue of the start and end information of any trip, so neither the source nor the destination can be retrieved and user privacy can be preserved. Only features used for pothole profiling are extracted. After features extraction, the raw data is discarded. Besides, the raw data will never be shared with any third party.

Extensibility: Both the built pothole profiling system and the proposed aggregation method can be easily extended to solve many other problems. For example, it is possible to integrate more transportation related features into this framework, such as road roughness condition monitoring, slippery road detection, traffic condition monitoring, driving behavior analysis, and so on. These features could help provide a more comfortable, efficient, and safety driving environment. In addition, the proposed aggregation algorithm can be applied to solve other problems with similar settings, i.e., multiple sources make potential conflicting claims on multiple objects while one source makes more than one claims on the same object. These problems can come from more broader domains in addition to transportation, such as, air quality sensing, weather condition estimation [25], and etc.

7 RELATED WORK

Potholes jeopardize road safety and transportation efficiency. They often contribute to car accidents. Therefore, various tools have been developed on the pothole detection and profiling problem. [4, 16, 27] used histogram and canny edge detection based method to detect potholes. [17] and [35] used images to infer the severity of potholes on the pavement. [14] presented a solution to automatically detect and assess the severity of potholes using vision-based data. It used 2D recognition to detect potholes and 3D reconstruction to measure the width and depth of potholes. [13] proposed a pothole detection system using a commercial black-box camera to reduce cost. [23] proposed a cost-effective solution to identify the potholes on roads and provided timely alerts to drivers to avoid accidents or vehicle damages. Ultrasonic sensors are used to identify the potholes and also to measure their depth. [15] used a vehicle mounted sensor system consisting of a Microsoft Kinect and a USB camera to detect and analyse potholes. These aforementioned methods can produce accurate results, but need specific devices which restricts the wide deployment of these systems due to the cost and/or efforts to install dedicated devices. To lower the cost and achieve large-scale deployment, smartphone based methods have been proposed. The Pothole Patrol [9] applied multiple filters on the acceleration data to remove non-pothole data and then used a machine learning model to identify potholes. Nericell [26] used simple threshold-based method, i.e. z-sus for speeds under 25 km/h and z-peak for speed above that to detect bumps and potholes. [24] used four different thresholds to detect potholes from acceleration data. [32] used the accelerometer readings from smartphones to estimate the depth and length of potholes. Due to the limitation of hardware and other factors, these methods usually suffer from low accuracy. In this work, we propose a new truth discovery method to improve the pothole profiling accuracy by taking advantage of crowdsensing data where a pothole is usually hit by multiple vehicles, and a pothole can even be hit by the same vehicle for more than once.

Truth discovery methods [6, 10, 18, 19, 21, 22, 25, 28, 29, 33, 36] are effective in estimating true properties of objects from conflicting claims. Authors in [25] took into consideration the correlation between sources, and divided sources into disjoint independent sets. 3-Estimates [10] used accuracy to infer the quality of sources. During the calculation, it took the difficulty of data records into consideration. CRH [19] formulated the conflict resolving task of heterogeneous data as an optimization problem to minimize the overall weighted distance

between the input and the estimated truths. CATD [18] introduced an algorithm to discover truths from conflicting data with long-tail phenomenon. In addition to estimating the source reliability, it considered the confidence interval of the estimation. GTM [36] was based on Bayesian probabilistic models to solve conflict resolution problem for numerical data. It got the maximum a posterior estimate for each object. KDEm [28] introduced the concept of trustworthy opinion of an entity. It treated the trustworthy opinion as a random variable and used its distribution to describe consistency and controversy. [22] proposed a fine grained truth discovery method to deal with the scenarios where users have various expertise levels on different topics. [33] and [6] incorporated dependency between data sources in truth discovery. Generally, there are two popular ways to formulate a truth discovery problem, i.e., optimization and maximum likelihood estimation. Our proposed RATD model is based on the optimization branch. However, different from the existing optimization based truth discovery methods, our RATD is designed for applications where each source makes multiple claims on the same object. Besides, instead of assigning each source a fixed reliability score, our model can adjust each source's reliability across different objects according to the statistics of the source's claims on each object.

8 CONCLUSIONS

In this paper, we design, develop and evaluate a mobile crowd sensing system for pothole profiling via motor vehicles. We propose a new method RATD to effectively aggregate reported profiles from different cars and find out each car's weight in determining the final results. Specifically, the RATD method can effectively handle the case that each car makes more than one claims on each object. In addition to assigning an overall reliability score to each source, RATD can adjust the score based on statistics of the car's claims on each pothole. We evaluate the system on the collected dataset. The experimental results show that our method can reduce the MAE, RMSE and Error Rate for the inference of pothole length by at least 35.3%, 30.43% and 35.08% respectively compared with state-of-the-art baselines.

ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation under Grants CNS-1652503 and CNS-1737590. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NSF.

REFERENCES

- [1] Heba Aly, Anas Basalamah, and Moustafa Youssef. 2016. Robust and ubiquitous smartphone-based lane detection. *Pervasive and Mobile Computing* 26 (2016), 35–56.
- [2] The American Automobile Association. 2016. *Pothole Damage Costs Drivers \$3 Billion Annually Nationwide*. Retrieved February 17, 2016 from <http://news.aaa-calif.com/news/pothole-damage-costs-drivers-3-billion-annually-nationwide>
- [3] Dimitri P Bertsekas. 1999. *Nonlinear programming*. Athena scientific Belmont.
- [4] Emir Buza, Samir Omanovic, and Alvin Huseinovic. 2013. Pothole detection with image processing and spectral clustering. In *Proceedings of the 2nd International Conference on Information Technology and Computer Networks*. 48–53.
- [5] Dongyao Chen, Kyong-Tak Cho, Sihui Han, Zhizhuo Jin, and Kang G Shin. 2015. Invisible sensing of vehicle steering with smartphones. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 1–13.
- [6] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. 2009. Integrating conflicting data: the role of source dependence. *Proceedings of the VLDB Endowment* 2, 1 (2009), 550–561.
- [7] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. 2018. CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363* (2018).
- [8] Yuchuan Du, Chenglong Liu, Difei Wu, and Shengchuan Jiang. 2014. Measurement of international roughness index by using-axis accelerometers and GPS. *Mathematical Problems in Engineering* 2014 (2014).
- [9] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. 2008. 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*. ACM, 29–39.

- [10] Alban Galland, Serge Abiteboul, Amélie Marian, and Pierre Senellart. 2010. Corroborating information from disagreeing views. In *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 131–140.
- [11] Yunfei Hou, Abhishek Gupta, Tong Guan, Shaohan Hu, Lu Su, and Chunming Qiao. 2017. VehSense: Slippery Road Detection Using Smartphones. In *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*. IEEE, 1–5.
- [12] Dryver R Huston, Noel V Pelczarski, Brian Esser, and Kenneth R Maser. 2000. Damage detection in roadways with ground penetrating radar. In *Eighth International Conference on Ground Penetrating Radar*, Vol. 4084. International Society for Optics and Photonics, 91–95.
- [13] Youngtae Jo and Seungki Ryu. 2015. Pothole detection system using a black-box camera. *Sensors* 15, 11 (2015), 29316–29331.
- [14] GM Jog, C Koch, M Golparvar-Fard, and I Brilakis. 2012. Pothole properties measurement through visual 2D recognition and 3D reconstruction. In *Computing in Civil Engineering (2012)*. 553–560.
- [15] Deon Joubert, Ayanda Tyatyantsi, Jeffry Mphahlehle, and Vivian Manchidi. 2011. Pothole tagging system. (2011).
- [16] Christian Koch and Ioannis Brilakis. 2011. Pothole detection in asphalt pavement images. *Advanced Engineering Informatics* 25, 3 (2011), 507–515.
- [17] Christian Koch, Gauri M Jog, and Ioannis Brilakis. 2012. Automated pothole distress assessment using asphalt pavement video data. *Journal of Computing in Civil Engineering* 27, 4 (2012), 370–378.
- [18] Qi Li, Yaliang Li, Jing Gao, Lu Su, Bo Zhao, Murat Demirbas, Wei Fan, and Jiawei Han. 2014. A confidence-aware approach for truth discovery on long-tail data. *Proceedings of the VLDB Endowment* 8, 4 (2014), 425–436.
- [19] Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, and Jiawei Han. 2014. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 1187–1198.
- [20] Xiao Li and Daniel W Goldberg. 2018. Toward a mobile crowdsensing system for road surface assessment. *Computers, Environment and Urban Systems* 69 (2018), 51–62.
- [21] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A survey on truth discovery. *ACM Sigkdd Explorations Newsletter* 17, 2 (2016), 1–16.
- [22] Fenglong Ma, Yaliang Li, Qi Li, Minghui Qiu, Jing Gao, Shi Zhi, Lu Su, Bo Zhao, Heng Ji, and Jiawei Han. 2015. Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 745–754.
- [23] Rajeshwari Madli, Santosh Hebbar, Praveenraj Pattar, and Varaprasad Golla. 2015. Automatic detection and notification of potholes and humps on roads to aid drivers. *IEEE sensors journal* 15, 8 (2015), 4313–4318.
- [24] Artis Mednis, Girts Strazdins, Reinholds Zviedris, Georgijs Kanonirs, and Leo Selavo. 2011. Real time pothole detection using android smartphones with accelerometers. In *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*. IEEE, 1–6.
- [25] Chuishi Meng, Wenjun Jiang, Yaliang Li, Jing Gao, Lu Su, Hu Ding, and Yun Cheng. 2015. Truth discovery on crowd sensing of correlated entities. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. ACM, 169–182.
- [26] Prashanth Mohan, Venkata N Padmanabhan, and Ramachandran Ramjee. 2008. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 323–336.
- [27] Seung-Ki Ryu, Taehyeong Kim, and Young-Ro Kim. 2015. Image-based pothole detection system for ITS service and road management system. *Mathematical Problems in Engineering* 2015 (2015).
- [28] Mengting Wan, Xiangyu Chen, Lance Kaplan, Jiawei Han, Jing Gao, and Bo Zhao. 2016. From truth discovery to trustworthy opinion discovery: An uncertainty-aware quantitative modeling approach. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1885–1894.
- [29] Yaqing Wang, Fenglong Ma, Lu Su, and Jing Gao. 2017. Discovering truths from distributed data. In *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 505–514.
- [30] Yan Wang, Jie Yang, Hongbo Liu, Yingying Chen, Marco Gruteser, and Richard P Martin. 2013. Sensing vehicle dynamics for determining driver phone use. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 41–54.
- [31] Jan Robert Wright and Jonathan Edward Cooper. 2008. *Introduction to aircraft aeroelasticity and loads*. Vol. 20. John Wiley & Sons.
- [32] Guangtao Xue, Hongzi Zhu, Zhenxian Hu, Jiadi Yu, Yanmin Zhu, and Yuan Luo. 2017. Pothole in the dark: Perceiving pothole profiles with participatory urban vehicles. *IEEE Transactions on Mobile Computing* 16, 5 (2017), 1408–1419.
- [33] Xiaoxin Yin, Jiawei Han, and S Yu Philip. 2008. Truth discovery with multiple conflicting information providers on the web. *IEEE Transactions on Knowledge and Data Engineering* 20, 6 (2008), 796–808.
- [34] Zhang Yong-chuan, Zuo Xiao-qing, Chen Zhen-ting, et al. 2011. Traffic congestion detection based on GPS floating-car data. *Procedia Engineering* 15 (2011), 5541–5546.
- [35] X Yu and E Salari. 2011. Pavement pothole detection and severity measurement using laser imaging. In *2011 IEEE International Conference on Electro/Information Technology*. IEEE, 1–5.
- [36] Bo Zhao and Jiawei Han. 2012. A probabilistic model for estimating real-valued truth from conflicting sources. *Proc. of QDB* (2012).

A APPENDIX

A.1 Solving Process of Eq. (2)

SOLUTION. Substitute $x(t) = Xe^{\lambda t}$ into it, and we can get a quadratic equation, which has two roots,

$$\lambda_{1,2} = -\frac{c}{2m} \pm \sqrt{\left(\frac{c}{2m}\right)^2 - \frac{k}{m}}. \quad (16)$$

Since in our case, it is underdamping vibration, i.e. the damping rate c is positive, but less than the critical damping rate $c_c = 2\sqrt{mk}$, and the system will oscillate freely from some initial displacement and velocity, the roots should be complex conjugates, i.e. $\lambda_1 = \lambda_2^*$. Therefore we can have,

$$\begin{aligned} \lambda &= -\frac{c}{2m} + i\sqrt{\frac{k}{m} - \left(\frac{c}{2m}\right)^2} \\ &= -\sigma + iw, \end{aligned} \quad (17)$$

where $\sigma = \frac{c}{2m}$, and $w = \sqrt{\frac{k}{m} - \left(\frac{c}{2m}\right)^2}$.

Letting $X = A + iB$, then $x(t)$ can be further expressed as,

$$\begin{aligned} x(t) &= Xe^{\lambda t} + X^*e^{\lambda^* t} \\ &= e^{-\sigma t} (2A \cos(wt) - 2B \sin(wt)) \\ &= \bar{X}e^{-\sigma t} \cos(wt + \phi), \end{aligned} \quad (18)$$

where \bar{X} and ϕ are the initial amplitude and phase of the underdamping vibration, respectively. □

A.2 Proof of Eq. (8)

PROOF. Let μ_n^s represent the mean of source s 's claims on object n , then

$$\begin{aligned} d_n^s &= \sum_{i=1}^{|\mathcal{X}_n^s|} \frac{(x_n^{s,i} - \hat{x}_n)^2}{|\mathcal{X}_n^s|} \\ &= \sum_{i=1}^{|\mathcal{X}_n^s|} \frac{(x_n^{s,i} - \mu_n^s + \mu_n^s - \hat{x}_n)^2}{|\mathcal{X}_n^s|} \\ &= \underbrace{\sum_{i=1}^{|\mathcal{X}_n^s|} \frac{(x_n^{s,i} - \mu_n^s)^2}{|\mathcal{X}_n^s|}}_{T_1} + \underbrace{\sum_{i=1}^{|\mathcal{X}_n^s|} \frac{2(x_n^{s,i} - \mu_n^s)(\mu_n^s - \hat{x}_n)}{|\mathcal{X}_n^s|}}_{T_2} + \underbrace{\sum_{i=1}^{|\mathcal{X}_n^s|} \frac{(\mu_n^s - \hat{x}_n)^2}{|\mathcal{X}_n^s|}}_{T_3}. \end{aligned} \quad (19)$$

In Eq. (19), the first term T_1 is exactly the variance of source s 's claims on object n , i.e. $\sigma_n^{s^2}$. The second term can be further evaluated as follows,

$$\begin{aligned} T_2 &= \sum_{i=1}^{|\mathcal{X}_n^s|} \frac{2(x_n^{s,i} - \mu_n^s)(\mu_n^s - \hat{x}_n)}{|\mathcal{X}_n^s|} \\ &= \frac{2(\mu_n^s - \hat{x}_n)}{|\mathcal{X}_n^s|} \sum_{i=1}^{|\mathcal{X}_n^s|} (x_n^{s,i} - \mu_n^s) \\ &= 0. \end{aligned} \quad (20)$$

Last, the third term can be evaluated as follows,

$$T_3 = \sum_{i=1}^{|\mathcal{X}_n^s|} \frac{(\mu_n^s - \hat{x}_n)^2}{|\mathcal{X}_n^s|} = (\mu_n^s - \hat{x}_n)^2. \quad (21)$$

In a summary,

$$d_n^s = T_1 + T_2 + T_3 = \sigma_n^{s^2} + (\mu_n^s - \hat{x}_n)^2. \quad (22)$$

□