Transportation Mode Detection using Mobile Phones and GIS Information

Leon Stenneth, Ouri Wolfson, Philip S. Yu, Bo Xu
University of Illinois, Chicago
Department of Computer Science
851 South Morgan, Chicago, IL 60607-7053
{lstennet, wolfson, psyu, boxu} @cs.uic.edu

ABSTRACT
The transportation mode such as walking, cycling or on a train denotes an important characteristic of the mobile user's context. In this paper, we propose an approach to inferring a user's mode of transportation based on the GPS sensor on her mobile device and knowledge of the underlying transportation network. The transportation network information considered includes real time bus locations, spatial rail and spatial bus stop information. We identify and derive the relevant features related to transportation network information to improve classification effectiveness. This approach can achieve over 93.5% accuracy for inferring various transportation modes including: car, bus, aboveground train, walking, bike, and stationary. Our approach improves the accuracy of detection by 17% in comparison with the GPS only approach, and 9% in comparison with GPS with GIS models. The proposed approach is the first to distinguish between motorized transportation modes such as bus, car and aboveground train with such high accuracy. Additionally, if a user is travelling by bus, we provide further information about which particular bus the user is riding. Five different inference models including Bayesian Net, Decision Tree, Random Forest, Naïve Bayesian and Multilayer Perceptron, are tested in the experiments. The final classification system is deployed and available to the public.

Categories and Subject Descriptors
I.5.2 [**Pattern Recognition**]: Design Methodology – *classifier design and evaluation*

General Terms
Algorithms, Design, Experimentation

Keywords
GIS, GPS, mobile phones, pattern recognition, context awareness.

1. INTRODUCTION
In ubiquitous and context aware computing, understanding the mobility of a client from sensor data is an important area of research. The transportation mode, such as walking, cycling, or train denotes some characteristics of the mobile user's context. With knowledge of a traveler's transportation mode, targeted and customized advertisements may be sent to the traveler's device. For example, if we discover that Alice is driving by car, the system may send her gas coupons or vehicle service specials.

Another motivation for transportation mode detection is transportation surveys. Travel demand surveys have taken multiple formats, such as telephone interviews and questionnaires. These data collection strategies rely on manual labeling of data after the trip, and thus, inaccuracies are introduced. For example, a traveler may not recall the exact time that she/he boarded a transportation mode. Using GPS devices is more reliable for reporting accurate location, trip time, and trip duration [12, 13, 14]. Hence, if the precise transportation modes of individual users are recognized, it is possible to provide a more realistic travel demand picture.

Many GPS trace sharing social networks have been implemented [21, 22, 23, 24]. These social networks enable friends to upload and share their GPS traces. Knowledge of transportation mode, added to these GPS traces, will enable the users to reflect on their past motion more meaningfully. It also allows users to obtain additional information from their friends' travel experience. Additionally, awareness of transportation mode of a user may help to determine the user's carbon footprint, or track the amount of calories burnt. Another application of transportation mode detection is crowd-sourced real-time traffic information in which traffic speeds are aggregated from probes such as mobile phones carried by travelers. Transportation mode detection enables the aggregation system to filter out the speed data submitted by non-motorized travelers or travelers on trains.

Transportation mode detection has been documented in the literature [1, 2, 4, 15, 16]. The existing approaches share the following general principle. First, from historical data, build a classification model in terms of mobility patterns. Then, when the transportation mode is to be determined, collect input from mobility sensors and feed the input to the classification model. The state-of-the-art is the technology developed in [4] which fuses input from GPS receiver and accelerometer. However, [4] only distinguishes between walking, running, biking, and motorized transport. It (i.e. [4]) does not distinguish between various modes under motorized transport, such as driving versus taking a bus. As shown in [4], using only GPS information reduces detection accuracy, compared to using both GPS and accelerometer information. Clearly, the accuracy of transportation mode detection may be higher if one utilizes more sensors. However, the objective of our work is to determine the added value of the transportation network data. Specifically, we consider adding to GPS data the real time locations of buses, spatial polylines representing rail line routes, and bus stop locations.

In this paper we propose a method that is able to distinguish not only between non-motorized transport and motorized transport, but also between various motorized modes including automobile, bus, and aboveground train. Additionally, if we determine that the transportation mode is bus, we further provide information on which particular bus the client is travelling on.

We follow the general principle of sensor data fusion and classification that has been used in prior work [1, 2, 4, 6, 15, 16]. Fusing GPS sensor data with external transportation network data makes transportation mode detection more robust. Intuitively, different transportation modes have different mobility patterns. For example, motorized transport generally has a higher speed than non-motorized transport. For another example, being constrained by a road, people driving a car or taking a bus cannot change their heading direction as flexibly as if they are walking or cycling. On the other hand, relying on a single type of input does not always work. For example, movement at 7 km/hr may be a brisk walk, or a slowly moving car or bus, in congestion.

Distinguishing between motorized and non-motorized transportation mode is not a difficult problem. However, with multiple motorized transportation modes, the problem becomes more difficult since buses, cars and trains may have similar GPS or accelerometer readings. We show that using a transportation network with real time and static spatial data, we can obtain high detection accuracy for various motorized and non-motorized transportation modes.

In summary, this paper is the first to address transportation mode detection using external transportation network data such as real time bus locations; this is in addition to mobile device sensor-information used in traditional approaches to the problem. Our contributions are as follows: (1) In addition to the traditional features on average speed and average acceleration, we identify for the first time the features of average bus closeness, average rail closeness, and average candidate bus closeness as the most effective features related to transportation mode detection, (2) The proposed work is the first to distinguish between motorized modes (bus, car, train) with such high accuracy, (3) There are other works that distinguish between cars and buses [1, 2, 15, 16]; however, the proposed approach is the first to consider aboveground train as a transportation mode, (4) We introduce a zip-code based indexing and pruning technique to speed up the feature computation, and (5) We present simulation results and real world results, showing the efficiency of the proposed approach.

The rest of this paper is organized as follows. In section 2 we discuss the relevant work. In section 3 we introduce the data model and the general idea of our mode detection algorithm. In section 4 we describe the system architecture and introduce the transportation network data. In section 5 we present the selection of mode detection features. In section 6 we evaluate our algorithm using real data. In section 7 we conclude the paper.

2. RELEVANT WORK

The work of Zheng et al. [2, 15] is based on transportation mode detection from GPS data alone; the authors introduce a robust and novel set of machine learning features that are sensitive to certain traffic and weather scenarios. Our work is different in that we consider transportation network data such as the real time location of buses to build classification features. Additionally, [2, 15] do not consider train as a transport mode. The approach proposed in this paper is over 17 % more accurate than [2, 15].

In [1, 16], the authors use an unsupervised learning technique to detect the transportation mode of a traveler. The transportation modes that are detected in [1, 16] include buses, cars and walk. The work in [1, 16] is able to predict the traveler's goals, such as trip destination and trip purpose. In addition to GPS and GIS data, [1, 16] use historical information about the user. Historical information includes, past user trips and information about where the users parked their cars. In our approach we do not consider historical information about the user. Furthermore, we use a supervised learning mechanism to detect transportation modes from the set {WALK, BUS, DRIVING, TRAIN, STATIONARY, BIKE}. Another difference is that we use different transportation network data than [1, 16] do. Particularly we use real time bus locations, rail line spatial data, and bus stop spatial data. [1, 16] use historical information about the bus stops at which a user boards, and where the user parks her/his vehicle. Importantly, the proposed bus stop feature is different than that in [1, 16]; the proposed classification feature captures the number of bus stops and duration at bus stops. A weakness of [1,16] is that the users' motion pattern such as where the user parks her/his vehicle daily are taken into consideration, and therefore the model relies on background information about the user. The accuracy of the proposed approach is higher than that of [1, 16] by 9%.

The proposed approach uses a single sensor (i.e. GPS) on the mobile device. There have been studies that consider multiple sensors for transportation mode recognition [4, 17, 18, 29, 30]. In [17, 18], over 20 sensors that are wearable on the human body are used. The input to the classification model includes information on the user's body condition such as temperature, heart rate and GPS position. We consider a smaller number of sensors, but add transportation network data. We believe that it is unlikely for normal users to carry over 20 sensors daily. [29] uses multiple accelerometers and [30] uses a single sensing unit with multiple sensors (accelerometer, audio, and barometer) for activity detection. The state of the art is [4] which uses GPS and accelerometer sensors for transportation mode detection. However, [4] does not distinguish between motorized transportation modes such as car and bus. This limitation is due to the similarity in features of these two modes of transportation, especially in traffic or extreme weather. Using GPS and GIS data, as shown in the proposed approach, can achieve a very high detection accuracy, as in [4]. However, in the proposed approach we distinguish between motorized transportation modes and we do not use accelerometer as in [4]. Figure 1 summarizes the related works that uses GPS.

Figure 1 – Related work with GPS sensor

	Classes	Sensor	Duration of test data	Users	Accuracy
[2]	driving, bus, bike, walk	GPS	10 months	65	76.2%
[4]	still, walk, run, bike, motor	GPS, accelerometer	50 days	16	93.6%
[1,16]	walk, bus, car	GPS,GIS	60 days	1	84%
[15]	car, bus, bike, walk	GPS	6 months	45	74%

The work in [19] is purely based on GSM, whereas we use GPS. In [37], the only sensor considered is the triaxial accelerometer. In [14], the authors' objective is to conserve mobile devices resources such as battery life. Thus, in [14] only critical location points are submitted. Furthermore, the set of classification features used in our work is different from [14].

Our prior research in [6] has a different focus; it considers extracting the semantic location from outdoor positioning systems. Likewise, [20] learns and recognizes the places a mobile user visited by observing the Wi-Fi and GSM radio fingerprints. This work does not consider Wi-Fi or GSM information. Instead, we consider GPS and transportation network data. Transportation network data is available freely to the public in many cities [25, 26, 28].

3. PRELIMINARIES

In this section, we discuss the data model and the general idea of our algorithm.

3.1 Data Model

Definition 1. *GPS sensor report.* A sensor GPS report p_i represents data submitted from the GPS sensor embedded on a traveler's mobile device. The format of the report is <*lat*, *lon*, *t*, *v*, *h*, *acc*> where: *lat* represents the latitude; *lon* represents longitude; *t* represents the timestamp of the sensor report; *v* represents the current ground speed of the device; *h* represents the direction of travel; and *acc* represents the accuracy level of the latitude and longitude coordinates.

The measurement units of the GPS sensor report attributes are: latitude (*lat*) and longitude (*lon*) are in decimal degree; current ground speed (*v*) is measured in meters per second; direction of travel (*h*) is specified in degrees counting clockwise from true north; accuracy level (*acc*) is defined in meters; and time *t* is in seconds.

Definition 2. A *GPS trace T* is a sequence of GPS sensor reports, $T = p_0 \rightarrow p_1 \rightarrow \cdot\cdot\cdot \rightarrow p_k$, where the timestamps in the sequence strictly increase.

3.2 General Idea

In general, our algorithm is a supervised learning mechanism with two stages. In stage 1 (learning stage), the data from the GPS sensor report is merged with the transportation network data and labeled ground truth. This data is used to create a classification feature set that we use to train our classification model. In this stage, mobile devices submit GPS sensor reports every *t* seconds, where *t* is a system parameter. These incoming sensor reports are labeled with the corresponding transportation modes.

Then, in stage 2 (inference stage), to determine a traveler's transportation mode, we first extract the same classification features as in stage 1. Subsequently, given the features, the classification system predicts the transportation mode of the traveler in a probabilistic format.

Specifically, our mode detection algorithm fuses inputs from the mobile devices' GPS receivers with real time locations of buses, rail line and bus stop location data. GPS technology is a built-in feature of many mobile devices, such as IPhones, BlackBerrys and Android phones. Given a GPS trace of a traveler, one way to build the classification model is as follows. For each GPS sensor report in the trace, various features including the closest Euclidian distance to rail lines, closest Euclidian distance to buses and closest Euclidian distance to bus stops are computed. Mean speed, heading, and acceleration are also obtained over a time window. These features form a sensor feature vector. The feature vector, plus the transportation mode label of the associated time interval, forms a training example. In this way, a training set is constructed. This procedure is illustrated by Figure 2.

GPS location, heading, speed and acceleration are features that have been used in existing studies. However, features like closest Euclidian distance to rail line, closest Euclidian distance to buses and bus stop closeness rate have never been used before. These are newly introduced in this paper.

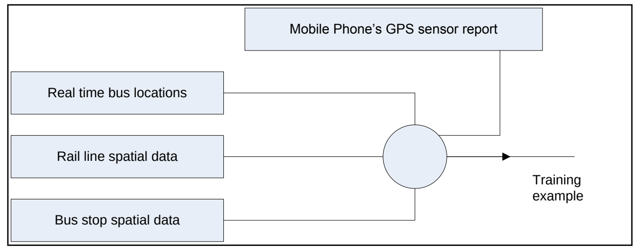


Figure 2 – Generating classification examples from GPS sensor and transportation network data

4. TRANSPORTATION MODE DETECTION

In this section, we present the system architecture and introduce the transportation network data that is utilized in our mode detection algorithm.

4.1 System Architecture

We use a centralized system architecture. Each mobile device submits its GPS sensor reports to the central server. After the central server receives a time window amount of GPS sensor reports, it predicts the transportation mode used, and sends this prediction to the mobile device. We believe this centralized system is more platform independent than the distributed counterparts, where classification is done directly on the mobile device. Furthermore, since the mode detection is performed at the central server, there is no need to store transportation network data on the mobile device. Hence, the centralized model consumes less of the device's memory, less processing time, less bandwidth, and less battery power. On the other hand, the distributed model is location privacy aware, since the location of the user is not submitted to a central authority. The privacy issue with the centralized system is addressed in our prior works [35, 36].

4.2 Transportation Network Data

We fuse data from the GPS sensor reports with data from the transportation network to create the classification feature vector. Specifically, for the city of Chicago, Illinois, USA, we consider: (1) real time location of public passenger buses, (2) rail line spatial information, and (3) public passenger bus stop spatial data. In Section 5, we will discuss the procedure to create the classification features from GPS sensor reports and transportation network information.

4.2.1 Real time bus location

Real time locations of public passenger buses for the city of Chicago is available to the public [26]. Each of these buses has a GPS receiver and can determine its location, and then report the location back to some server. Likewise, real time public transit tracking is performed in many other cities such as London, New York, San Francisco, Toronto, and Washington. For the city of Chicago, the system considers the real time locations of buses belonging to Chicago Transit Authority (CTA). CTA has over 1,700 buses in service that operates over 140 routes. On an average weekday, 1.7 million rides are taken [25]. The real time locations of the buses are updated every 20-30 seconds, and the data is available freely to the public as an API in XML format. Information available about the CTA buses includes: route, latitude, longitude, final stop, bus identification, and direction. In Figure 3, we create a "MashUp" using Google Maps [27] and the real time locations of the CTA buses.

4.2.2 Rail Lines

Spatial data of the rail tracks (**train routes**) in the city of Chicago is also available to the public [28], as geometric polylines. Figure 4 depicts a diagram of the Chicago Transit Authority (CTA) rail network. This rail line trajectory's location information is used in the proposed system. Spatial rail line information fused with GPS sensor data creates a classification feature in the proposed system. The classification feature is the Euclidian distance between the traveler's mobile device and closest rail line.

4.2.3 Bus Stops

The CTA services over 11,577 bus stops [25]. Spatial information, name, and identification of these public passenger bus stops are available [25]. Public passenger bus stops' spatial information, merged with GPS sensor data from the traveler's device, creates a classification feature in the proposed system.

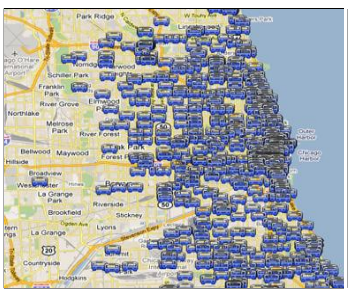


Figure 3 – Real-time bus locations

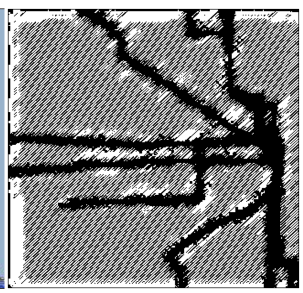


Figure 4 – Rail Network

5. CLASSIFICATION FEATURE SELECTION

This section deliberates the classification features used in the proposed transportation mode classification system. Additionally, the motivation for each feature, and the algorithm used for calculating the feature values are discussed. In this paper we explore 4 novel classification features related to motorized transportations: (1) average bus location closeness, (2) candidate bus location closeness, (3) average rail line trajectory closeness, and (4) bus stop closeness rate. This is in addition to the traditional features considered in the literature: average accuracy of GPS coordinates, average speed, average heading change, and average acceleration. In the rest of this section we describe each of the novel features and the traditional features..

5.1 Average accuracy of GPS coordinates

The estimated horizontal accuracy is a measure of the confidence on the location reported by the GPS sensor; it is a component of a GPS sensor report (see Definition 1). The accuracy is reported in meters. Different transportation modes should have different estimated accuracies. For example, traveling by aboveground trains should have worse accuracy than walking, since walking has a clearer view of the GPS satellites in the sky. Additionally, we consider the average accuracy for a set of GPS reports as a feature, instead of the instantaneous accuracy as done in [14]. Taking the average accuracy is more realistic since the GPS system may introduce uncertainties.

Let $\{p_1, p_2, p_3, p_4 \ldots p_n\}$ be a finite set of GPS sensor reports submitted from the traveler's mobile device within a time window.

$$\textbf{Average accuracy} = \left(\textstyle\sum_{i=1 \text{ to } n} p_i^{acc}\right) / n \qquad (1)$$

where p_i^{acc} is the estimated accuracy of the reported GPS position.

5.2 Average speed

In terms of speed, we use the speed value returned by the GPS sensor when it is available; this is more accurate than calculating the speed from consecutive GPS location points [4]. Otherwise, if the direct speed is not available, it can be computed from consecutive location changes. For a sequence of GPS reports we compute the average speed. This feature has been used in many existing works [2, 4, 18].

Let $\{p_1, p_2, p_3, p_4 \ldots p_n\}$ be a finite set of GPS sensor reports submitted within a time window.

$$\textbf{Average speed} = \left(\textstyle\sum_{i=1 \text{ to } n} p_i^{v}\right) / n \qquad (2)$$

where p_i^v is the current ground speed obtained from the GPS sensor report.

5.3 Average heading change

The heading is the direction from true north. For a set of GPS reports, we compute the average heading change. The heading change is an important feature for distinguishing between motorized and non-motorized transportation mode as observed by Zheng et al. [2]. This proposed classification feature is different from the heading feature in [2] because we compute the average heading change whereas [2] computes the heading change rate. The heading change rate in [2] is defined to be the number of times the heading change exceeds a certain threshold. It is computed as the ratio $|P_c|$ / distance, where $|P_c|$ represents the number of points where the traveler changes heading beyond the heading threshold. The heading change rate as defined in [2] cannot be used to distinguish between transportation modes with heading change rate below the chosen heading threshold. Let $\{p_1, p_2, p_3, p_4 \ldots p_n\}$ represent a finite set of GPS reports submitted within a time window.

$$\textbf{Average heading change} = \left(\textstyle\sum_{i=1 \text{ to } n} |p_i^{h} - p_{i-1}^{h}|\right) / n \qquad (3)$$

$$\forall\, 2 \le i \le n$$

where p_i^h is the direction from true north included in the GPS sensor report.

5.4 Average acceleration

Let $\{p_1, p_2, p_3, p_4 \ldots p_n\}$ be the finite set of GPS reports submitted within a time window.

$$p_i^{acceleration} = (p^{v}_{i} - p^{v}_{i-1})/(p^{t}_{i} - p^{t}_{i-1}),\ \forall\, 2 \le i \le n$$

$$\textbf{Average acceleration} = \left(\textstyle\sum_{i=1 \text{ to } n} p_i^{acceleration}\right) / n \qquad (4)$$

where $p_i^{acceleration}$ is the acceleration of the mobile device.

5.5 Bus location closeness

This feature aggregates the traveler's GPS location with the real time locations of public passenger buses. Bus location closeness is useful for determining if the mobile device is on a bus or not. We develop two algorithms to determine if a mobile user is traveling by bus; (1) *average bus closeness*, and (2) *candidate bus closeness*.

Let the location of the mobile user at time t be represented as p_t^{loc}, based on the GPS sensor report. Also, let the m buses in the city be bus_1 to bus_m, where $bus_{x.t}^{loc}$ is the location of bus bus_x at time t. Below, line 1 shows the mobile user's location trace. Line 2 to line 5 represent the location traces of all the m buses (bus_1, bus_2, $bus_3 \ldots bus_m$).

1. p_1^{loc}, p_2^{loc}, p_3^{loc}, $p_4^{loc} \ldots p_n^{loc}$
2. $bus_{1.1}^{loc}$, $bus_{1.2}^{loc}$, $bus_{1.3}^{loc}$, $bus_{1.4}^{loc} \ldots bus_{1.n}^{loc}$
3. $bus_{2.1}^{loc}$, $bus_{2.2}^{loc}$, $bus_{2.3}^{loc}$, $bus_{2.4}^{loc} \ldots bus_{2.n}^{loc}$
4. $bus_{3.1}^{loc}$, $bus_{3.2}^{loc}$, $bus_{3.3}^{loc}$, $bus_{3.4}^{loc} \ldots bus_{3.n}^{loc}$

   …

5. $bus_{m.1}^{loc}$, $bus_{m.2}^{loc}$, $bus_{m.3}^{loc}$, $bus_{m.4}^{loc} \ldots bus_{m.n}^{loc}$

Average bus closeness (ABC)

From GPS sensor reports $\{p_1, p_2, p_3, p_4 \ldots p_n\}$ that are submitted within a time window, we obtain the set of locations points $\{p_1^{loc}, p_2^{loc}, p_3^{loc}, p_4^{loc} \ldots p_n^{loc}\}$. For each location point p_t^{loc}, we compute d_t^{bus} as the Euclidian distance between p_t^{loc} and the closest bus $bus_{x.t}^{loc}$ at time t. Subsequently, given d_t^{bus}, we calculate the feature *average bus closeness* (ABC), as the average Euclidian distance of $(d_1^{bus}, d_2^{bus}, d_3^{bus}, d_4^{bus} \ldots d_n^{bus})$, for the set of GPS sensor reports $\{p_1, p_2, p_3, p_4 \ldots p_n\}$.

$$\mathbf{ABC} = (\textstyle\sum_{i=1 \text{ to } n} d_i^{bus}) / n \qquad (5)$$

This feature is used to capture whether the traveler is traveling via bus transportation mode.

Candidate bus closeness (CBC)

First, we obtain the set of locations points $\{p_1^{loc}, p_2^{loc}, p_3^{loc}, p_4^{loc} \ldots p_n^{loc}\}$ from GPS sensor reports $\{p_1, p_2, p_3, p_4 \ldots p_n\}$ that are submitted within a time window. For each location point p_t^{loc}, we compute the Euclidian distance $d_{j.t}^{bus}$ $1 \le j \le m$ to each bus bus_j, in the set of all buses $\{bus_1, bus_2, bus_3 \ldots bus_m\}$ at time t. Then, for each bus bus_j, we compute the total Euclidian distance D_j over the time window as follows.

$$D_j = \textstyle\sum_{t=1 \text{ to } n} d_{j.t}^{bus} \quad 1 \le j \le m \qquad (6)$$

Given D_j for all the m buses, we compute CBC as follows.

$$\mathbf{CBC} = \min (D_j) \quad 1 \le j \le m \qquad (7)$$

The classification feature CBC is the minimum D_j value. Using the CBC feature requires more memory than the ABC counterpart, since the Euclidian distance from the device to every bus in the city needs to be computed and stored for each GPS sensor report. For ABC we only compute the distance to the closest bus. To the best of our knowledge, this work is the first to consider the real time location of buses for transportation mode detection.

5.6 Rail line trajectory closeness

This classification feature relates the traveler's GPS location with spatial data representing the rail network. This feature may be useful to detect if a person is travelling on an aboveground train. For underground trains (subways), since GPS does not work well underground, this feature may not be effective. We do not consider subways in this work. The Euclidian distance d_i^{rail} between a person's mobile device and the closest rail line is computed for each GPS sensor report p_i. We then calculate the classification feature average rail location closeness (ARLC) as follows. Let $\{p_1, p_2, p_3, p_4 \ldots p_n\}$ be a finite the set of GPS reports submitted within a time window.

$$\mathbf{ARLC} = \textstyle\sum_{i=1 \text{ to } n} d_i^{rail} \; / \; |n| \qquad (8)$$

To the best of our knowledge, the proposed work is the first to use this rail line feature for transportation mode detection. The predictive power of this feature on transportation mode detection is evaluated in Section 6.

5.7 Bus stop closeness rate

This classification feature relates the traveler's GPS location with spatial bus stop data. First, from experiments we determine a *bus stop closeness threshold*. This threshold is a Euclidian distance measure and may be used to concur if a person is at a bus stop. We calculate the classification feature BSCR (*bus stop closeness rate*) as follows. Let $\{p_1, p_2, p_3, p_4 \ldots p_n\}$ be a finite set of GPS sensor reports submitted within a time window. BSCR stands for the number of GPS sensor reports p_i, whose Euclidian distance $d_i^{busstop}$ to the closest bus stop, is less than *the bus stop closeness threshold* within a unit time.

$$\mathbf{BSCR} = |\,PS\,| \,/\, \text{window size} \qquad (9)$$

where $PS = \{p_i \mid p_i \,\varepsilon\, \{p_1, p_2, p_3, p_4 \ldots p_n\},\ d_i^{busstop} < \textit{bus stop closeness threshold}\}$.

Experiments to determine bus stop closeness threshold value

Below, we explain how to obtain the *bus stop closeness threshold* value. For experiments, a traveler carried a mobile device and boarded a CTA bus. We then measured the Euclidian distance to the closest bus stop from the traveler's device. From over 450 GPS sensor reports, we plot the graph of Figure 5. The vertical axis represents the Euclidian distance to the closest bus stop. The horizontal axis represents the corresponding GPS sensor report number. From Figure 5, we observe how the mobile device's Euclidian distance to the closest bus stop fluctuates during the travel. When the traveler on a bus is at the bus stop, the distance is at a minimum. As the bus moves away from the bus stop, the distance increases. It peaks at the midpoint between two bus stops. Afterward, it decreases and reaches a new minimum when the bus reaches the next bus stop.

When the traveler is traveling via bus mode and at a bus stop, the Euclidian distance to the closest bus stop is less than 13 meters. Thus, for *bus stop closeness threshold*, we used a value of 13m.

For BSCR, we compute the number of times the Euclidian distance to the closest goes below the *bus stop closeness threshold* per unit time. We believe that if a traveler is traveling by bus, the BSCR should be greater than if they are not travelling by bus. We also evaluate the effectiveness of BSCR on predicting the transportation mode in the proposed work.

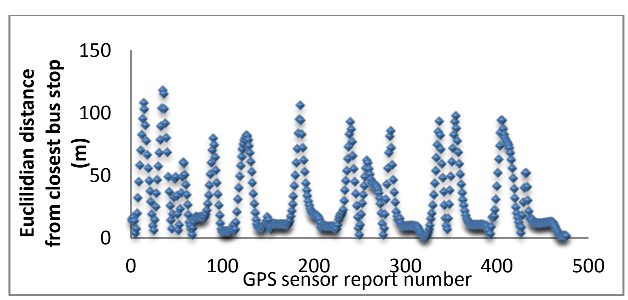


Figure 5 - Mobile user' Euclidean distance to closest bus stop while riding a bus

5.8 Zip code based Indexing and Pruning

Recall that from the user's GPS sensor reports we compute the closest Euclidian distances from real time locations of buses, rail line trajectories, and bus stops. Doing a linear comparison with all these locations can be time consuming.

We build a flat indexing scheme using zip codes; this scheme alleviates the overhead of doing the costly linear comparisons. First, we pre-compute the zip codes for all the bus stops, bus routes and train lines. Then, for each zip code in the city, we maintain a bus stop candidate list, bus route candidate list, and rail line candidate list. We cache this zip code index on the central server. Next, when a mobile user submits a GPS sensor report, instead of doing a linear comparison with all the bus stops, buses, and rail lines, we only compare against those in the same zip code from which the GPS sensor report was submitted.

In the proposed work, we compute the zip codes by reverse geocoding the spatial data. Reverse geocoding is the process of converting a location point to a readable address or place name. For example, reverse geocoding latitude: 41.976216 and longitude: -87.90331, produces the address 1-99 Access Road Chicago, Illinois, 60666, USA. From the address, we extract the zip code (i.e. 60666) component. This zip code extraction is done for each bus stop, rail line, and bus route to construct the zip code index. For reverse geocoding services, we use Yahoo's Reverse Geocoding API [34].

6. EVALUATION

In this section we discuss our training data collection procedure and the experimental results. We present the mode detection accuracy results when we ignore the transportation network, compared to detection accuracy results after we include transportation network information. Additionally, we use classification feature selection to rank our initial set of features. Given this ranking, we then select the highest rank features to build a final system,

6.1 Data Collection

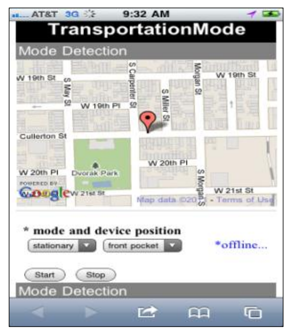
For training the classifier, we collected traces on six different modes of transportation (walking, bus, car, stationary, aboveground train and bike). The data was collected by 6 individuals, 3 females and 3 males. The data was collected over a 3 week period. Additionally, three types of mobile devices were considered for data collection: (1) HP IPAQ PDA, (2) Android based Samsung Galaxy mobile phone (3), IPhone 3G. These devices are shown in Figure 6. Our application platform is the mobile web. Transportation mode ground truth was labeled on the GPS sensor reports by each individual using the user interface (UI) of the mobile web application (see Figure 7a). In Figure 7b, we present a table, depicting the duration of mode specific labeled training data, collected from the six experiment participants.

Figure 6 - Devices used for data collection (HP IPAQ, Samsung Galaxy, IPhone 3G)

6.2 Training data preprocessing

The accuracy of GPS varies. For example, GPS tends to underperform if it does not have a clear view of the sky (e.g. in urban canyons). For this reason, we perform a noise filtering step before training the classifier. Invalid GPS points are suppressed based on the GPS accuracy and the change in speed. GPS sensor reports with high inaccuracy readings and unrealistic changes in speed are pruned. This is a manual step before classifier training. GPS noise filtering before classifier training is not a new concept. The authors of [4] perform a preprocessing step before training their classifier.

GPS sensor reports are submitted by the mobile user from the mobile device every 15 seconds. A window size of 30 seconds was chosen as the period of classification. Therefore, for every two GPS sensor reports received, we constructed the classification feature set. We observed that submitting GPS reports is very power consuming. Thus, submitting GPS reports every second as done in [4] will exhaust battery power.



Mode	Time (min)
train	62
bus	69
car	52
walk	91
bike	47
stationary	53

Figure 7 - (7a) User interface for data collection and ground truth labeling (7b) Amount of training data collected

6.3 Window Size

In the proposed work, transportation mode detection accuracy was not sensitive to window size. However, larger window sizes result in longer transportation mode detection time.

6.4 Classifier selection

To determine the most accurate classifier for the proposed transportation mode detection algorithm, we compared precision and recall accuracy of five distinct classification models. The five models are: (1) Naive Bayes (**NB**), (2) Bayesian Network (**BN**), (3) Decision Trees (**DT**), (4) Random Forest (**RF**), (5) Multilayer Perceptron (**ML**). Readers are referred to [32, 33] for discussions on various classification models. To evaluate the different classification models on transportation mode detection, the WEKA machine learning tool set [32] was used.

The results indicate that Random Forest (**RF**) is the best model, with an average precision accuracy of 93.70% and recall of 93.80%. Thus, Random Forest classification system is chosen as the final classification model that is deployed to the public.

6.5 Mode detection accuracy

In this section, we analyze the performance and effectiveness of the transportation mode classifiers. We evaluate the classification schemes using two metrics: (1) *precision accuracy* and (2) *recall accuracy.*

Precision Accuracy (M)=(*number of correctly classified instances of mode M) / (number of instances classified as mode M)*

Recall Accuracy (M) = (*number of correctly classified instances of mode M) / (number of instances of mode M)*

Three sets of results were obtained and are presented in subsections 6.5.1, 6.5.2, and 6.5.3, respectively. Each set contains the precision accuracy and recall accuracy for the five classification schemes. For each set, we used 10-fold cross validation. In 10-fold cross validation, the original sample was randomly divided into 10 subsamples. Of the 10 subsamples, a single subsample was retained as the validation data for testing the classification model, the remaining 9 subsamples were used for training data. The cross-validation process was then repeated 10 times, with each of the 10 subsamples used exactly once as the validation data. The 10 results were then averaged to produce a single estimation.

6.5.1 Classification without Transportation Network data

Figure 8 shows the first set of results which are the precision accuracy and recall accuracy when transportation network related features are not considered. The only features considered are average speed, average acceleration, average heading change, and average GPS position accuracy. Thus, real time bus locations, rail line trajectory and bus stop locations are removed.

From Figure 8 it can be observed that *Random Forest* classification model is the most accurate model since it has higher average precision and recall accuracy compared to the other four classification models.

In general, when transportation network related features are not considered, the accuracy is below 76% for the five classification models (BN, NB, DT, RF, and ML). Additionally, we observe that motorized transportation and bikes show the lowest precision accuracy. For example, consider the case of Random Forest. The precision accuracy results for car, bus, and train are 58.1%, 56.5%, and 69.8%, respectively. The precision accuracy for bike is 71.4%. On the other hand, for non-motorized modes, such as walk and stationary, the precision accuracy is 100% and 96.8%. For all five models (BN, NB, DT,RF, and ML), the precision accuracy is best for walk and stationary. This implies that the transportation network data is not very helpful for detecting stationary and walk mode. On the other hand, features such as speed, heading, acceleration, and GPS accuracy are not sufficient for distinguishing between motorized modes because of feature similarities.

6.5.2 Classification with Transportation Network data

Figure 9 shows the second set of results which are precision and recall accuracy when all the classification features discussed in Section 5 are used. The difference between Figures 8 and 9 is obvious. The main difference between Figure 8 and Figure 9 is that, in Figure 8, transportation network related classification features are not considered, while in Figure 9 transportation network related features are considered.

In Figure 9, all classifiers (**BN**, **NB**, **DT**, **RF**, **ML**) tested, with the exception of the Neural Network based Multilayer Perception (**ML**), achieve over 90% average precision and recall accuracy. On the other hand, when transportation network features are suppressed, the average precision and recall accuracy is below 76% (see Figure 8). This suggests that the transportation network related features are effective for transportation mode detection.

In the study, the most effective classification model is again Random Forest (**RF**), with an average precision accuracy of 93.7% and recall of 93.8%. This work is the first to distinguish between motorized transportation modes with such high accuracy [1, 2, 15, 16].

Figure 8 - Transportation network features not considered.

	Precision Accuracy					Recall Accuracy				
	NB	**BN**	**DT**	**RF**	**ML**	**NB**	**BN**	**DT**	**RF**	**ML**
train	70.0	50.0	50.6	69.8	47.8	45.2	85.5	62.9	59.7	51.6
bus	47.0	43.9	40.6	56.5	37.8	54.4	31.6	22.8	45.6	24.6
stationary	100	100	100	100	96.2	100	100	100	100	67.6
walk	94.7	93.8	93.8	92.7	83.3	97.8	100	98.9	97.8	98.9
car	42.3	90.0	43.5	58.1	30.2	42.3	17.3	38.5	69.2	25.0
bike	70.2	71.0	68.8	71.4	54.5	89.2	86.5	89.2	81.1	81.1
average	71.8	74.9	66.9	75.4	59.1	71.4	71.4	69.0	75.9	60.7

Figure 9– Transportation network features considered

	Precision accuracy					Recall accuracy				
	NB	**BN**	**DT**	**RF**	**ML**	**NB**	**BN**	**DT**	**RF**	**ML**
train	98.3	96.8	96.8	98.4	89.1	91.9	96.8	96.8	98.4	79.0
bus	85.0	88.3	88.9	88.3	83.3	89.5	93.0	84.2	93.0	87.7
stationary	100	100	100	100	96.6	100	100	100	100	75.7
walk	96.7	94.7	94.7	96.8	86.5	95.6	97.8	98.9	98.9	98.9
car	78.2	85.4	85.1	87.5	67.3	82.7	78.8	76.9	80.8	67.3
bike	88.9	88.6	85.5	88.9	75.0	86.5	83.8	94.6	86.4	73.0
average	91.6	92.5	92.2	93.7	83.3	91.4	92.6	92.3	93.8	83.0

From Figures 8 and 9, it can be seen that the precision and recall accuracy of motorized transportation modes and bikes increases more than the non-motorized modes of walk and stationary. For example, in the case of **RF** classification model, when transportation network features are used (Figure 9), the precision accuracies of car, train, bus, and bike are 87.5%, 98.4%, 88.3% and 88.9% respectively. On the other hand, when transportation network related features are not used (i.e. Figure 8), in the case of **RF**, the precision accuracies for car, train, bus, and bike are 58.1%, 69.8%, 56.5% and 71.4.% respectively.

We conclude that our novel transportation network features are most effective for motorized transportation mode detection, and also effective for bike mode detection. This makes sense, since bikes and motorized modes may have similar speed and acceleration in traffic and therefore are difficult to distinguish using traditional motion pattern features. However, features of the transportation network, such as bus locations, help the distinguishing between buses and bikes.

Distinguishing among motorized transportation modes is useful in practice. For example, companies such as Google collect data from travelers' mobile phones in order to estimate the traffic speed of a road segment. For this purpose the speed estimation system should only use the speed reports submitted by mobile devices on cars or buses but not those on trains. Distinguishing the train mode from the other motorized modes enables the speed estimation system to filter out speed reports submitted from trains.

The proposed approach is 17 % more accurate than [2, 15] which uses GPS only and distinguishes between two motorized modes (bus and car). Also, the proposed approach is 9% more accurate than [1, 16] which uses GPS/GIS and distinguishes between bus and car. Using the newly proposed classification features, we show that we can detect transportation mode with high accuracy. These classification features are robust and most effective for detecting motorized transportation and bikes.

6.5.3 Transportation Mode Classification Feature Selection

Feature selection is a data-mining concept [31], which chooses the subset of input features by eliminating classification features that are less predictive. Using two commonly used feature selection algorithms, we ranked the eight classification features to identify the most relevant features for detecting transportation mode in the proposed work. The feature selection algorithms used are: (1) Chi Squared [31] and (2) Information gain [31]. The ranking of the initial eight classification features are shown in Figure 10. Removing irrelevant classification features reduces the computational cost for training and transportation mode detection.

From Figure 10 we can see that the set of five top ranked classification features is the same for Chi Square [31] and Information Gain [31]. Thus, from Figure 10, we selected the five top ranked classification features, namely average speed, average acceleration, average rail line closeness, average bus closeness, and candidate bus closeness. We used these features to build a final classification model. The precision and recall accuracy of this final classification model is shown in Figure 11.

According to Figure 11, when the top five features are selected and the other three classification features are pruned, the precision accuracy hardly changes (see Figure 9 as well). Observe that Random Forest (**RF**) classification model is still the dominating classification model in Figure 11, with a precision and recall accuracy of 92.8% and 92.9% respectively. For **RF**, only a 0.9 % reduction in recall and precision accuracy is noticed when the five top ranked features are considered, as opposed to considering all eight classification features. This indicates that the top five classification features are enough to detect transportation mode in the proposed work.

In some cases, there is an increase in precision accuracy when only five features are considered. For example, consider the case of Random Forest precision accuracy for bus or bike transportation mode. For another example, the precision accuracy for **DT** when we consider the top five classification features is greater for cars and bikes, than if we consider all initial eight classification features.

In general, Figure 11 shows that even though we pruned three classification features, the accuracy is unaffected. This suggests that the three pruned features are redundant for detecting the transportation mode in the proposed work.

Figure 10 – Classification feature ranking and selection

Rank	Chi Squared	Information gain
1	average speed	average speed
2	average rail line closeness	average rail line closeness
3	average. bus closeness	average acceleration
4	average acceleration	average. bus closeness
5	candidate bus closeness	candidate bus closeness
6	average heading change	average heading change
7	average bus stop closeness	average bus stop closeness
8	average accuracy	average accuracy

Figure 11 - Only five high order classification features used.

	Precision accuracy					Recall accuracy				
	NB	**BN**	**DT**	**RF**	**ML**	**NB**	**BN**	**DT**	**RF**	**ML**
train	96.5	92.2	96.8	95.1	82.9	88.7	95.2	96.8	93.5	93.5
bus	81.3	85.5	83.9	89.7	71.6	91.2	93.0	91.2	91.2	93.0
stationary	100	97.3	100	100	84.2	91.9	97.3	100	100	86.5
walk	94.6	94.7	95.7	96.8	88.2	95.6	98.9	96.7	100	90.1
car	79.2	82.2	90.5	83.7	78.8	80.8	71.2	73.1	78.8	50.0
bike	91.7	93.9	85.4	89.2	85.7	89.2	83.8	94.6	89.2	64.9
average	90.6	91.0	92.4	92.8	82.2	90.2	91.1	92.3	92.9	81.8

The most effective features are average speed, average acceleration, average rail line closeness, average bus closeness, and candidate bus closeness.

Now we discuss Figure 10 from the perspective of transportation network data availability. Depending on its availability, the transportation network data can be categorized into three levels. The most widely available data is network topology data such as rail line routes. Figure 10 shows that this data is also most useful among transportation network features for mode detection. This is a good property of our approach. It means that our approach can be deployed to many regions in the world and is likely to achieve good performance there. The less widely available data is bus stop locations. Figure 10 shows that this data is least useful among the top ranked features. This means that our approach would not lose too much performance in the regions where bus stop information is unavailable. The least available data is real-time bus locations, which is a very predictive feature (i.e., average bus closeness) according to Figure 10. Thus our approach will not be able to utilize this predictive feature in many regions of the world. This is a limitation of our approach.

6.6 Performance and scalability

The speed at which we create the transportation mode classification features for training and inference is important. Recall, from the user's GPS sensor reports, we need to compute the closest Euclidian distances to real time locations of buses, rail line trajectories, and bus stops.

There are over 11,500 bus stops in the city of Chicago. For larger cities such as New York, the number of bus stops may be even greater. Doing a linear comparison with all the bus stops, buses, and rail lines is time consuming. Doing a linear comparison took us over 2 minutes on a HP Laptop with a 4GB RAM and 2.54GHz Intel Core 2 Duo processor. This is impractical and ineffective in the real world, since in two minutes, users may transfer from one transportation mode to another, or become frustrated with the system. The proposed zip-code based indexing and pruning approach reduces our feature creation time from over 2 minutes to below 10 seconds. This can be further improved by using more sophisticated techniques, such as indexing by R-trees. However, performance was not a focus of this work.

6.7 Extended real world evaluation

The final classification model (Random Forest), using the top five ranked features, was deployed to the public via the mobile web. As explained earlier, we focused on the centralized server model. In this model, mobile users submit their GPS sensor reports via the web to our central server for classification. The central server then responds to the user with the corresponding transportation mode. Below, in Figure 12, we show the final deployed transportation mode classification system under operation in an IPhone 3G.

When the detected transportation mode is "bus" in Figure 12, we provide further information by giving the bus's identification. The bus identification is a finer granularity of transportation mode detection than bus route; we can also detect the bus route on demand. This work is the first to infer such detailed transportation mode detection.

We also evaluated the deployed system to learn how the system performs in the real world under everyday usage. For evaluation purpose, new individuals that were not considered for the initial training data collection were given IPhone 3Gs with access to the classification system. We considered new individuals for this experiment, because we wanted to learn how the system would perform for new users that are not covered by the training data. These new users mounted the IPhone in any desired position (i.e. waist, arm, pocket, or bag), then tally the percentage of time the mode detection is correct. For example, if the mode was detected 8 out of 10 times correctly, the accuracy is 80%. The results of the real world evaluation for a mobile user are presented in Figure 13.

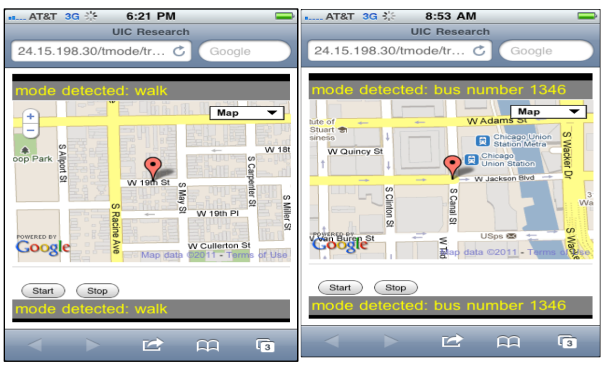


Figure 12– Deployed classification system

Figure 13 shows that, when deployed in the real world, under everyday usage, we achieved an average detection accuracy of 93.42% for the proposed mode detection system. The results indicate that the proposed approach is effective under everyday usage, and new training data collection is not necessary for new users. Also, the identified transportation network related features are very robust to traffic condition changes.

Figure 13 – Evaluation of deployed system

Mode	Duration (min)	Accuracy %
train	35	93
bus	30	95
car	30	89
walk	30	92
bike	30	93
stationary	34	98.5

6.8 Bus mode detection discussion

We presented three new classification features that can detect if a traveler is travelling via bus. The three features are: (1) average bus closeness (ABC), (2) candidate bus closeness (CBC), and (3) bus stop closeness rate (BSCR). From the feature selection in Figure 12, we observe that BSCR was overshadowed by ABC and CBC.

ABC captures the Euclidian distance to the closest bus for each snapshot. This Euclidian distance is summed over all the snapshots in a time window. Then, the average Euclidian distance is represented as ABC. ABC does not capture the traveler's relationship with all the buses, only the closest bus.

CBC requires the knowledge of the Euclidian distances to all the buses. Then, the single closest bus over a time window is chosen as the candidate bus. Thus, CBC does not capture bus transfers. For example, if a traveler alights from a bus, and boards another bus, CBC may not identify the correct bus.

According to Figure 10, ABC is a more effective classification feature than CBC for transportation mode detection. In order to quantify the contribution of CBC to bus mode detection accuracy as a classification feature, we present bus mode accuracy results using the Random Forest Model, when the CBC feature is suppressed. The four high order features (average speed, average acceleration, average rail closeness, and average bus closeness) of Figure 10 are used in Figure 14. From Figure 14, we observe that the precision accuracy of buses decreases to 85.1 % (Figure 14) from 89.7% (Figure 11), when only the top four features are used. This indicates that even though CBC may be more time consuming to compute, and use more memory than ABC, it is a worthwhile feature for buses. On the other hand, if speed and memory is critical; CBC can be suppressed, and bus mode detection will remain over 85% accurate.

Figure 14 – Effects of CBC on bus mode detection

	Precision Accuracy %	Recall Accuracy %
Bus	85.1	91.3

7. CONCLUSION

In this paper we proposed a new robust approach to detecting transportation modes. In the proposed work, we considered and used transportation network data consisting of real time location of buses, rail lines, and bus stops spatial data. The real time location of buses is available in many cities such as Chicago, New York, Toronto, London, Washington DC, and San Francisco.

Using the transportation network data, we showed that it is possible to address the weakness of previously proposed solutions [1, 2, 4, 15, 16]; that is, to distinguish between motorized modes such as trains, buses, and cars with high accuracy. Furthermore, if we detect that a traveler is traveling by bus, we can further identify on which particular bus the person is traveling.

Among the five classification models considered, Random Forest model is the most dominating classification model with over 93 % precision and recall accuracy. When transportation network classification features are not considered, the precision accuracy decreased to below 76%. This reduction of accuracy, upon omission of transportation network related features, is more notable for motorized transportation modes and bikes. This implies that transportation network data is effective for detecting motorized transportation, and bikes.

We also realized that, in order to achieve high precision and recall accuracy, only a subset of our initial set of classification features is necessary. In addition to traditional features on average speed and average acceleration, we identified for the first time the features on average bus closeness, average rail line closeness, and average candidate bus closeness. Using only this subset of features, and suppressing the other classification features that are not necessary, the precision accuracy was still over 92.5%.

Finally, users that have not participated in the initial training data collection evaluated the deployed system in the “real world” under everyday usage. The real world evaluation of the deployed system resulted in a precision accuracy of 93.42%. This indicates that additional training data collection is not necessary for new users; and the system is robust under every day usage.

8. ACKNOWLEDEMENT

This work is supported in part by the Google Mobile 2014 Program and NSF through grants CNS-1115234, DBI-0960443 and DGE-0549489.

9. REFERENCES

[1] D. Patterson, L. Liao, D. Fox, and H. Kautz, Inferring High-Level Behavior from Low-Level Sensors, ACM UBICOMP 2003.

[2] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W. Ma, Understanding mobility based on GPS data. In Ubiquitous Computing, ACM New York, 2008, pp. 312-321.

[3] L. Liao, D. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. Artif. Intell. 171, 2007.

[4] S. Reddy, M. Mun, J. Burke, D. Estrin, M Hansen, and M. Srivastava. Using Mobile Phones to Determine Transportation Modes. ACM Transactions on Sensor Networks, Vol. 6, No. 2, Article 13, 2010.

[5] E. Miluzzo, N. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. Eisenman, X. Zheng, and A. Campbell. Sensing meets mobile social networks: The design implementation and evaluation of the CenceMe application. 6th ACM Conference on Embedded Network Sensor Systems, 2008.

[6] J. Liu, O. Wolfson, H. Yin. Extracting Semantic Location from Outdoor Positioning Systems. Int. Workshop on Managing Context Information and Semantics in Mobile Environments (MCISME), 2006.

[7] H. Pfeifer A long term fix with GPS and GARtrip http://www.gartrip.de/long.htm

[8] S. Jiang, “Finding Ways out of Congestion for the Chicago Loop - - A Micro-simulation Approach”.

[9] K. Ahmed (1999) Modeling Drivers’ Acceleration and Lane Changing Behavior. MIT Ph.D. Dissertation.

[10] J. Ko, R. Guensler, M. Hunter. Analysis of effects of driver/vehicle characteristics on acceleration noise using GPS-equipped vehicles. Transportation Research Part F 13 (2010), 21–31.

[11] David Bernstein and Alain Kornhauser. An introduction to map matching for personal navigation assistants. Technical report, New Jersey TIDE Center, New Jersey Institute of Technology, Newark, New Jersey, 1996.

[12] Murakami, E., Wagner, D. P., Neumeister, D. M. (1997) “Using Global Positioning Systems and Personal Digital Assistants for Personal Travel Surveys in the United States,” International Conference on Transport Survey Quality and Innovation, Grainau, Germany. July 2004.

[13] E. Murakami, and D. P. Wagner, Can using global positioning system (GPS) improve trip reporting? Transportation Research Part C, 7(2/3):149-165, 1999.

[14] P. Gonzalez, J. Weinstein, S. Barbeau, M. Labrador, P. Winters, N. Labib, R. Perez, Automating mode detection using neural networks and assisted GPS data collected using GPS enabled mobile phones. 15th World Congress on Intelligent Transportation Systems, 2008.

[15] Y. Zheng, L. Liu, L. Wang, X. Xie. Learning Transportation Mode from Raw GPS Data for Geographic applications on the Web. WWW 2008

[16] L. Liao, D. Fox, H. Kautz. Learning and Inferring Transportation Routines. AAAI 2004

[17] Ermes, M., Parkka, J., Mantyjarvi, J., Korhonen I., Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions, IEEE Transactions on Information Technology in Biomedicine 12, 1(2006), 20-26.

[18] Parkka, J., Ermes, M., Korpipaa P., Mantyjarvi J., Peltola, J., Activity classification using realistic data from wearable sensors, IEEE Transactions on Information Technology in Biomedicine 10, 1 (2006), 119-128.

[19] T. Sohn, A. Varshaasky, A. LaMarca, M. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W. Griswold, E. Lara, Mobility Detection Using Every day GSM Traces. ACM UBICOMP 2006

[20] J. Hightower, S. Consolvo, A. LaMarca, I. Smith, J. Hughes. Learning and Recognizing the Places We Go. ACM Conference on Ubiquitous Computing, 2005.

[21] Walk Jog Run. http://www.walkjogrun.net/

[22] Mountain Bike. http://www.mtbroutes.co.uk/northyorkmoors/default.aspx

[23] SportsDo. http://sportsdo.net/Activity/ActivityBlog.aspx

[24] Wikiwalki. http://www.wikiwalki.com

[25] Chicago Transit Authority http://www.transitchicago.com/about/facts.aspx

[26] Chicago Transit Authority Bus Tracker http://www.ctabustracker.com/

[27] Google Maps http://googlemaps.com/

[28] http://www.transitchicago.com/assets/1/maps/ctatrainmap_2010jan.pdf

[29] L. Bao, S.S Intille. Activity Recognition from User Annotated Acceleration Data. Pervasive Computing, 2004.

[30] J. Lester et al. A Practical Approach to Recognizing Physical Activities. Pervasive Computing, 2006

[31] I. Guyon, A. Elisseef, An Introduction to Variable and Feature Selection. Journal of Machine Learning Research 3 (2003) 1157-1182, 2003

[32] I. Witten, E. Frank, Data Mining: Practical machine learning tools and techniques. Morgan and Kaufmann, San Francisco, 2005

[33] R. Duda, P. Hart, D. Stork, Pattern Classification. Wiley, New York, 2000

[34] http://developer.yahoo.com/geo/placefinder/

[35] L. Stenneth, P. S. Yu, O. Wolfson, Mobile Systems Location Privacy “MobiPriv” a Robust K-Anonymous System. IEEE WiMob, 2010

[36] L. Stenneth, P. S. Yu. Global Privacy and Transportation Mode Homogeneity Anonymization in Location Based Mobile Systems with Continuous Queries. 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010

[37] Y. Yang, T. Toida, C. Hong. Transportation prediction using build-in triaxial accelerometer in cell phone. International Conference on Business Information, 2010.