Experiences with GreenGPS—Fuel-Efficient Navigation Using Participatory Sensing

Fatemeh Saremi, Omid Fatemieh, Hossein Ahmadi, Hongyan Wang, Tarek Abdelzaher, Raghu Ganti, Hengchang Liu, Shaohan Hu, Shen Li, and Lu Su, *Member, IEEE*

Abstract—Participatory sensing services based on mobile phones constitute an important growing area of mobile computing. Most services start small and hence are initially sparsely deployed. Unless a mobile service adds value while sparsely deployed, it may not survive conditions of sparse deployment. The paper offers a generic solution to this problem and illustrates this solution in the context of *GreenGPS*; a navigation service that allows drivers to find the most fuel-efficient routes customized for their vehicles between arbitrary end-points. Specifically, when the participatory sensing service is sparsely deployed, we demonstrate a general framework for generalization from sparse collected data to produce models extending beyond the current data coverage. This generalization allows the mobile service to offer value under broader conditions. GreenGPS uses our developed participatory sensing infrastructure and generalization algorithms to perform inexpensive data collection, aggregation, and modeling in an end-to-end automated fashion. The models are subsequently used by our backend engine to predict customized fuel-efficient routes for both members and non-members of the service. GreenGPS is offered as a mobile phone application and can be easily deployed and used by individuals. A preliminary study of our green navigation idea was performed in [1], however, the effort was focused on a proof-of-concept implementation that involved substantial offline and manual processing. In contrast, the results and conclusions in the current paper are based on a more advanced and accurate model and extensive data from a real-world phone-based implementation and deployment, which enables reliable and automatic end-to-end data collection and route recommendation. The system further benefits from lower cost and easier deployment. To evaluate the green navigation service efficiency, we conducted a user subject study consisting of 22 users driving different vehicles over the course of several months in Urbana-Champaign, IL. The experimental results using the collected data suggest that fuel savings of 21.5 over the fastest, 11.2 percent over the shortest, and 8.4 percent over the Garmin eco routes can be achieved by following GreenGPS green routes. The study confirms that our navigation service can survive conditions of sparse deployment and at the same time achieve accurate fuel predictions and lead to significant fuel savings.

Index Terms—Application, participatory sensing, transportation, energy, navigation

1 Introduction

The proliferation of smart phones has led to increased interest in mobile participatory sensing as an important branch of mobile computing. Mobile participatory sensing relies on user devices that are on the move to obtain sensing coverage of large areas for purposes of interest to the mobile service [2], [3], [4], [5]. Early examples include mapping of physical phenomena or computing community statistics of interest [6], [7], [8], [9], [10], [11], [12], [13], [14]. An inherent challenge in such a service is therefore to handle conditions of sparse deployment, where coverage is small. Clearly, a mobile participatory sensing service must offer value to customers even when sparsely deployed. Otherwise, it may not survive to see a larger deployment. The fundamental way to improve value under conditions of sparse deployment is to develop models for generalization from sparse data. This paper describes a general approach for such generalization and applies it to the specific context of GreenGPS, a novel navigation service that finds the most *fuel-efficient* (hence, *green*) routes for drivers as opposed to the traditional shortest or fastest routes, offered by such services as Google maps [15] and MapQuest [16]. We show that we are successful at generalizing from sparse data and are able to offer value (i.e., fuel savings) in conditions of sparse deployment.

GreenGPS collects the necessary information to compute and answer queries on the most fuel-efficient route. We show that the most fuel-efficient route between two points may be different from the shortest and fastest routes. For example, a fastest route that uses a freeway may consume more fuel than the most fuel-efficient route because fuel consumption increases non-linearly with speed or because it is longer. Similarly, the shortest route that traverses busy city streets may be suboptimal because of downtown traffic.

A GreenGPS client is offered as an Android application that can be installed on participants' smart phones. The application collects data parameters involved in engine fuel consumption, vehicle speed (VSS) and location. Fuel consumption parameters are provided by the *On-Board*

- F. Saremi, H. Wang, T. Abdelzaher, S. Hu, and S. Li are with the Department of Computer Science, University of Illinois at Urbana-Champaign, IL 61801.
E-mail: {saremi1, wang44, zaher, shu17, shenli3}@illinois.edu.
- O. Fatemieh is with the Computer Software, Amazon, 1918 8th Ave (16th floor), Seattle, WA 98101. E-mail: omid.fatemi@gmail.com.
- H. Ahmadi is with Computer Software, Google, 747 6th Street South, Kirkland, WA 98033. E-mail: hahmadi@google.com.
- R. Ganti is with IBM T.J. Watson Research Center, Hawthorne, NY 10523. E-mail: rganti@us.ibm.com.
- H. Liu is with the School of Computer Science and Technology, University of Science and Technology, China. E-mail: hcliu@ustc.edu.cn.
- L. Su is with the Department of Computer Science and Engineering, State University of New York at Buffalo, NY 14260. E-mail: lusu@buffalo.edu.

Manuscript received 14 Apr. 2014; revised 13 Mar. 2015; accepted 27 Mar. 2015. Date of publication 10 Apr. 2015; date of current version 2 Feb. 2016.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2015.2421939

Diagnostic (OBD-II) interface of the vehicles, standardized in all vehicles sold in the United States since 1996. The OBD-II is a diagnostic system that monitors the health of the automobile using sensors that measure approximately 100 different engine parameters. Other examples of monitored measurements include engine RPM, coolant temperature, vehicle speed, and engine idle time. A comprehensive list of measured parameters can be obtained from standard specifications as well as manufacturers of OBD-II scanners.

There exist several commercial OBD-II scanner tools [17], [18], [19], [20], that can read and record the sensor values. Apart from such scanners, remote diagnostic systems such as GM's OnStar, BMW's ConnectedDrive, and Lexus Link are capable of monitoring the car's engine parameters from a remote location (e.g. home of driver of the car). With respect to the increase in the use of bluetooth devices (e.g., cell-phones), GreenGPS utilizes a typical OBD-II to bluetooth adaptor in conjunction with its participatory data collection framework. This enables GreenGPS to be offered at a very low price. For example, in our deployment we use ELM327 OBD-II bluetooth wireless transceiver dongle [21] which is available for less than $10 at the time of writing. The fuel consumption data, read via the adaptor, are wirelessly transmitted to the user-side hub of sensing, the phone application, upon request. The application combines the OBD-II data with other sensory data and opportunistically uploads them to an aggregation and modeling backend upon availability of WiFi Internet connectivity.

The general challenge in participatory sensing applications addressed in this paper is the sparsity of their high dimensional data space. To address the data sparsity challenge, GreenGPS exploits prediction models that enable it to extrapolate from fuel-efficiency data of some vehicles on some streets to the fuel consumption of arbitrary vehicles on arbitrary streets. The developed generalization methodology employed by GreenGPS can be adopted by a variety of other participatory sensing applications as well, where data follows discoverable models. The constructed prediction models in GreenGPS abstract vehicles and routes by a set of parameters such that fuel efficiency can be computed simply by plugging in the parameters of the right car and route.

Thanks to its generalization methodology, GreenGPS offers value even when sparsely deployed. Sparse deployment, here, refers to the deployment of the GreenGPS application, not deployment of OBD-II measurement devices (as those are abundant in modern cars). One specific instance of generalization in GreenGPS in the sparse deployment scenario is to support two types of users; members and non-members. Members are those who contribute their data to the GreenGPS repository from the OBD-II sensors described above. They have GreenGPS accounts and benefit from more accurate estimates of route fuel-efficiency, customized to the performance of their individual vehicles. Non-members can use GreenGPS to query for fuel-efficient routes as well. Since GreenGPS does not have measurements from their specific vehicles, it answers queries based on the average estimated performance for their vehicle's attributes such as make, model, year and class (or some subset thereof, as available). GreenGPS also allows members to get navigation advice on routes they had never driven before using models developed from data collected on other routes.

The motivation for GreenGPS does not need elaboration. GreenGPS users might be driven by benefits such as savings on fuel or positive impacts on the environment by reducing motor emissions such as CO_x and NO_x air poisoning gases. Further, GreenGPS equipment is very inexpensive and the entire procedure of GreenGPS operation described is performed in an end-to-end automated fashion.

A user subject study was conducted over the course of several months using 22 different cars with different drivers and a total of over 3,200 miles of data was collected for our experimental study to determine the accuracy of the prediction models. It is shown that on average fuel-savings of 21.5 percent over the fastest route, 11.2 percent over the shortest route, and 8.4 percent over the Garmin eco-route can be achieved by users.

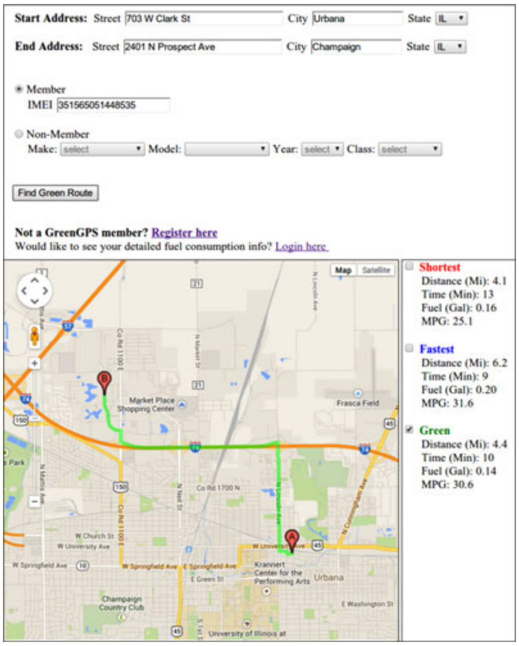
In summary, the contributions of the paper can be briefly enumerated as follows:

1) Demonstrates how to build an easy-to-deploy and inexpensive participatory sensing system to support data collection for building a fuel-saving navigation system.
2) Demonstrates how to build a general but personalizable fuel-saving navigation system using the data collected by the participatory sensing system.
3) Demonstrates how sparse samples of high-dimensional spaces can be generalized to develop models of complex nonlinear phenomena, where one size (i.e., model) does not fit all.
4) Provides an experimental performance evaluation of the developed system from vehicles driven in the area of Urbana-Champaign.

The rest of the paper is structured as follows. Section 2 presents an overview of our green navigation service. Section 3 describes the participatory sensing framework utilized for data collection. Fuel consumption modeling and model generalization are elaborated in Section 4 and Section 5, respectively. Implementation details are presented in Section 6. Then Section 7 provides evaluation of the service as how accurate the prediction models are and how much fuel savings can be achieved. Section 8 discusses our experiences with GreenGPS and lessons learned. Finally, Section 9 reviews related work and Section 10 concludes the paper.

2 The GreenGPS Approach

A study of GreenGPS reported, on average, over 16 percent fuel savings on selected routes, compared to the fastest and shortest alternative routes. To estimate the amount of savings that can be achieved on a global scale, we provide approximate calculations based on data from the Environmental Protection Agency (EPA) [22]. An estimated 200 million light vehicles (passenger cars and light trucks) are on the road in the US. Each of them is driven, on an average, 12,000 miles in a year. The average mile-per-gallon (mpg) rating for light vehicles is 20.3 mpg. Even if 10 percent of these vehicles adopted GreenGPS and 16 percent fuel savings were achieved on only 30 percent of the routes traveled by each of these vehicles, the amount of overall fuel savings is over 567 million gallons of fuel per year $((12{,}000/20.3) * (0.10 * 200M) * 0.16 * 0.30)$. This translates into over



Start Address: Street 703 W Clark St City Urbana State IL

End Address: Street 2401 N Prospect Ave City Champaign State IL

● Member
IMEI 351565051448535

○ Non-Member
Make: select Model: Year: select Class: select

Find Green Route

Not a GreenGPS member? Register here
Would like to see your detailed fuel consumption info? Login here

□ **Shortest**
Distance (Mi): 4.1
Time (Min): 13
Fuel (Gal): 0.16
MPG: 25.1

□ **Fastest**
Distance (Mi): 6.2
Time (Min): 9
Fuel (Gal): 0.20
MPG: 31.6

☑ **Green**
Distance (Mi): 4.4
Time (Min): 10
Fuel (Gal): 0.14
MPG: 30.6

Fig. 1. The user interface of GreenGPS with the most fuel-efficient route between two points for a member's vehicle.

1.6 billion dollars in savings at the pump (based on the current national average pump prices for a gallon of gasoline [23]). Authors consider the above prospective savings acceptable.

The service provided by GreenGPS is similar to a regular map application, such as Google maps [15] or MapQuest [16]. Google maps and MapQuest provide the shortest or fastest routes between two points, whereas GreenGPS computes the most fuel-efficient route. A snapshot of the Web-based GreenGPS's user interface is shown in Fig. 1 along with the most fuel efficient route between two points for a member vehicle.

Individuals who want to compute the most fuel-efficient route between two points enter the source and destination address via the interface provided by GreenGPS. Members of GreenGPS (i.e., those individuals who contributed participatory data) can register their vehicles that were used for data collection. Hence, GreenGPS can compute the route specifically for the registered vehicle. Other users may enter their vehicle's make, model, and year of manufacture. Since different vehicles have different fuel consumption characteristics, these car details are used to compute the most fuel-efficient route for the given vehicle brand.

It is impractical to assume that GreenGPS members will measure all city streets and cover all vehicle types. Instead, measurements of GreenGPS members are used to calibrate generalized fuel-efficiency *prediction models*. These models, discussed in Section 5, show that the fuel consumption on an arbitrary street can be predicted accurately from a set of *static* street parameters (e.g., the number of traffic lights, the number of stop signs, and the slope of the roads) and a set of *dynamic* street parameters (such as the average speed on the street or the average congestion level), plus the route parameters (such as the number of left turns and right turns), the vehicle parameters (e.g., weight and frontal area) and the driving behavior (e.g., making high acceleration or hard breaking). It is the mathematical model describing the relation between these general parameters and fuel-efficiency that gets estimated from participant data. Hence, the larger and more diverse is the set of participants, the better the generalized model.

For most streets, static street parameters can be obtained from traffic databases. (We show in this paper, how to estimate static parameters not in databases, such as locations of traffic lights and stop signs.) Dynamically changing parameters such as the congestion levels or average speed are more tricky to obtain. In larger cities, real-time traffic monitoring services can supply these parameters [24], [25], [15]. Many GPS device vendors, such as Garmin and TomTom, also collect and provide congestion information. In this paper, speed information is obtained from the collected data using our participatory sensing infrastructure described in the next section.

Finally, note that the increasing availability of vehicular fuel efficiency measurements to drivers in modern vehicles is not a substitute for green navigation. In order to exploit fuel efficiency measurements, a driver who wants to find a most fuel-efficient route to a given destination would have to drive on all the different alternative routes to that destination multiple times and note the average fuel consumption over a statistically significant number of trips on each route, then decide (for future reference) which route was better. In contrast, our service computes the answer automatically from a model trained using other trips on other routes that the driver already drove. This highlights the benefits of our generalization models over present affordances of modern cars.

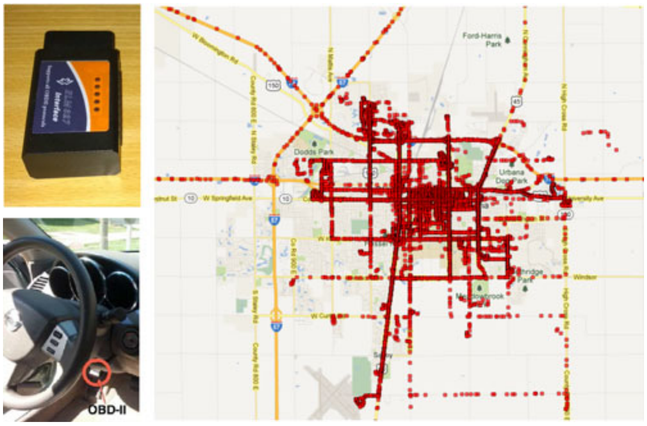
3 A Participatory Sensing System for Data Collection

In this section, we present the participatory sensing framework that we utilize for data collection and sharing. We implement a client-side interface for data collection that automatically uploads all data to a central server called the GreenGPS *aggregation server*. An individual who wishes to share their OBD-II sensor and location data simply downloads our client-side software, publicly available as an Android application on Google Play Store, and uses it to automatically upload their data to the aggregation server. The aggregation server uses the data to calibrate models that relate street and vehicle parameters to fuel-efficiency and offers the GreenGPS navigation interface for fuel-efficient routes.

Individuals who wish to contribute OBD-II data to GreenGPS, install an off-the-shelf and inexpensive OBD-II to bluetooth adapter in their vehicle (Fig. 2a). The GreenGPS phone application communicates with the vehicle OBD-II via bluetooth to obtain the engine fuel consumption data. The data is then timestamped and stored in a small database on the phone. The parameters obtained from the car and the GPS sensor on the phone include instantaneous vehicle speed, mass air flow (MAF), command equivalence ratio (EQV), engine rpm, throttle position, latitude, longitude, altitude, bearing, time and phone IMEI.

3.1 OBD-II Communication

We sample fuel parameters from the OBD-II unit using the OBD-II to bluetooth adaptor. The key parameters, namely



(a) (b)

Fig. 2. (a) Deployed OBD-II to bluetooth adaptor; (b) Coverage map for the paths on which data were collected.

mass air flow, speed, command equivalence ratio, engine rpm, and throttle position are queried in sequential order. The sequential sampling provides better overall response rate as we discovered that frequently querying the OBD-II for all the parameters (at the same time) resulted in response gaps. For example, for the majority of the vehicles, if we query for all five parameters at once, the likelihood of receiving all five responses before reaching our timeout is low. However, if we query for parameter values one at a time, the likelihood of all values being present is very high.

The sampling is ordered in the sequence described above to minimize the timing differences when calculating fuel rate and fuel economy. Since we only calculate two fields, we try to group the sampling parameters together so that the values used for fuel equations are closer in time.

a) Fuel Rate uses two queries, mass air flow and command equivalence ratio, and is calculated in gallons per second as,

$$FuelRate = \frac{MAF}{(14.7 \times EQV) \times 454.0 \times 6.17}, \tag{1}$$

wherein MAF is in grams per second, 14.7 is grams of air to 1 gram of gasoline (ideal air to fuel ratio), $|EQV| \leq 1$, 454.0 is grams per pound, and 6.17 is pounds per gallon of gasoline.

b) Fuel Economy needs three queries, MAF, EQV, and vehicle speed, and is calculated in miles per gallon (mpg) as,

$$FuelEconommy = \frac{(14.7 \times EQV) \times 454.0 \times 6.17}{MAF} \times \frac{VSS \times 0.621371}{3{,}600}, \tag{2}$$

wherein VSS is in kilometers per hour, 0.621371 is kilometers per hour to miles per hour conversion ratio, and $3{,}600$ is seconds per hour.

The engine rpm and throttle position are collected for future uses.

We try to generate samples as quickly as possible, however, the sampling rate is not constant across all vehicles. More specifically, the sampling rate varies with the OBD protocol being used, the age of the vehicle and its OBD-II unit, and the version of the OBD-II to bluetooth adaptor (newer models support higher data transfer rates), please see [21] p61 for more details.

3.2 Opportunistic Uploading

One of the design goals of the GreenGPS's participatory sensing framework was to eliminate the need for cellular data connections for data collection. This helps to avoid imposing communication overhead of data collection on users, for which they may be reluctant to use their own data plans (as opposed to the route navigation step that they experience immediate benefit return and would be willing to utilize their cellular data connections). The vehicles in our study at the University campus presented DTN-like mobility patterns. Because individual devices had a low probability of coming into contact with the wireless access points located around campus, we embraced the notion of opportunistic uploading. We begin by storing generated samples in a small database on the phone. Once our application sends its samples to the data storage server, it clears out the delivered samples to free up resources within the database. We reduced the amount of characters per transfer by replacing parameter names with numeric constants. Duplicate samples were filtered out.

3.3 Collected Data

We conducted a study involving 22 users (with different cars) over the course of several months. A total of over 3,200 miles was driven by our users to construct the initial models. Fig. 2b shows a partial map of the paths on which data was collected. The details of the car make, model, year, class, and the number of miles of data collected for each car are summarized in Table 1. The distribution for the trips distance is depicted in Fig. 3a. It can be observed that the majority of the trips are very short. In particular, about 70 percent of the trips are less than 4 miles long and the remaining 30 percent are from 4 to 10 miles long. The speed distribution for various one-mile road segments driven is plotted in Fig. 3b and represents a mixture of two normal distributions. The distribution denotes that most of the road segments are low speed (less than 45 miles per hour) and that is due to the type of streets in the town in which exist only few highways. Fig. 3c presents the average number of stop signs, traffic lights, left turns and right turns per one-mile road segments with respect to the distance of the trips. It is denoted that, as path length increases, the average number of stop signs per segment shows an overall decreasing trend while the average number of traffic lights, left turns and right turns do not exhibit such overall trend change. This is expected considering that short trips are mostly the ones driven in campus and in low speed streets that an intersection appears almost at every block.

4 Modeling

In this section, we derive the fuel consumption model structure and explain how the impact of dynamic traffic conditions on fuel consumption is modeled. We then elaborate how the required information regarding the location of traffic signs can be derived.

TABLE 1
The Average Error Percentage (Magnitude) for the Individual Car Models, the Generalized Case When All the Data Is Used to Obtain the Model, and the Cluster-Based Model Constructed Based on the Optimal Generalization Order

Car Make	Car Model	Car Year	Car Class	City MPG	Hwy MPG	Miles Driven	Individual Error %	General Error %	Cluster-based Error %
Toyota	Camry	2004	Mid-Size	24	33	80	1.55	8.44	1.72
Chevrolet	Impala	2002	Large	21	32	69	3.02	17.16	2.48
Ford	Ranger	2008	Van	15	19	29	0.89	25.26	5.26
Toyota	Corolla	2000	Compact	31	38	259	6.06	10.68	6.01
Buick	LeSabre	2002	Large	20	29	54	3.38	7.46	2.45
Ford	E-250	2011	Van	13	17	99	3.59	7.93	3.59
Toyota	Corolla	2010	Compact	26	35	53	4.31	18.47	9.32
Toyota	Celica	2001	Sub-Compact	28	34	497	4.94	11.69	4.94
Nissan	Altima	2006	Compact	24	31	95	3.83	7.04	3.83
Subaru	Impreza	2010	Sub-Compact	19	24	26	0.09	3.82	4.74
Toyota	Corolla	2004	Compact	32	40	141	3.67	13.59	3.67
Mazda	Mazda6	2003	Mid-Size	23	29	62	3.94	18.5	3.94
Audi	A4	2005	Compact	22	31	88	6.86	14.58	6.86
Toyota	Camry	2012	Mid-Size	25	35	90	4.96	7.59	4.96
Subaru	Impreza	2010	Sub-Compact	19	24	69	9.22	15.47	8.23
Hyundai	Santa-Fe	2001	Sport-Utility	21	28	87	3.3	17.92	3.3
Ford	Taurus	2002	Mid-Size	20	28	65	4.01	5.51	5.06
Mitsubishi	Eclipse	2002	Sub-Compact	23	30	184	5.32	15.91	5.32
Nissan	Altima	2010	Mid-Size	23	32	103	2.44	9.59	2.44
Mitsubishi	Galant	2002	Mid-Size	21	28	112	4.45	12.19	8.11
Toyota	Celica	2000	Compact	28	34	882	6.24	8.74	6.06
Toyota	Camry	2004	Mid-Size	24	33	57	0.73	13.76	2.21
Average Error Percentage (magnitude):							4.91	11.33	5.07

4.1 Derivation of Model Structure

The first part of data generalization is to derive a model structure.

To motivate the need for modeling, we plot the distribution of miles per gallon for all the data collected in Fig. 3d. We observe from this figure that the distribution spans a wide range of values between 2 and over 60. The standard deviation of the mpg distribution is 9.4 miles per gallon, which is pretty high. Hence, an appropriate model is needed to estimate the fuel consumption on various segments.

The difference from the models in the literature [26], [27], [28] lies in that we are interested in developing a model whose parameters can be easily measured by our participatory sensing system and later utilized in the route navigation phase. This imposes restrictions on what parameters can be used which makes it different from developing first-principle models whose goal is simply to understand the physics.

Several factors affect the fuel consumption on streets. We classify these parameters into five categories, which are (i) *static street parameters*, (ii) *dynamic street parameters*, (iii) *route parameters*, (iv) *car specific parameters*, and (v) *personal parameters*. Static street parameters model the street characteristics and do not change (or change with a very high time constant) over a period of time. For example, the speed limits of streets change much less frequently and the number of traffic lights on the street (in a given stretch) remain more or less constant. The dynamic street parameters are characteristics that change with time, for example, the congestion levels on a street or the average speed on a street. The static and dynamic street parameters together determine the fuel efficiency of a particular street. The fuel usage is also affected by the number of left turns and right turns through the route. Hence, route parameters are parameters that depend on the shape of the overall route (such as turns), as opposed to the individual street segments. Other variations in the fuel consumption can occur due to the type of car being driven and the nature of the person's driving. For example, a big SUV may consume more fuel than a small sedan or a person who is aggressive (making higher acceleration or hard braking) is likely to consume more fuel than a

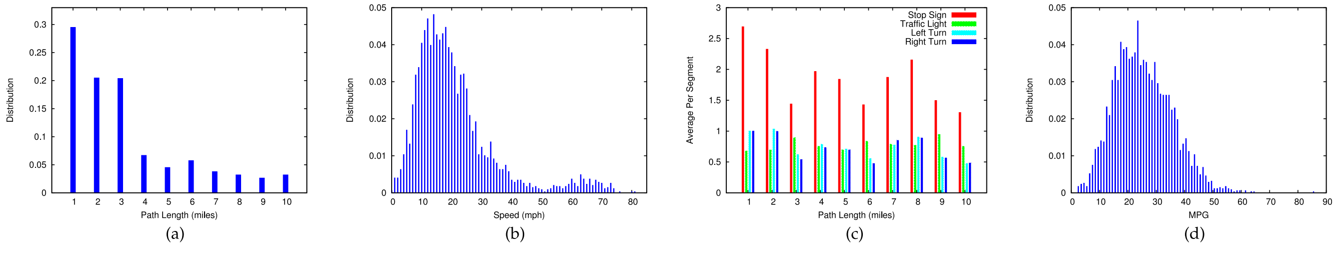


Fig. 3. The distribution of trip data collected from all cars: (a) The path distance distribution; (b) The average speed distribution; (c) The average number of stop signs, traffic lights, left turns and right turns per one-mile road segments with respect to the distance of the trips; (d) The real mpg distribution.

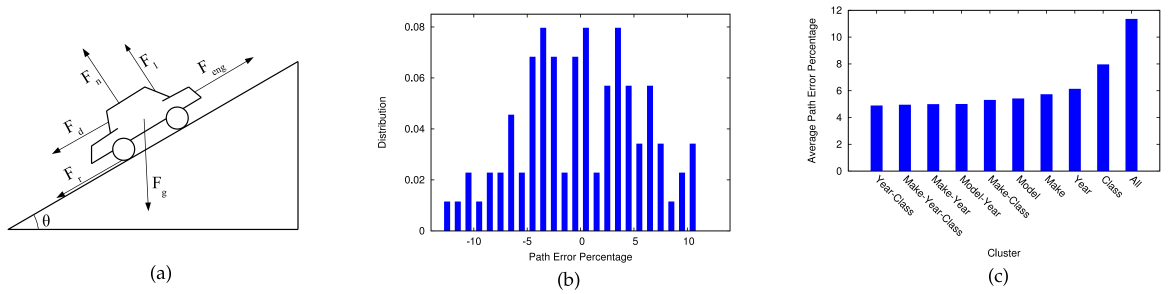


Fig. 4. (a) The free body diagram of a car; (b) Intersection approach concept and classification features; (c) The path error percentage distribution for one car; (d) Average error percentage (magnitude) of the models obtained from various clusters.

sluggish driver. These parameters account for the variation in fuel consumption due to the route parameters, the car type and the driver behavior.

The inputs to our prediction model include street segment parameters, route parameters, and car parameters. We do not consider driver factors in the model and will explore it in our future work. Note that, we are interested in predicting long-term fuel consumption only. While actual savings of a user on individual commutes to work may vary, the user might be more concerned with their net long-term savings. Hence, it is important to capture only the statistical averages of fuel consumption. As long as the errors have near zero mean, the savings are accurate in the long term. As a given user drives more segments, a value of interest is the end-to-end prediction error that results, which improves over time and represents how far we are off in our estimate of total fuel consumption.

The free body diagram of a car is given in Fig. 4a. Assuming that the car is on an upslope, the final force acting on the car is given by the following equation:

$$F_{car} = F_{eng} - F_d - F_r - F_{g_x}, \tag{3}$$

where F_{eng} is the engine force, F_d is the air resistance force (drag), F_r is the rolling resistance force, and F_{g_x} is the gravitational force acting on the car. These forces will be elaborated on in the following.

Assuming that the engine RPM is ω, the torque generated by the engine is $\tau(\omega)$, the k-th gear ratio is r_{g_k}, the differential ratio is r_d, the transmission efficiency is e_t and the radius of the tire is r, then the engine force F_{eng} is given by the following equation:

$$F_{eng} = \frac{\tau(\omega) \cdot r_{g_k} \cdot r_d \cdot e_t}{r}. \tag{4}$$

The force due to air resistance, F_d, is given by the following equation:

$$F_d = \frac{1}{2} \cdot \rho \cdot c_d \cdot A \cdot v^2. \tag{5}$$

In the above equation, ρ is the air density, c_d is the drag coefficient, A is the frontal area of the car, and v is the instantaneous speed of the car. The drag coefficient quantifies the resistance in a fluid environment (air). For example, for a streamlined body the coefficient is about 0.05, for a regular sedan is about 0.4-0.5, and for a truck could be about 1.

The rolling resistance force F_r is characterized by the instantaneous speed of the car, the normal force, and the corresponding coefficients as:

$$F_r = c_{r_1} \cdot v + c_{r_2} \cdot F_n \tag{6}$$

in which F_n is the normal force given by:

$$F_n = F_{g_y} - F_l, \tag{7}$$

wherein F_{g_y} is the gravitational force acting on the car and F_l is the lift force. The F_{g_y} is given as follows:

$$F_{g_y} = m \cdot g \cdot cos(\theta), \tag{8}$$

where m is the mass of the car, g is the gravitational acceleration, and θ is the slope of the road. The F_l is given as follows:

$$F_l = \frac{1}{2} \cdot \rho \cdot c_l \cdot A \cdot v^2. \tag{9}$$

The gravitational force due to the slope, F_{g_x}, is given by the following equation:

$$F_{g_x} = m \cdot g \cdot sin(\theta). \tag{10}$$

In order to obtain a relation between the fuel consumed and the above forces, we note that the fuel consumed is related to the power generated by the engine at any instance of time t. If f_r is the fuel rate (fuel consumption at a given time instance) and P is the instantaneous power, then $f_r \propto P$. Power is related to the torque function and engine RPM as follows:

$$P = 2 \cdot \pi \cdot \omega \cdot \tau(\omega). \tag{11}$$

Hence, we obtain,

$$f_r = \beta \cdot \omega \cdot \tau(\omega). \tag{12}$$

In the above equation, β is a constant. Further, we also have the following relationship from rotational dynamics:

$$v = r \cdot \frac{\omega}{r_{g_k} \cdot r_d}. \tag{13}$$

Substituting for ω in Equation (12) from Equation (13) and for $\tau(\omega)$ in Equation (4) from Equation (12), F_{eng} can be written as:

$$F_{eng} = \frac{e_t f_r}{\beta v}. \tag{14}$$

Subsequently, substituting Equation (14) and Equations (5) to (10) in Equation (3) gives the following:

$$\begin{aligned} F_{car} &= ma \\ &= \frac{e_t f_r}{\beta v} - \frac{1}{2}\rho c_d A v^2 - c_{r_1} v - c_{r_2} mgcos(\theta) \\ &\quad + \frac{1}{2} c_{r_2} \rho c_l A v^2 - mgsin(\theta), \end{aligned} \tag{15}$$

where a is the instantaneous acceleration of the car.

From the above equation, we obtain the fuel consumption rate as a function of the forces acting on the car shown below:

$$\begin{aligned} f_r &= k_0 mav + k_1 c_d A v^3 + k_2 v^2 + k_3 mvcos(\theta) \\ &\quad + k_4 A v^3 + k_5 mvsin(\theta), \end{aligned} \tag{16}$$

wherein $k_0, \ldots, k_5$ are constant coefficients.

In order to further derive a model that can be used for regression analysis, we will detail the various components that are part of the fuel consumption of a car. As shown in the above equation, a moving car at a constant speed on a straight road which does not encounter any stop signs, traffic lights or turns will only need to overcome the frictional forces caused by the air, the road, and gravity. These are represented by $k_1 c_d A v^3$, $k_2 v^2 + k_3 mvcos(\theta) + k_4 A v^3$, and $k_5 mvsin(\theta)$, respectively. On the other hand, the first component $k_0 mav$ can be broken down further into two components, one is the extra fuel rate due to congestion, and the second one is the extra fuel rate due to encountering stop signs (*ST*), traffic lights (*TL*), left turns (*LT*) and right turns (*RT*). Hence, the previous equation now becomes the following:

$$\begin{aligned} f_r &= k_1 c_d A v^3 + k_2 v^2 + k_3 mvcos(\theta) \\ &\quad + k_4 A v^3 + k_5 mvsin(\theta) + k_{00} mav \\ &\quad + (k_{01} + k_{02} mav)(\nu'_1 n_{ST} + \nu'_2 n_{TL} + \nu'_3 n_{LT} + \nu'_4 n_{RT}), \end{aligned} \tag{17}$$

where ν'_1, ν'_2, ν'_3 and ν'_4 are constant coefficients, n_{ST}, n_{TL}, n_{LT} and n_{RT} are the number of stop signs, traffic lights, left turns and right turns, respectively. In the above equation, the last component represents the fuel rate during the idle time and consequent acceleration when encountering traffic signals, stops and turns.

Finally, we can obtain the equation for the consumed fuel, f_c, by integrating the rate of the fuel consumption with respect to time:

$$f_c = \int_{t_{ini}}^{t_{fin}} f_r(t)\, dt \tag{18}$$

in which t_{ini} denotes the time a new trip is initiated, t_{fin} denotes the time the trip is finished.

If we assume the road gradient θ remains constant, for each road segment i replace v with $\bar{v_i}$, the segment average speed, and consider $a = dv/dt$, we can further simplify the above integral to the following equation for the purpose of regression analysis:

$$\begin{aligned} f_c &= k_1 c_d A \sum_{i=1}^{n} \bar{v_i}^2 \Delta L_i + k_2 \sum_{i=1}^{n} \bar{v_i} \Delta L_i + k_3 mLcos(\theta) \\ &\quad + k_4 A \sum_{i=1}^{n} \bar{v_i}^2 \Delta L_i + k_5 mLsin(\theta) + k_6 m(v_{fin}^2 - v_{ini}^2) \\ &\quad + k_7(\nu_1 n_{ST} + \nu_2 n_{TL} + \nu_3 n_{LT} + \nu_4 n_{RT}) \\ &\quad + k_8 m\left(\nu_1 \sum_{i=1}^{n_{ST}} \bar{v_i}^2 + \nu_2 \sum_{i=1}^{n_{TL}} \bar{v_i}^2 + \nu_3 \sum_{i=1}^{n_{LT}} \bar{v_i}^2 + \nu_4 \sum_{i=1}^{n_{RT}} \bar{v_i}^2\right), \end{aligned} \tag{19}$$

wherein $k_1, \ldots, k_8$ are regression coefficients, n is the total number of road segments along the trip, L is the trip distance, and ν_1, ν_2, ν_3 and ν_4 are constant coefficients. In the equation, $\bar{v_i}$ denotes the speed of the segment immediately following the traffic signals, stops or turns which lays on the path. Note that at the beginning of such street segment $v_{i_{ini}} = 0$ as the vehicle has come to stop at the intersection.

In Section 5.1, we show that the coefficients of our model, $k_1, \ldots, k_8$ differ among different vehicles making it harder to generalize from vehicles we have data for to those we do not.

4.2 Dynamic Traffic Conditions Modeling

Our experience reveals, not surprisingly, that the degree of traffic congestion plays the largest role in accounting for fuel consumption variations among individual trips of the same vehicle. To model the effect of dynamically changing traffic, the street segments real-time speed should be used as the speed rating in the fuel consumption model presented in equation (19). However, it should be noticed that the current speed at distant locations would become obsolete when the vehicle arrives there. Therefore, for distant areas the appropriate future traffic status should be predicted to be used in the model. Here we address such spatio-temporal parameters contributing to the model.

Let the overall speed of a street segment at location x at time t be denoted by $v_{x,t}$ and defined as:

$$v_{x,t} = \mu_{x,t} + \gamma_{x,t}, \tag{20}$$

wherein $\mu_{x,t}$ represents the speed mean value and $\gamma_{x,t}$ represents the deviation from the mean. The former, $\mu_{x,t}$, is calculated through a weighted average over the past speed values taken from traffic history for street segment located at x. In the calculation higher weights are given to the more recent speed values. The latter, $\gamma_{x,t}$, can be modeled as a stationary process with mean zero modeled using an autoregressive moving average (ARMA) model, as follows:

$$\gamma_{x,t} = \sum_{l=1}^{p} \phi_l \gamma_{x,t-l} + e_{x,t} - \sum_{l=1}^{q} \theta_l e_{x,t-l}, \tag{21}$$

where the first p terms correspond to the autoregressive terms and the last q terms correspond to the moving average terms. The coefficients $\phi_1, \ldots, \phi_p$ and $\theta_1, \ldots, \theta_q$ are the model parameters. The subscript l denotes the time lag and $t - l$ means l time units before the current time t. The $e_{x,t}$'s are independent, identically distributed random variables each with mean zero and variance σ_e^2.

However, it is evident that there exists spatial correlation in road traffic, that is, the traffic status at some street depends on that of the neighboring streets as well. In order to incorporate the spatial correlation into the model, let the spatial correlation matrix be denoted as $\Pi^{\langle\tau\rangle} = \left[\pi^{\langle\tau\rangle}_{x,x'}\right]_{N\times N}$ where $x, x' \in \{1 \cdots N\}$ and N denotes the number of street segments. The entry $\pi^{\langle\tau\rangle}_{x,x'} \in \mathbb{N}$ specifies the number of time units needed for the traffic at street segment x' to influence the traffic at x according to the average historical speed of the area. Note that $\pi^{\langle\tau\rangle}_{x,x'} = 0$ implies $x = x'$. Also that, when there is no spatial correlation between x and x' at time interval τ, $\pi^{\langle\tau\rangle}_{x,x'} = \infty$. The superscript τ will be described shortly.

The spatial correlation is then reflected in the model as follows:

$$\begin{aligned}\gamma_{x,t} = {} & \sum_{l=1}^{p}\sum_{x'=1}^{N} \phi_l \; I(\pi^{\langle\tau\rangle}_{x,x'} \le p-l+1) \; \gamma_{x',t-l} + e_{x,t} \\ & - \sum_{l=1}^{q}\sum_{x'=1}^{N} \theta_l \; I(\pi^{\langle\tau\rangle}_{x,x'} \le q-l+1) \; e_{x',t-l}.\end{aligned} \tag{22}$$

Thus, to predict the future street speed, the model expression includes not only the impact of the traffic history at the same location x, but also the effect of the traffic at nearby correlated streets as well. To make the model expression concise, let $\Gamma_t = [\gamma_{1,t} \; \cdots \; \gamma_{N,t}]^t$, $e_t = [e_{1,t} \; \cdots \; e_{N,t}]^t$, $\Upsilon_p = \big[I(\pi^{\langle\tau\rangle}_{x,x'} \le p-l+1)\big]_{N\times N}$ and $\Upsilon_q = \big[I(\pi^{\langle\tau\rangle}_{x,x'} \le q-l+1)\big]_{N\times N}$. The model can thus be rewritten as:

$$\Gamma_t = \sum_{l=1}^{p} \phi_l \; \Upsilon_p \; \Gamma_{t-l} + e_t - \sum_{l=1}^{q} \theta_l \; \Upsilon_q \; e_{t-l}. \tag{23}$$

To compute the most fuel-efficient route the speed values in equation (19) are computed as follows. The real-time speed $V_t = \mathcal{M}_t + \Gamma_t$, where $V_t = [v_{1,t} \; \cdots \; v_{N,t}]^t$ and $\mathcal{M}_t = [\mu_{1,t} \; \cdots \; \mu_{N,t}]^t$, is used for the speed of the street segments up to 5 minutes (one time unit) away from the source address. For streets $t + 5n$ to $t + 5(n+1)$ minutes away, where $n \in \{1 \cdots 11\}$, the predicted speed value $V_{t+n} = \mathcal{M}_t + \Gamma_{t+n}$ is utilized. To calculate Γ_{t+n}, $n > 1$, first the future speed Γ_{t+1} is computed through equation (23) and using the real-time speed Γ_t and the speed values from history, Γ_{t-l}. The predicted speed Γ_{t+1} is then used in the prediction of the Γ_{t+2}. The computation continues until Γ_{t+n} is calculated. Finally, for streets more than one hour away, the average historical speed $\mathcal{M}_t$ is utilized. Note that utilizing the predicted speed values the approximate time that the vehicle reaches each street segment along the path can be computed.

The computed most fuel-efficient route is updated every 5 minutes using the most recent traffic information. This calls for the speed predictions to be performed every 5 minutes, however, the spatial correlation matrix is computed once. To compute $\Pi^{\langle\tau\rangle}$, we divide the time horizon based on the time of the day and the day of the week, and then for each time period, referred to by τ, the spatial correlation matrix is computed accordingly. For example for Friday 3 pm to 8 pm $\Pi^{\langle Fri\ 3pm-8pm\rangle}$ is computed once. For holidays a separate time period can be considered.

It should be mentioned that the results reported in this paper are based on data collected in the area of Urbana-Champaign. The county is almost never congested and has very low traffic variability that renders the extensions mentioned in this section unnecessary. The approach can be used in larger cities, where savings will likely be higher than those reported in this paper due to the the larger variability in traffic conditions that could be taken advantage of, and because of the larger connectivity which offers more alternatives in the choice of route. Currently, Google maps [15], INRIX [29], Nokia Here [30], Microsoft Bing [31], MapQuest [16], PeMS [25] and 511NY [32] are traffic data providers that offer real-time and/or historical traffic information.

4.3 Detection of Traffic Signs Location

A considerable portion of fuel consumption in transportation is contributed by the traffic regulators due to the implicit non-negligible idling time and acceleration. To build accurate models leading to navigation of reliable most fuel-efficient routes the impact of these players cannot be ignored. As also invoked by equation (19) we should be able to locate traffic lights and stop signs along a route to measure its fuel efficiency. This becomes an issue as there exist no public database providing the information on the location of traffic signs. Such information is either not present at all (for some areas) or fragmented in the municipalities (mostly in the form of physical copies). On the other hand, the collection of such information would be a very time and labor expensive task. Consequently we aim at establishing an automated learning-based methodology for this purpose.

To detect the location of traffic signs we train a classifier utilizing the map information provided by *OpenStreetMap (OSM)* [33] and exploit it in modeling and navigation stages. Our designed approach follows: we describe how our required data is obtained, explain our learning approach, and present its detection accuracy.

We extract our required data from OSM which provides good coverage across the world. OSM is the equivalent of Wikipedia for maps, where data are collected from various free sources (such as the US TIGER database [34], Landsat 7 [35], and user contributed GPS data) and an editable street map of the given area is created in an XML format. The OSM map is essentially a directed graph, which is composed of three basic object types, *nodes*, *ways*, and *relations*. A node has fixed coordinates and expresses points of interest (e.g. junction of roads, Marriott hotel). A way is an ordered list of nodes with tags to specify the meaning of the way, e.g. a road, a river, a park. A relation models the relationship between objects, where each member of the relation has a specific role. Relations are used in specifying routes (e.g. bus routes, cycle routes), enforcing traffic (e.g. one way routes).

The intersections are extracted from the OSM through finding nodes present in more than one way. Afterwards some data cleaning is carried out to refine valid street intersections. The intersections are then decomposed into multiple *approaches* corresponding to the joined ways and directions. For example, a four-way intersection is decomposed into four approaches.

The collection of the intersection approaches serves as input to train our classifier. The approach features used in the training are Street Length, Street Speed, Road Type, and Distance to the Nearest Intersection. The street length denotes the total end-to-end length of the streets which intersect at the junction. The street speed is defined as the OSM assigned speed of the intersecting street segments. The road type denotes the category and importance of the road within the road network. The distance to the nearest intersection is equal to the length of the street segment between the junction and the nearest intersection on the corresponding approach.

The classification is performed based on *support vector machines (SVM)* which utilizing a non-linear mapping transform the original feature space into a higher dimensional space, resulting in better separation of the training classes with linear boundaries. The SVM is able to maximize the geometric margin while minimizing the classification error.

The classifier is provided with a training set, containing the set of intersection approaches with their features and labeled with the type of the approach traffic sign. The label could be either TL, denoting the presence of a traffic light, ST, denoting the presence of a stop sign, or $None$, denoting the absence of any traffic regulator.

To evaluate the performance of the methodology, we collected data from three different cities: Urbana, Champaign (most of the city covered), and Los Angeles (part of the city covered). This choice aimed at considering two extremes: a small campus town (Urbana and Champaign) and a large city. A total of 3,691, 2,803, and 7,561 intersection approaches were extracted for the city of Urbana, Champaign, and LA, respectively, the ground truth data for which was gathered manually through GoogleStreetView. We first considered training and testing a classifier using data from the same city. Hence, for each city dataset, we divided the data in half, one part served for training the classifier and the other part was used as the test set. It turned out that our methodology achieves 82, 83, and 84 percent accuracy in predicting whether a given intersection approach faces a stop sign, a traffic light, or neither in the cities of Urbana, Champaign, and LA, respectively.

We then evaluated the accuracy of the classifier when the training and test data are from different cities. Specifically, the dataset gathered from LA was used as training data. The trained classifier was then utilized to predict the existence of stop signs, traffic lights, or the absence thereof in the area of Urbana-Champaign. It resulted in a classification accuracy of 80 percent. The result shows that classifier training and testing does not need to use same city data. A trained classifier from LA was able to predict traffic regulators in the small college town of Urbana-Champaign almost as accurately as a classifier trained in Urbana-Champaign. This observation eliminates the need for city-by-city training. Note also that the trained classifier needs only data from OSM maps to perform the classification. This is in contrast to crowd-sensing based methods [36] that require GPS traces. OSM maps are freely available and have broad coverage worldwide.

5 Model Generalization to Predict Green Routes

In this section, we demonstrate the foundations of one of the key mechanisms in participatory sensing applications that are tolerant to conditions of sparse deployment; namely, the generalization from sparse multidimensional data. The generalization mechanism solves a key problem at a critical phase of most newly deployed systems, which makes it important. Such generalization is complicated by the fact that, in high-dimensional datasets, one size does not fit all. Hence, for example, developing a single regression model to represent all data is highly suboptimal. In the case of GreenGPS, the data contributed by users of our participatory sensing application will be a sparse sampling of routes and cars. Hence, we aim to use data collected by a smaller population to build models capable of predicting the fuel consumption characteristics of a larger population.

5.1 Model Evaluation: One Size Fits All?

Regression analysis is a standard technique for estimating coefficients of models with known structure. In this section, we demonstrate that a single regression model is a bad fit for our data. Said differently, while a regression model that accurately predicts fuel consumption can be found for each car from data of that one car, the model found from the collective data pool of all cars is not a good predictor for single vehicles. Hence, in a sparse dataset (where data is not available/sufficient for all cars) it is not trivial to generalize. We illustrate that challenge by first evaluating the performance of car models obtained from their own data (which is good), then comparing it to the trivial generalization approach: one that finds a single model based on all car data then uses it to predict fuel consumption of other cars. A solution to the challenge is presented in the next section.

We evaluate the accuracy of models derived from vehicle data according to a cross validation approach. We predict fuel consumption of a randomly chosen trip using a model trained based on data from other trips. We distinguish models based on other trips of the same car from models based on data from other cars as well in predicting the fuel consumption of the one trip. The eighth and ninth columns of Table 1 summarize the resulting errors, respectively, for the set of cars used. More specifically, to compute the error of a particular trip, the trip is removed and a model is trained based on other trips of the same car which is then utilized to predict fuel consumption for the trip. Using the collected actual fuel consumption of the trip, the relative prediction error percentage is then computed. This is repeated for all trips in the dataset. The average error percentage across all trips of the same car (i.e., the summation of all trips' absolute errors divided by the number of trips) is considered as Individual error percentage. As for the General error percentage, when training the model, the trips of other cars are included in the training dataset as well. The errors reported here are for trips from four miles up to 10 miles; the errors for shorter and longer trips will be presented later in Fig. 6.

We also plot the error distribution for individual trips (for one car) in Fig. 4b. We observe that the distribution is near normal and the mean is near zero (-0.14 percent). We observe a similar distribution for other cars too.

We also observe from Table 1 that the prediction errors of the single model computed from the data of all cars are significantly (over several times) worse than those of the models obtained from each individual car. This suggests the existence of non-trivial bias in the error of the former model that does not cancel out by aggregation. In the next section, we propose a way to mitigate this problem based on grouping cars into clusters, such that prediction can be done based on other *similar* cars by some metric of similarity.

5.2 Model Clustering

The above suggests a need for better generalization over vehicle data. Different car types behave differently. Even though the model is parameterized by factors such as car weight and frontal area, they are not enough to account for differences among cars. This is a common problem in high-dimensional datasets collected in participatory sensing applications. The question becomes, if we cannot generalize over the whole set, can we generalize over a subset of dimensions?

A solution is borrowed from the general literature on data cubes [37]. Data cubes are structures for online analytical processing (OLAP) that are widely used for multidimensional data analysis. They group data using multiple attributes and extract similarities within each group. For example, previous work showed how to efficiently construct regression models for various subsets of data [38]. The data cube framework can thus help compute the optimal generalization order in that it can help generalize data based on those dimensions that result in the minimum modeling error.

We consider four major attributes (data dimensions) of a given car: *make*, *model*, *year* and *class*.[1] Based on these four attributes, data can be grouped in 16 ways, out of which six are redundant since vehicle model specifies make and class as well. At one extreme, all cars may be grouped together, thus producing a single regression model (which we have shown is not acceptable). At the other extreme, cars can be partitioned into clusters based on their four attributes. Intermediate clusters are constructed based on a subset of these attributes. A separate model is derived for each cluster. One should note that in cluster (model, year) for example, a Camry 2004 is modeled differently from a Camry 2012 and a Civic 2004.

Between the two extremes, to find out which clustering scheme gives the best accuracy, we obtain the average percentage error for each scheme. The results, summarized in Fig. 4c, show that different generalizations have different quality. These generalizations are better than using all cars data lumped together. While our dataset is small to make general conclusions, as more data is collected in our deployed participatory sensing infrastructure (e.g., say deployment reaches 100 s of cars), progressively better generalizations can be attained. In the figure it can be observed that some of the clusters present quite similar accuracy. This behavior is induced due to limited vehicle type overlap in our dataset and the performance of the intermediate clusters is not well differentiated thereof. Specifically, these clusters end up having several single vehicle groups in common. To draw general conclusions, a further scaled vehicle set with adequate vehicle overlap with respect to the considered attributes is required.

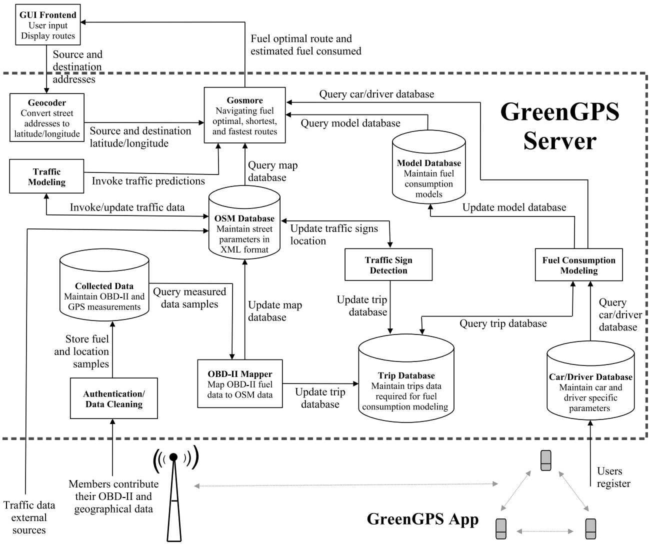

GUI Frontend
User input
Display routes
Source and destination addresses
Fuel optimal route and estimated fuel consumed
GreenGPS Server
Geocoder
Convert street addresses to latitude/longitude
Source and destination latitude/longitude
Gosmore
Navigating fuel optimal, shortest, and fastest routes
Query car/driver database
Query model database
Model Database
Maintain fuel consumption models
Traffic Modeling
Invoke traffic predictions
Query map database
Invoke/update traffic data
OSM Database
Maintain street parameters in XML format
Update traffic signs location
Update model database
Traffic Sign Detection
Fuel Consumption Modeling
Collected Data
Maintain OBD-II and GPS measurements
Query measured data samples
Update map database
Update trip database
Query trip database
Query car/driver database
Store fuel and location samples
OBD-II Mapper
Map OBD-II fuel data to OSM data
Update trip database
Trip Database
Maintain trips data required for fuel consumption modeling
Car/Driver Database
Maintain car and driver specific parameters
Authentication/ Data Cleaning
Users register
Traffic data external sources
Members contribute their OBD-II and geographical data
GreenGPS App

Fig. 5. The various modules of GreenGPS.

To use results of Fig. 4c, one would build models for each cluster shown in the Fig. 4c which has sufficient data for reliable modeling. The reader is encouraged to refer to [39] on how the reliability of a model can be inferred. To model a car, an instantiated cluster with the same attributes as the car is utilized that has the least error. If a car is encountered for which none of the clusters match the car, we have no recourse but to use the model computed from all data. That is, the clusters in Fig. 4c are traversed sequentially, from the most accurate to the least accurate, until a cluster containing sufficient data is reached. We evaluate the performance of the *Cluster-based* modeling technique by measuring how accurately an individual car can be modeled using the data from cars with similar attributes. Specifically, we construct the model cluster while removing data of a certain car trip. We use the model cluster to estimate the fuel consumption for the given car trip. This is done for all car trips. The resulting average error percentage is presented in the 10-th column of Table 1. As it can be observed from the table, the cluster-based modeling technique has led to significant accuracy improvements compared to the General model. In a few cases, such as the second vehicle in the table (Chevrolet Impala 2002 Large) the error has reduced even over the Individual model. This is because the individual vehicles involved did not collect representative enough data to generate an accurate model. Hence, improvements are achieved from grouping of this vehicle and Buick LeSabre 2002 Large into the same cluster (i.e., Year-Class) that results in reducing the errors even over the Individual model for both vehicles.

6 Implemented Green Navigation

The GreenGPS server combines several open source software services to provide the fuel-efficient route computation service. The various modules that are part of the GreenGPS implementation are depicted in Fig. 5.

[1] Other vehicle attributes can be employed as well, for example, *city mpg*, *highway mpg*, *mpg difference* (the difference between highway mpg and city mpg) and *mpg ratio* (the ratio of highway mpg to city mpg).

6.1 Data Collection

We implement the user-facing participatory sensing module as an Android application in Java that runs on users' smart phones. This application gathers fuel consumption and speed information data from the car's engine, combines that with location data gathered using phone's GPS, and opportunistically uploads the data to the backend aggregation server.

For further details about the implementation refer to Section 3.

6.2 Modeling and Generalization

The OBD-II data shared by individuals is used to compute regression models that predict the fuel consumption on specific streets given the car details (e.g. make, model, age, category). The regression variables are stored in the Trip Database, whereas the car specific variables are stored in a similar database. The modeling module queries this database to compute fuel consumption on a given way for a given car.

Each trip is organized as a row in a database where 14 of its attributes are the values of the physical model parameters in Equation (19) and are used for regression. Four other attributes (Make, Model, Year, Class) are used for grouping. After computing the regression models for all clusters , search for a specific four-tuple of (Make, Model, Year, Class) is done according to the optimal generalization order based on Fig. 4c. The first regression model that matches the query is used for prediction.

6.3 Detection of Traffic Signs Location

To implement the traffic signs location detection module, we built our SVM-based classifier using the "kernlab" package [40] in the statistical tool R. The classifier was trained using a dataset collected in part of the city of Los Angeles and used to predict the traffic signs at each intersection in the area of Urbana-Champaign, needed for evaluating the performance of the GreenGPS in Section 7.1. For details on the foundation and construction of the classifier please refer to Section 4.3.

6.4 Navigation

GreenGPS maintains the map of a given area as an OSM. Navigation is achieved in GreenGPS by customizing the open source routing software, Gosmore [41]. Gosmore is a C++ based implementation of a generic routing algorithm that provides shortest and fastest routes between two arbitrary end-points. Gosmore uses OSM XML map data for doing routing. Gosmore's routing algorithm, A*, by default computes the shortest route. This routing algorithm works on the OSM map, where the nodes of the graph are OSM nodes and the edges of the graph are OSM ways and the weights of the edges are the lengths (distance) of the ways. The fastest route is then computed by multiplying the distance by an inverse speed factor (thus giving lower weights to faster ways). Our fuel-optimal routing algorithm multiplies the distance by an inverse mpg metric that results in lower weights for fuel-optimal ways.

6.5 Graphical User Interface (GUI)

When a query is posed to GreenGPS for the fuel-optimal route between the source address and destination address provided by the user inputs, the addresses are first translated into latitude/longitude pairs using the open source geocoding perl module, Geo::Coder::US. This module is used for geocoding US addresses only. Geocoding is the process of finding corresponding latitude/longitude data given a street address, intersection, or zipcode.

After the source and destination addresses are geocoded into their corresponding latitude and longitude pairs using the geocoder module, the latitude and longitude pairs are fed to the navigation module which computes the fuel-optimal route (along with the shortest and fastest routes) using the OSM XML database and the prediction models of fuel consumption on streets (computed from the OBD-II sensor data contributed by users). The computed routes are then displayed on the GUI frontend along with the estimated fuel consumption for the given routes. The GUI frontend to display the routes (shown in Fig. 1) utilizes Microsoft Bing maps. Routes are color coded and rendered as *polylines* on Bing maps. For example, the fuel-optimal route is a "green" color polyline.

7 Experimental Evaluation

The performance of GreenGPS is evaluated in two stages. First, we evaluate performance of our model by using it to predict the end-to-end fuel consumption for long routes. Second, we evaluate the potential fuel savings of an individual using GreenGPS.

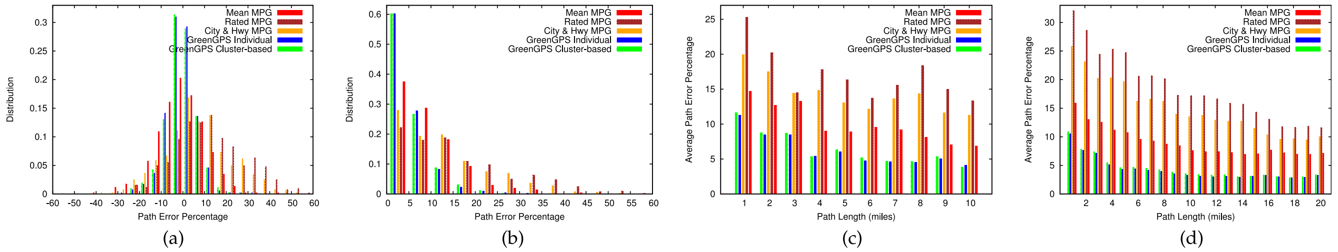
7.1 Part I: Green Navigation Model Accuracy

In this section we evaluate the accuracy of our prediction model in estimating fuel consumption on long routes. For that, the attributes contributed to each trip in our collected driving dataset in the Urbana-Champaign, called for by Equation (19), are extracted and/or computed for each corresponding path.

In the experimental evaluation, the number and location of stop signs and traffic lights along each path is predicted using our SVM-based classifier. The classifier is trained using a dataset collected from part of the city of Los Angeles (and not from Urbana or Champaign). It was tested in Urbana-Champaign to demonstrate cross-city generalizability. When testing, street features were extracted from OSM maps for each intersection then input to the classifier. Ground truth (for both training and testing) was collected using GoogleStreetView. As mentioned earlier in Section 4.3, the LA-based classifier achieved an accuracy level of 80.2 percent in predicting the existence and types of traffic regulators on the streets of Urbana-Champaign. The next question was: given the imperfect prediction of traffic regulators, what is the accuracy in predicting fuel consumption?

The accuracy of our green navigation service is measured using path-based cross validation in which the fuel consumption along one path is predicted using the models trained based on data collected along other paths. The prediction error for the path is then obtained. This is repeated for all paths.

The path error distribution corresponding to the above experiment when prediction for each car is done based on data of the same car (on other paths) is shown in Fig. 6a as "GreenGPS Individual". We observe that the path error



(a) (b) (c) (d)

Fig. 6. Distribution of path error percentage for different prediction models: (a) signed error, (b) unsigned error. Mean path error percentage for different prediction models when path length is varied: using (c) original data, (d) synthetic data.

distribution is nearly normal and that the mean of this distribution is near zero (-0.28 percent).

We conduct a similar experiment to derive the path error distribution that is achieved by employing Cluster-based training such that fuel consumption of a car trip is predicted from the model trained based on trips of other cars in the nearest cluster as well, as described in Section 5.2. The prediction error for each path is computed as before and the distribution is presented in the figure as "GreenGPS Cluster-based". Again, a normal distribution of the path errors is observed with near zero mean (-0.25 percent).

In order to compare the accuracy of our technique, three other fuel prediction approaches are evaluated in Fig. 6a in which mpg values are the basis of the prediction. In these approaches the fuel consumption along a path is estimated using:

$$f_c^{mpg} = \frac{L}{MPG} \tag{24}$$

in which L is the length of the path and MPG is the mpg of the car. In *Mean MPG* approach, the MPG is the average mpg computed from data of the car. In *Rated MPG* approach, the MPG is computed as the average of rated city mpg and rated highway mpg for the car. In the last approach, *City & Hwy MPG*, for each individual road segment along a path, depending on the road segment type either city mpg or highway mpg is used for fuel prediction.

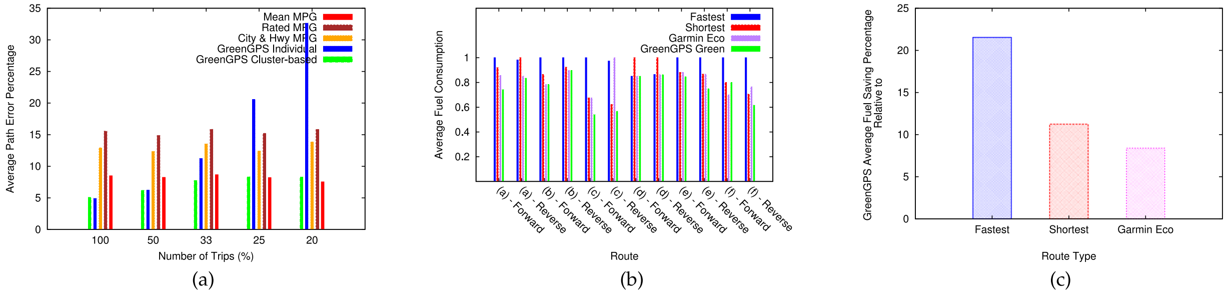
In order to compare the approaches more clearly, the distribution of the corresponding unsigned error is shown in Fig. 6b. As depicted in the figure, GreenGPS approach outperforms the other prediction methods. It is observed in the figure that GreenGPS Individual and Cluster-based training approaches differ only slightly in accuracy. The reason lies behind the lack of overlap among car types in our vehicle set. As a result, for most of the cars the nearest cluster in Cluster-based training becomes a cluster with one single car—the car for which prediction accuracy is being calculated. Therefore it should be emphasized that these two approaches may significantly differ from each other for a different dataset; this is explained later in Fig. 7a.

It is worth noticing that, as expected, the Mean MPG approach beats the other mpg-based approaches in Fig. 6b. This is because the Mean MPG approach uses the collected data to compute cars' mpgs as opposed to considering a predetermined fixed constant.

In order to understand how path errors vary with path lengths, we bin the paths based on their length and compute the average of the absolute path errors as a function of path length. We repeat this experiment for the case where models are derived for each car individually and the case where models are derived for clusters and the nearest cluster is used. We plot the mean of the absolute path errors for varying path lengths in Fig. 6c.

We observe from Fig. 6c that the error decreases with increasing path length for both GreenGPS and mpg-based approaches, which is what we want. In order to show the performance of these approaches for longer routes beyond 10 miles, the trips in our original dataset are concatenated to form longer trips. We concatenate every up to ten chronologically consecutive trips (time-stamped based on start and finish time) together and form longer trips. The features of the new trips (such as distance and the number of traffic regulators) are computed based on those of the original constituting trips. We then added the new longer trips to the original set of trips. Fig. 6d presents the accuracy results on the new dataset. As expected, the decreasing trend of the prediction errors continues for trips beyond 10 miles long as well. The average percentage error for the dataset is 4.74 percent and for trips longer than four and ten miles is 3.67 and 3.08 percent, respectively.

We have not explored if the progressively improving accuracy of the approaches with respect to the trip distance



(a) (b) (c)

Fig. 7. (a) Impact of the amount of training data on different prediction models accuracy; (b) Average normalized fuel consumption for the various trips between different landmarks; (c) Percentage fuel saved by using GreenGPS green routes, relative to the Fastest, Shortest, and Garmin Eco routes.

holds true when the commutes have large dynamics in speeds, such as in larger cities. The current dataset is limited in that it was collected in a fairly quiet town.

The accuracy of our approach depends on the amount of training data. Fig. 7a presents the impact of the training dataset size on the performance of fuel prediction approaches. The 100 percent point denotes using the whole dataset, 50 percent denotes using half of the dataset, and so on. The dataset down-scaling was performed in an alternate manner on the set of all chronologically ordered trips that were grouped based on the contributing vehicles. For example, for the 50 percent dataset size, one out of every two consecutive trips in the list was selected, for the 33 percent dataset size, one out of every three consecutive trips was selected, and so on and so forth.

As depicted in the figure, as the training dataset becomes quite small, the GreenGPS Individual training becomes inaccurate. This is while the accuracy of the Cluster-based approach slightly decreases and it significantly outperforms Individual training approach for small datasets. Hence as the dataset becomes smaller, the performance gap between the Individual training and the Cluster-based training increases. At the same time, the accuracy of the mpg-based approaches remains nearly constant. This suggests to adopt an mpg-based approach at the very beginning of the deployment phase (when there is no or very limited data collected) and then shift to GreenGPS train-based approach as sufficient data for constructing reliable models is collected. The figure also depicts the GreenGPS potential for further increase in precision (compared to the results presented in this paper) through collection of more driving data.

From the perspective of building participatory sensing applications, the above suggests the importance of finding models that do not have *biased error*. Since the models often try to predict aggregate or long-term behavior (such as long term exposure to pollutants, annual cost of energy consumption, eventual weight-loss on a given diet, etc.), if the error in day-by-day predictions is normally distributed with zero mean, the long-term estimates will remain accurate. Hence, rather than worrying about exact models, GreenGPS attempts to find *unbiased* models, which is easier.

7.2 Part II: Fuel Savings in Urbana-Champaign

In this section, we evaluate the fuel savings achieved when using the GreenGPS system. To evaluate fuel savings, we chose landmarks in the city of Urbana-Champaign that are regularly visited in our commutes, such as library, the university health center, stadium, frequently visited restaurants and parks, and shopping complexes. Then the shortest, the fastest, the Garmin eco-route, and the GreenGPS green routes were looked up for each pair of landmarks. Each person selected two pairs of landmarks and for each of which drove twenty round trips (of approximately 15-35 minutes each): five on the shortest route, five on the fastest route, five on the Garmin eco-route, and five on the GreenGPS green route. The actual fuel consumption for each trip was recorded. The landmarks together with the shortest, fastest, Garmin eco, and GreenGPS green routes are shown in Fig. 8. The routes for the trips in the opposite direction (i.e., driving from point B to point A) are very similar to the ones presented in the figures for forward direction and are thus omitted.

We observe from Fig. 8 that the fuel-optimal route for the source-destination pair in the b, c, and e were similar to the shortest route and in the d it was the fastest route, whereas, in the a and f the fuel-optimal route was neither the shortest, nor the fastest. Hence, picking the shortest or fastest routes consistently is not optimal.

The average fuel consumption for the trips in the experiment are shown in Fig. 7b. It can be observed that the GreenGPS, except for the trip $(f) - Forward$, consistently finds the most fuel-efficient route. To confirm that the differences in fuel consumption between the compared routes are not due to measurement noise, we tested the statistical significance of the difference in means using the two-way ANOVA. The test yielded that the differences are statistically significant with a confidence level of 95 percent.

The average fuel saving percentage achieved by following the GreenGPS green routes as opposed to the fastest, the shortest, and the Garmin eco routes is presented in Fig. 7c. The results report that the GreenGPS routes can lead to fuel savings of on average, 21.5 percent over the fastest routes, 11.2 percent over the shortest routes, and 8.4 percent over the Garmin eco routes. Although only a handful of routes were used in the experiments above, it nevertheless shows promise as a proof of concept.

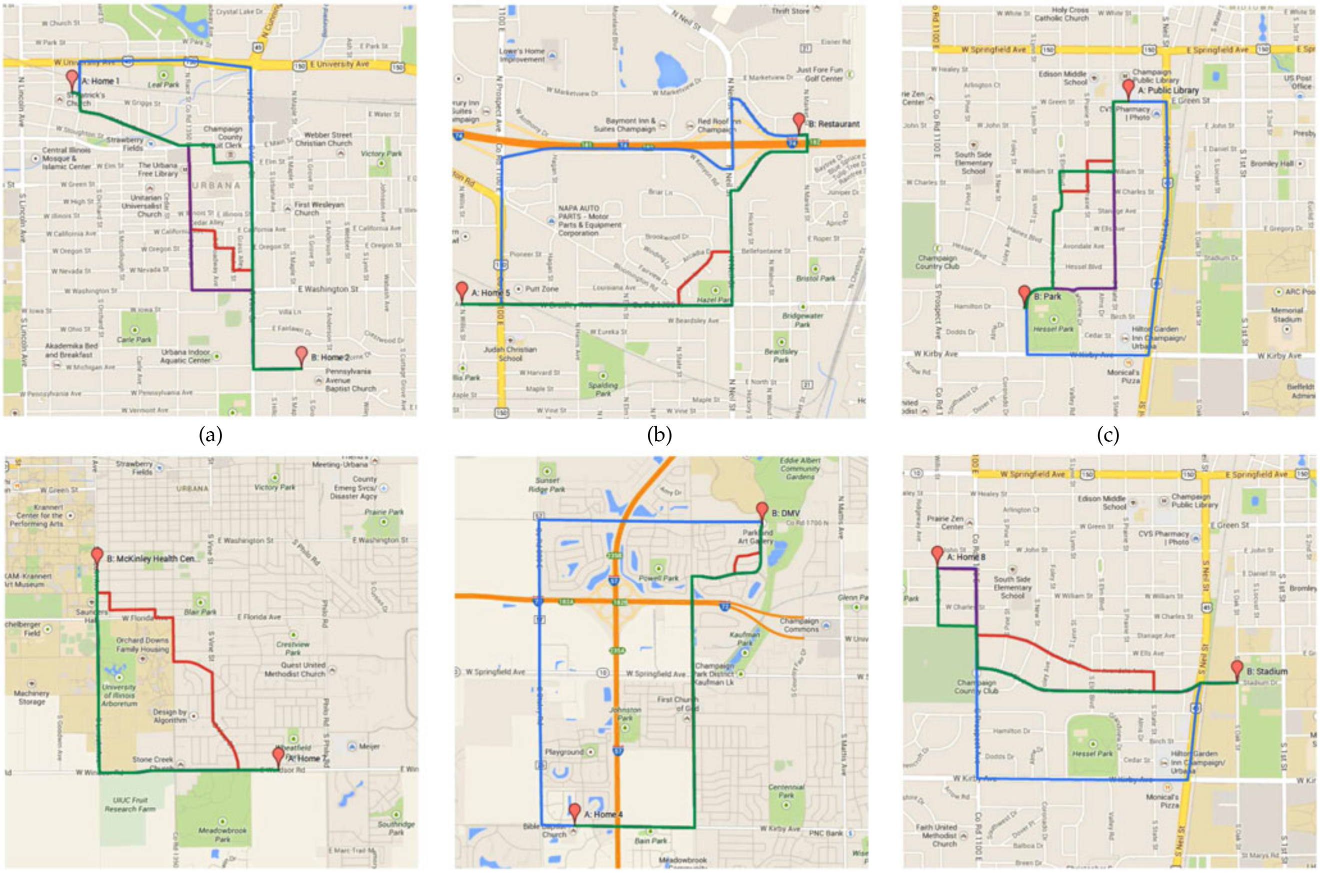
8 Discussion

This section presents a brief discussion of lessons learned and experiences with the GreenGPS service and its components, as a participatory sensing application using a mobile platform.

Data cleaning. We observed that data cleaning is an important problem and it is application dependent. We had several occasions when several fields were missing from the data (e.g., some OBD parameters were empty due to timing subtleties). A simple scheme was used to filter complete datasets from those that were missing values.

Heterogeneity. An application-specific challenge was observed due to the variations in the OBD-II standards among different cars. It was experienced that some car manufacturers use non-standard OBD-II parameter identifiers (PIDs). A few such examples we encountered in our initial deployment include Honda Civic 2004, Honda Accord 2005 and General Motors Sonoma 2002. As a result we had to discard data from those vehicles due to missing fuel parameters. This suggests that participatory sensing applications involve a large number of heterogeneous components (e.g., different car types in GreenGPS) that one should take into account and resolve before scaled deployments.

Slow start. A major hurdle in getting participatory sensing systems off the ground is to provide the right incentives to the individuals (who are part of the system) [42]. We believe that the initial deployment, which tends to be sparse, should be carefully designed in order to provide incentives for larger adoption. It should therefore be useful from the very early stages. The very low price of the GreenGPS was one of our main design targets in order to incentivize users to adopt the service. In addition, at the



(a) (b) (c)

(d) (e) (f)

Fig. 8. The landmarks and the corresponding shortest (in red), fastest (in blue), Garmin eco (in purple), and GreenGPS green (in green) routes: (a, b): Toyota Camry 2004; (c, d): Nissan Altima 2006; (e, f): Toyota Corolla 2000.

early deploy stage collected data may not be sufficient for building reliable models for some cars. Instead, an mpg-based prediction approach is employed. As further related data is collected and probabilistic guarantees on constructing reliable models are provided, the Individual or Cluster-based training approach is utilized. This ensures one that even at the very early stages GreenGPS would not lead to lower savings compared to available baselines employed by commercial products such as Garmin.

Utility of generalization. The utility of the generalization methodology described in this paper is not compromized by the increasing prevalence of fuel-efficiency measurements in modern cars. This is because modern cars measure fuel efficiency on routes they traverse. Cars do not predict fuel efficiency before route traversal. Hence, the only way drivers can compare gas consumption on different routes at present would be to drive all of them and compare results. In contrast, GreenGPS predicts the final answer without the driving. The contribution of this paper is thus complementary to (and *not* subsumed by) affordances offered in modern vehicles.

Privacy. In participatory sensing systems, privacy challenges come to the forefront. A large class of participatory sensing systems monitor location information continuously, which poses significant privacy issues. Simple anonymization of data will not work in such situations, as the GPS traces can lead to privacy breaches (e.g., reveal the home location of the user and thus uncover their identity). Techniques such as the one proposed in [43] can be used to preserve privacy, while still allowing accurate modeling. In [43], measurement samples are first integrated into, so called, *segments* in order to remove correlation. The uncorrelated segments are then converted into some *neutral features* appropriate to be used in modeling the phenomena (vehicles fuel consumption) while preserving the users privacy. The privacy preserving methodology has been applied to our green navigation service as a case study in the paper. In our current study, individual users simply switch off data collection application when they feel the need for privacy. The latter is simple and fast, however, the participatory sensing service employing it may be permitted for gathering data only intermittently. Nevertheless, the former approach and data perturbation-based approaches such as [44] and [45] enable perpetual privacy-preserving data collection for a reasonable extra computation cost.

Long term investment. As expected, the main factors affecting fuel consumption of a vehicle on a path are the average speed, the speed variability (estimated by averaging the speed squared), and the engine idle time (estimated from the number of stop signs, traffic lights and turns on the path). Rather than exploring the use of real-time traffic conditions, we opted to use statistical averages of speed, speed variability and idle time. It is easy to see how such statistical averages can be computed for different hours of the day and different days of the week given a sufficient amount of historical data, yielding expected fuel consumption (in the statistical sense of expectation). The outcome is that individual trips may differ significantly from the statistical expectation. However, by consistently following routes that have a lower expected fuel consumption, savings will accumulate in the long term. Drivers may think of GreenGPS as a long-term investment. Short-term results may vary, but long-term expectations should tend to come true.

One should add that our evaluation is not intended to be a definitive study on vehicular fuel consumption. For example, we evaluate fuel consumption in Urbana-Champaign only, which is quite flat. Hence, $\theta = 0$ is a good approximation. Furthermore, the range of cars used in the study is rather skewed towards sedans, and hence not representative of the diversity of cars on the streets. Fortunately, even this rather homogeneous dataset was sufficient to show that the generalization challenge is hard.

With the above caveats, we believe that the study remains of interest in that it explores problems typical to many participatory sensing applications, such as overcoming conditions of sparse deployment, adjusting to heterogeneity, and living with large day-to-day errors towards estimating cumulative properties. The GreenGPS study could therefore serve as an example of what to expect in building similar services, as well as a recipe for some of the solutions.

9 Related Work

Previous work in participatory sensing and transportation fuel saving are relevant and reviewed below.

9.1 Participatory Sensing

Our navigation service is an example of participatory sensing services, that have recently become popular in networked sensing. The concept of participatory sensing was introduced in [2]. In participatory sensing, individuals are tasked with data collection which is then shared for a common purpose. A broad overview of such applications is provided in [3], [4], [5]. Several early applications include CenWits [6], a participatory sensing network to search and rescue hikers, CarTel [7], a vehicular sensor network for traffic monitoring, CabSense [8], a cabs sensor network to find best corners to catch taxis depending on the current location and time, BikeNet [9], a bikers sensor network for monitoring popular cyclist routes, and ImageScape [46], a cellphone camera network for sharing diet related images. Some more recent applications include Micro-Blog [10], a content-sharing platform, PEIR [11], a report system enabling individuals to measure and compare their impact on environment as well as their exposure to environmental emissions, Escort [12], an electronic escort system that enables localizing people, MoVi [13], a service for covering social events, and [14], a service to determine buses arrival time. Our application, GreenGPS, introduces a novel example of this genre that enables individuals to compute fuel efficient routes customized for their vehicles.

9.2 Fuel Saving in Transportation

There exist a body of work addressing transportation fuel consumption factors to achieve savings in the field. A comprehensive study that provides optimal route choices for lowest fuel consumption is presented in [47]. In the paper, fuel consumption measurements are made through the extensive deployment of sensing devices (different from the OBD-II) in experimental cars. In contrast, our service uses a sparse deployment to build mathematical models for predicting fuel consumption for other streets and cars.

UbiGreen [48] is a mobile tool that tracks an individual's personal transportation and provides feedback regarding their CO_2 emissions. Chen et al. [49] proposes to exploit information on surrounding vehicles and road conditions in designing eco-driving systems to achieve higher fuel-saving. Social drive [50] is a crowdsourcing service that provides feedback to drivers regarding their driving behavior and enables them to share their real-time trip information through social networks, stimulating users to reduce their gas consumption. CarMA [51] provides high-level abstractions for sensing and tuning automobile engine parameters to achieve fuel efficiency. The tuning can be done at the granularity of individual trips.

SignalGuru [52], a participatory sensing based system, is a traffic signal schedule advisory application that assists drivers to adjust speed and avoid coming to a complete stop. The authors in [53] propose a mechanism based on communication between traffic light signals and approaching vehicles in which a fuel-optimal speed is computed and sent to the vehicles to reduce fuel consumption. Qiao et al [54] proposes to use RFID-based e-stop signs at unsignalized intersections to alter drivers behavior properly early and achieve potential emissions reduction and fuel-economy improvement.

There exist a large category of works, such as VTrack [55], that collect real-time traffic information and provide estimations on road travel times in order to aid users route around traffic congestion, being a major cause of excess fuel consumption. VTrack utilizes WiFi and GPS sensors of smart phones to perform localization in an energy-aware fashion. Kyun [56] develops a networked sensors based real-time traffic queue monitoring system for developing countries which can lead to improved automatic traffic signals scheduling in order to reduce fuel inefficiency. Some other works like PhonePark [57] approach reduction of vehicles gas consumption by detection of available on-street parking spaces which enables users to minimize their travel distance searching for parking. PhonePark uses the GPS and accelerometer sensors of travelers mobile phones.

coRide [58], among others, proposes the use of carpooling to share rides and reduce gas consumption. coRide service adopts a fare model that incentivizes both drivers and passengers to participate. In a separate study [59], it was shown that rising obesity has a significant impact on the total fuel consumption in the US. Models were developed that studied the impact of obesity on the amount of fuel consumed in passenger vehicles.

In contrast, GreenGPS represents a participatory sensing service that aims at improving fuel consumption through green routing. Using sparse data collected from volunteer participants, models are built and continuously updated that enable vehicle customized navigation on the minimum-fuel route for both members and non-members of the service.

10 Conclusions

We presented GreenGPS, an end-to-end automated participatory sensing navigation service that finds fuel-efficient routes. GreenGPS is offered as a phone application and can be easily deployed and used by individuals. The required data is collected from the engine OBD-II of members' vehicles and processed on the backend server in an end-to-end automated manner. GreenGPS enables users including non-members to acquire fuel-efficient routes customized for their vehicles between any arbitrary end-points. To survive conditions of sparse deployment, GreenGPS exploits a sparse data generalization technique from data mining literature to construct reliable fuel prediction models. A moderate-sized user subject study was conducted in Urbana-Champaign and data on users daily commutes was collected and used to train our fuel consumption models and evaluate efficiency of our green navigation service. Lessons were presented that extrapolate from experiences with our deployed service to broad issues with participatory sensing service design in general. Our experimental results show that significant fuel savings can be achieved by using GreenGPS, which not only reduces the cost of fuel, but also has a positive impact on the environment by reducing engine emissions of air poisoning gases. Importantly, the results demonstrate the feasibility of generalization from sparse deployment data.

Acknowledgments

This research was sponsored in part by IBM Research and NSF Grants CNS 10-59294, CNS 10-40380 and CNS 13-45266. Fatemeh Saremi is the corresponding author.

References

[1] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "Greengps: A participatory sensing fuel-efficient maps application," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Serv.*, 2010, pp. 151–164.

[2] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," presented at the Workshop World-Sensor-Web, co-Located ACM SenSys, Boulder, CO, USA, 2006.

[3] T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich, "Mobiscopes for human spaces," *IEEE Pervasive Comput.*, vol. 6, no. 2, pp. 20–29, Apr.–Jun. 2007.

[4] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G.-S. Ahn, "The rise of people-centric sensing," *IEEE Internet Comput.*, vol. 12, no. 4, pp. 12–21, Jul./Aug. 2008.

[5] M. Srivastava, T. Abdelzaher, and B. Szymanski, "Human-centric sensing," *Philos. Trans. Roy. Soc. A: Math., Phys. Eng. Sci.*, vol. 370, no. 1958, pp. 176–197, 2012.

[6] J.-H. Huang, S. Amjad, and S. Mishra, "Cenwits: A Sensor-based loosely coupled search and rescue system using witnesses," in *Proc. 3rd Int. Conf. Embedded Netw. Sens. Syst.*, 2005, pp. 180–191.

[7] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "Cartel: A distributed mobile sensor computing system," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst.*, 2006, pp. 125–138.

[8] Sense networks, Cab sense [Online]. Available: http://www.cabsense.com, Mar. 2015.

[9] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "The bikenet mobile sensing system for cyclist experience mapping," in *Proc. 5th Int. conf. Embedded Netw. Sens. Syst.*, 2007, pp. 87–101.

[10] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt, "Micro-blog: Sharing and querying content through mobile phones and social participation," in *Proc. 6th Int. Conf. Mobile Syst., Appl., Serv.*, 2008, pp. 174–186.

[11] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "Peir, the personal environmental impact report, as a platform for participatory sensing systems research," in *Proc. 7th Int. Conf. Mobile Syst., Appl., Serv.*, 2009, pp. 55–68.

[12] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury, "Did you see bob?: Human localization using mobile phones," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 149–160.

[13] X. Bao and R. Roy Choudhury, "Movi: Mobile phone based video highlights via collaborative sensing," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Serv.*, 2010, pp. 357–370.

[14] P. Zhou, Y. Zheng, and M. Li, "How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Services*, 2012, pp. 379–392.

[15] Google Maps [Online]. Available: http://maps.google.com, Mar. 2015.

[16] MapQuest [Online]. Available: http://www.mapquest.com, Mar. 2015.

[17] Actron, Elite autoscanner [Online]. Available: http://www.actron.com/product_category.php?id=249, Mar. 2015.

[18] Auterra, Dashdyno [Online]. Available: http://www.auterraweb.com/ dashdynoseries.html, Mar. 2015.

[19] AutoTap, Autotap reader [Online]. Available: http://www.autotap.com/products.asp, Mar. 2015.

[20] AutoXRay, Ez-scan [Online]. Available: http://www.autoxray.com/product_category.php?id=338, Mar. 2015.

[21] ELM, Elm327 [Online]. Available: http://elmelectronics.com/DSheets/ELM327DS.pdf, Mar. 2015.

[22] EPA. (2005). Emission facts: Greenhouse gas emissions from a typical passenger vehicle [Online]. Available: http://www.epa.gov/OMS/climate/420f05004.htm

[23] AAA. National average gas prices [Online]. Available: http://www.fuelgaugereport.com/, Mar. 2015.

[24] Traffic, Real-time traffic conditions [Online]. Available: http://www.traffic.com/, Mar. 2015.

[25] California Department of Transportation, Pems [Online]. Available: http://pems.dot.ca.gov, Mar. 2015.

[26] D. M. Bevly, R. Sheridan, and J. C. Gerdes, "Integrating INS sensors with GPS velocity measurements for continuous estimation of vehicle sideslip and tire cornering stiffness," in *Proc. Am. Control Conf.*, 2001, vol. 1, pp. 25–30.

[27] J. Farrelly and P. Wellstead, "Estimation of vehicle lateral velocity," in *Proc. IEEE Int. Conf. Control Appl.*, 1996, pp. 552–557.

[28] H. E. Tseng, "Dynamic estimation of road bank angle," *Vehicle Syst. Dyn.*, vol. 36, no. 4/5, pp. 307–328, 2001.

[29] INRIX [Online]. Available: http://www.inrix.com/, Mar. 2015.

[30] Nokia Here [Online]. Available: http://developer.here.com/, Mar. 2015.

[31] Microsoft Bing [Online]. Available: http://msdn.microsoft.com/en-us/library/hh441725.aspx, Mar. 2015.

[32] 511NY [Online]. Available: http://511ny.org, Mar. 2015.

[33] OpenStreetMap [Online]. Available: http://wiki.openstreetmap.org/, Mar. 2015.

[34] US Census Bureau, Tiger database [Online]. Available: http://www.census.gov/geo/www/tiger/, Mar. 2015.

[35] National Aeronautics and Space Administration (NASA), Landsat data [Online]. Available: http://landsat.gsfc.nasa.gov/data/, Mar. 2015.

[36] D. Wang, T. Abdelzaher, L. Kaplan, R. Ganti, S. Hu, and H. Liu, "Exploitation of physical constraints for reliable social sensing," in *Proc. 34th IEEE Real-Time Syst. Symp.*, 2013, pp. 212–223.

[37] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh, "Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals," *Data Min. Knowledge Discov.*, vol. 1, no. 1, pp. 29–53, 1997.

[38] Y. Chen, G. Dong, J. Han, J. Pei, B. W. Wah, and J. Wang, "Regression cubes with lossless compression and aggregation," in *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 12, pp. 1585–1599, Dec. 2006.

[39] H. Ahmadi, T. Abdelzaher, J. Han, N. Pham, and R. K. Ganti, "The sparse regression cube: A reliable modeling technique for open cyber-physical systems," in *Proc. IEEE/ACM Int. Conf. Cyber-Physical Syst.*, 2011, pp. 87–96.

[40] A. Karatzoglou, A. Smola, K. Hornik, and A. Zeileis (2004). kernlab—an S4 package for kernel methods in R. *J. Stat. Softw.* [Online]. *11(9)*, pp. 1–20 Available: http://www.jstatsoft.org/v11/i09/

[41] Nic Roets, Gosmore [Online]. Available: http://wiki.openstreetmap.org/wiki/Gosmore, Mar. 2015.

[42] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment framework for participatory sensing data collections," in *Proc. 8th Int. Conf. Pervasive Comput.*, 2010, pp. 138–155.

[43] H. Ahmadi, N. Pham, R. Ganti, T. Abdelzaher, S. Nath, and J. Han, "Privacy-aware regression modeling of participatory sensing data," in *Proc. 8th ACM Conf. Embedded Netw. Sensor Syst.*, 2010, pp. 99–112.

[44] R. K. Ganti, N. Pham, Y.-E. Tsai, and T. F. Abdelzaher, "Poolview: Stream privacy for grassroots participatory sensing," in *Proc. 6th ACM Conf. Embedded Netw. Sens. Syst.*, 2008, pp. 281–294.

[45] N. Pham, R. K. Ganti, Y. S. Uddin, S. Nath, and T. Abdelzaher, "Privacy-preserving reconstruction of multidimensional data maps in vehicular participatory sensing," in *Proc. 7th Eur. Conf. Wireless Sens. Netw.*, 2010, pp. 114–130.

[46] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen, "Image browsing, processing, and clustering for participatory sensing: Lessons from a DietSense prototype," in *Proc. 4th Workshop Embedded Netw. Sens.*. 2007, pp. 13–17.

[47] E. Ericsson, H. Larsson, and K. Brundell-Freij, "Optimizing route choice for lowest fuel consumption–potential effects of a new driver support tool," *Transportation Res. Part C: Emerging Technol.*, vol. 14, no. 6, pp. 369–383, 2006.

[48] J. Froehlich, T. Dillahunt, P. Klasnja, J. Mankoff, S. Consolvo, B. Harrison, and J. A. Landay, "Ubigreen: Investigating a mobile tool for tracking and supporting green transportation habits," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2009, pp. 1043–1052.

[49] Y. Chen, D. Zhang, and K. Li, "Enhanced eco-driving system based on v2x communication," in *Proc. 15th Int. IEEE Conf. Intell. Transportation Syst.*, 2012, pp. 200–205.

[50] X. Hu, V. Leung, K. G. Li, E. Kong, H. Zhang, N. S. Surendrakumar, and P. TalebiFard, "Social drive: A crowdsourcing-based vehicular social networking system for green transportation," in *Proc. 3rd ACM Int. Symp. Des. Anal. Intell. Veh. Netw. Appl.*, 2013, pp. 85–92.

[51] T. Flach, N. Mishra, L. Pedrosa, C. Riesz, and R. Govindan, "CarMA: Towards personalized automotive tuning," in *Proc. 9th ACM Conf. Embedded Netw. Sens. Syst.*, 2011, pp. 135–148.

[52] E. Koukoumidis, M. Martonosi, and L.-S. Peh, "Leveraging smartphone cameras for collaborative road advisories," *IEEE Trans. Mobile Comput.*, vol. 11, no. 5, pp. 707–723, May 2012.

[53] M. Alsabaan, K. Naik, T. Khalifa, and A. Nayak, "Optimization of fuel cost and emissions with vehicular networks at traffic intersections," in *Proc. 15th Int. IEEE Conf. Intell. Transportation Syst.*, 2012, pp. 613–619.

[54] F. Qiao, J. Wang, X. Wang, J. Jia, and L. Yu, "A rfid based e-stop sign and its impacts to vehicle emissions," in *Proc. 15th Int. IEEE Conf. Intell. Transportation Syst.*, 2012, pp. 206–211.

[55] 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 *Proc. 7th ACM Conf. Embedded Netw. Sens. Syst.*, 2009, pp. 85–98.

[56] R. Sen, A. Maurya, B. Raman, R. Mehta, R. Kalyanaraman, N. Vankadhara, S. Roy, and P. Sharma, "Kyun queue: A sensor network system to monitor road traffic queues," in *Proc. 10th ACM Conf. Embedded Netw. Sens. Syst.*, 2012, pp. 127–140.

[57] B. Xu, O. Wolfson, J. Yang, L. Stenneth, P. S. Yu, and P. C. Nelson, "Real-time street parking availability estimation," in Proc. *IEEE 14th Int. Conf. Mobile Data Manag.*, 2013, vol. 1, pp. 16–25.

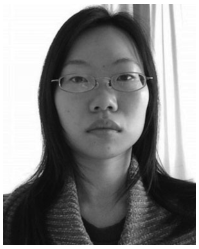
[58] D. Zhang, Y. Li, F. Zhang, M. Lu, Y. Liu, and T. He, "coRide: Carpool service with a win-win fare model for large-scale taxicab networks," in *Proc. 11th ACM Conf. Embedded Netw. Sens. Syst.*, 2013, pp. 1–14.

[59] S. H. Jacobson and L. A. McLay, "The economic impact of obesity on automobile fuel consumption," *Eng. Economist*, vol. 51, no. 4, pp. 307–323, 2006.

Fatemeh Saremi received the BS and MS degrees in computer engineering from Sharif University of Technology, Iran. She is currently working toward the PhD degree in the Department of Computer Science, University of Illinois at Urbana-Champaign. Her research interests include distributed systems and mobile computing, wireless sensor networks, and cyber-physical systems.

Omid Fatemieh received the PhD degree on secure data aggregation in distributed systems from the Department of Computer Science, University of Illinois, Urbana-Champaign, in 2011. He is a software engineer in Amazon's AWS cloud services, where he is focused on addressing networking and scalability challenges in building highly available and large-scale distributed systems. Prior to that, he was a technical program manager at Microsoft's Windows Kernel team.

Hossein Ahmadi received the PhD degree from the Department of Computer Science, University of Illinois, Urbana-Champaign, in 2011. He is a software engineer at Google working on large-scale data analysis and processing systems. His research is mainly focused on high-performance data transfer and distributed data processing algorithms.



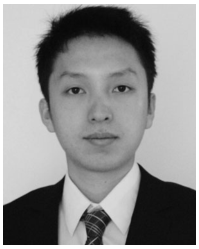
Hongyan Wang received the BS degree in computer science from the University of Illinois at Urbana-Champaign. She enjoys architecting large, complex systems in Unix-like environments.

Tarek Abdelzaher received the PhD degree from the University of Michigan, Ann Arbor, in 1999, under Professor Kang Shin. He was an assistant professor at the University of Virginia from August 1999 to August 2005. He then joined the University of Illinois at Urbana-Champaign as an associate professor with tenure, where he became a full professor in 2011. His interests lie primarily in systems, including operating systems, networking, sensor networks, distributed systems, and embedded real-time systems. He is especially interested in developing theory, architectural support, and computing abstractions for predictability in software systems, motivated by the increasing software complexity and the growing sources of non-determinism. Applications range from sensor networks to large-scale server farms, and from transportation systems to medicine.

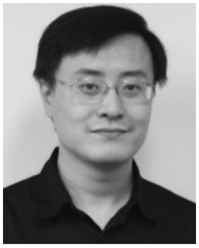
Raghu Ganti received the MS and PhD degrees from the Department of Computer Science, University of Illinois at Urbana-Champaign. He is a research staff member at IBM T.J. Watson Research Center, Yorktown Heights. His research interests span wireless sensor networks, privacy, and data mining. He is currently dabbling with spatiotemporal analytics—the analysis of moving objects and has been developing various algorithms for spatiotemporally enabling IBM's big data products.

Hengchang Liu received the PhD degree from the University of Virginia in 2011, under professor John Stankovic. He is an assistant professor at USTC. His research interests mainly include cyber-physical systems, mobile systems, named data networking, and wireless (sensor) networks.

Shaohan Hu received the MS degree from the Department of Computer Science, Dartmouth College. He is currently working toward the PhD degree. He is a research assistant at the Department of Computer Science, University of Illinois, Urbana-Champaign. His main research interests lie in designing and building mobile sensing and computing systems.



Shen Li received the BS degree in 2010 from the Computer Science Department at Nanjing University, China. He is currently working toward the PhD degree in the Department of Computer Science, University of Illinois at Urbana-Champaign. His research interests fall in green computing, cloud computing, and data centers.



Lu Su received the MS degree in statistics and the PhD degree in computer science both from the University of Illinois at Urbana-Champaign, in 2012 and 2013, respectively. He is an assistant professor in the Department of Computer Science and Engineering, SUNY Buffalo. His research focuses on the general areas of cyber-physical systems, wireless and sensor networks, and mobile computing. He was also in IBM T.J. Watson Research Center and National Center for Supercomputing Applications. He is a member of the ACM and IEEE.