www.ietdl.org

Published in IET Intelligent Transport Systems
Received on 30th January 2009
Revised on 12th June 2009
doi: 10.1049/iet-its.2009.0029

Special Issue – selected papers from the 15th World Congress on ITS

ISSN 1751-956X

# Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks

*P.A. Gonzalez J.S. Weinstein S.J. Barbeau M.A. Labrador P.L. Winters N.L. Georggi R. Perez*

*Center for Urban Transportation Research and Department of Computer Science & Engineering, University of South Florida, Tampa, FL 33620, USA*
*E-mail: barbeau@cutr.usf.edu*

**Abstract:** Travel surveys collect trip data such as origin, destination, mode, duration, distance and purpose of trips, as well as socioeconomic and demographic data for analysis. Transportation planners, policymakers, state departments of transportation, metropolitan planning organisations, industry professionals and academic researchers use survey data to better understand the current demand and performance of the transportation infrastructure, and to plan in preparation for future growth. Next-generation travel surveys will utilise global positioning systems (GPS) to collect trip data with minimal input from survey participants. Owing to their ubiquity, GPS-enabled mobile phones are developing into a promising survey tool. TRAC-IT is a mobile phone application that collects real-time GPS data and requires minimal input from the user for data such as trip purpose, mode and vehicle occupancy. To ease survey burden on participants and enable real-time, mode-specific location-based services, new techniques must be explored to derive more information directly from GPS data. As part of travel survey collection, TRAC-IT is able to passively determine trip mode using GPS-enabled mobile phones and neural networks. The mode detection technique presented in this article can be optimised using a critical point, pre-processing algorithm to reduce the size of required GPS datasets obtained from GPS-enabled mobile phones, thus reducing data collection costs while conserving precious mobile phone resources such as battery life.

## 1 Introduction and motivation

Travel surveys are important tools used by transportation professionals to plan, design, evaluate and maintain the transportation system. Over the years, a variety of formats have been used for travel surveys, including mail-in paper forms, travel journal/diaries and computer-assisted telephone interviews (CATI). These methods rely on participants recalling trips completed within the past few days. Travel times, durations, locations and distances recorded by these surveys may contain inaccuracies because of misreporting by participants or manual data entry errors [1, 2]. An emphasis on activity-based modelling with day-to-day variability of travel behaviour necessitates multi-day surveys [3]. Increased burden on participants and concerns over non-responsiveness and/or survey cessation challenges transportation professionals. An important focus of these surveys is to identify origin and destination (OD) of trips, since other electronic devices such as loop detectors or manual counting devices on roadways do not collect this information. Mobile technology data provide greater accuracy for the locations of origins and destinations, when compared to self-reporting of multiple addresses that survey participants may not know well enough to convey [4]. Global positioning systems (GPS)-enhanced surveys may also provide more accurate information than conventional surveys on the start and end times of travel, detailed route travelled by the respondent, and total travel time and distance. Fully passive devices decrease the burden on participants making it possible to collect multiple days of travel data from a random sample of households within any study area.

Recently, in-vehicle GPS loggers were successfully used in several surveys to minimise the underreporting of trips [5]. Data collected by vehicle-based GPS are more reliable in reporting accurate travel times and locations. However, vehicle-based GPS surveys do not capture all modes of transportation used by the sampled population such as walking, biking and public transportation. Therefore vehicle-based GPS surveys record vehicle, not person trips. Person-based, wearable GPS loggers have been tested as travel diaries; however, this provided an additional burden on the participant to remember to carry the extra device, as well as the cost to the surveyor for purchasing the equipment [6]. Additionally, since the data loggers have a limited storage capacity and store data locally on the device, the devices must be manually retrieved from the survey participants to download the survey data. On-device storage means that if the logger is lost, the survey data are also lost. The GPS data collected from these stand-alone units can also differ in quality depending on the type of device. For some low-cost GPS-logger units that are carried in a user's pocket, the device may not have sufficient sensitivity to obtain a GPS fix and collect travel behaviour data by modes such as bus, biking or walking, where the user does not have a flat surface to place the data logger with a clear view of the sky, such as a car dashboard. The authors evaluated a Super Trackstick GPS data logger and were not able to collect any travel behaviour data when the device was carried in the pocket of the user. Since the person-based data loggers also use batteries, the batteries may have to be recharged or replaced during the survey period, resulting in additional burden to the participant and cost to the surveyor if the device does not feature rechargeable batteries.

In today's world, many people own and carry GPS-enabled mobile phones. Owing to the e911 mandate in the USA, wireless carriers must be able to locate a 911 mobile-phone caller to within 50–300 m of accuracy [7]. Various technologies have been developed to satisfy this mandate, including embedded GPS hardware in mobile phones. Subsidised as part of the cell phone manufacturing cost, GPS technology used in modern GPS-enabled cell phones exceeds the performance of consumer-grade, person-based GPS data loggers because of the required reliability of providing location information in an emergency. Cell phones utilise both assisted GPS, which utilises information from the cellular network to quickly achieve a first GPS fix (e.g. a 'cold-start'), as well as high-sensitivity GPS, an improved hardware design that can obtain a GPS fix even in highly obstructed environments such as indoors or between high-rise buildings [8]. For example, a Qualcomm MSM6125 chipset used in CDMA cell phones features a high-sensitivity −159 dBm GPS receiver, while the Super Trackstick data logger GPS sensitivity is specified at only −133 dBm [9, 10]. Additionally, the cold-start time for the Super Trackstick is rated at 52 s, while the cold-start for the cell phone is nearly instantaneous. While the three personal GPS data loggers studied by Auld and Williams do have high-sensitivity GPS, these devices also have lengthy cold-start times greater than 35 s [6, 11–13]. As a result, modern GPS-enabled cell phones are able to quickly acquire a GPS fix in environments where person-based GPS data loggers may not collect any data, including important cold-start situations such as when a user is leaving a building to start a trip and re-entering GPS signal coverage. The longer cold-start periods of GPS data loggers could miss part, or all, of a trip if the GPS receiver is unable to acquire a first fix before the user enters their car, and the level of GPS receiver obstruction again increases (e.g. if the logger is in the user's pocket, or bag inside the car). The regular use of the cell phone throughout the day may also place the phone in a better position to receive GPS signals (e.g. outside a pocket or bag) as compared to a personal data logger, which is likely to remain unmoved by the user once it stored.

The implementation of high-precision positioning technologies in cell phones has led to the creation of a class of software applications known as location-based services (LBS), which use the device's location in coordination with other data to create location-aware applications. Location-aware applications are the focus of several research projects at the Center for Urban Transportation Research and the Department of Computer Science and Engineering at the University of South Florida. One particular project, TRAC-IT, is a group of software applications used to monitor and analyse travel behaviour using GPS data gathered from GPS-enabled mobile phones [14]. The primary data collection software is a Java Micro Edition (Java ME) application that runs in real-time on the user's mobile phone, requiring no special hardware device for GPS data collection. Therefore TRAC-IT can be carried with the participant on all modes of transportation and can record user-specific travel behaviour, without requiring the user to carry any additional equipment. Even if a mobile phone is shared within the same household, the TRAC-IT software allows users to log in with a unique account so that collected data are attributed to the correct user. Therefore TRAC-IT collects trip data per user instead of per vehicle so that complex intra-household behaviour among multiple users can be analysed. The travel behaviour data collected using TRAC-IT can be utilised for many different purposes, including travel demand models for policymakers and transportation professionals as well as real-time traffic information services for the user personalised by their real-time location. To the authors' knowledge, TRAC-IT is the only Java ME software worldwide that has been designed specifically as both an electronic personal travel survey tool,



doi: 10.1049/iet-its.2009.0029

as well as a real-time travel information tool for cell phone users based on their current location. Therefore personalised travel information services, such as incident or congestion alerts for events that may lie ahead on the user's predicted path, provide the TRAC-IT user a direct incentive for continuing to participate in long-term travel surveys. Additionally, there is no need for the phones to be retrieved from the participant to download data since collected data are transferred to a server in real-time. Real-time data collection also prevents data loss if a participant loses their cell phone.

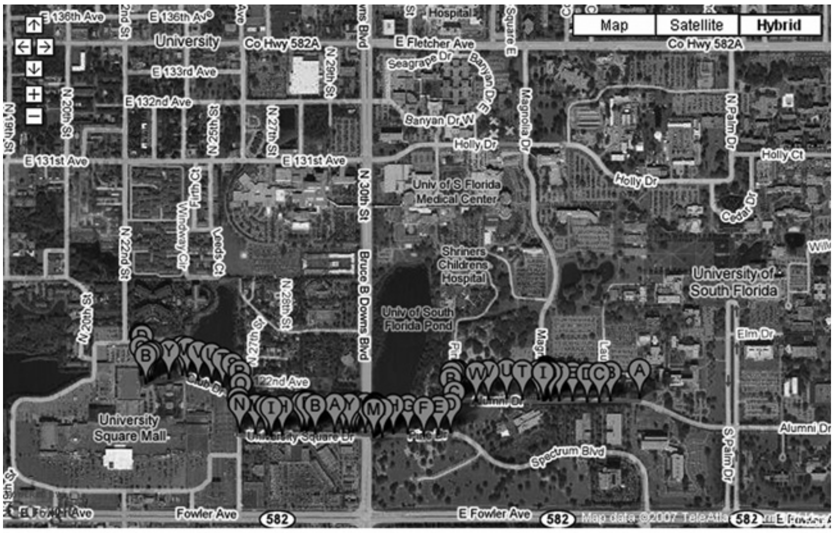
While GPS data are collected passively from the mobile phone, the TRAC-IT application user interface can also actively collect other trip information not directly recorded by GPS alone, such as trip purpose, mode of transportation and vehicle occupancy. As with other travel surveys that rely on participant's input, manual data input raises accuracy concerns. The burden of repeated manual data entry may cause participant fatigue and result in individuals dropping out of the survey. For these reasons, it is desirable to automatically derive trip attributes directly from the GPS data and eliminate the need for active user input. Additionally, passively detecting the mode of transportation enables new types of precisely targeted, real-time, location-aware applications that deliver information relevant to the user's current mode, such as transit-specific alerts related to delays in bus schedules, or traffic-incident alerts and alternate routes for drivers.

Many difficulties are encountered when attempting to automate mode detection. Buses and cars tend to display similar attributes as seen in Fig. 1. When comparing walking to other modes of transportation, the difference is noticeable (Fig. 1). Noteworthy here is that the walking trip is shorter than the other two trips, and less distance is travelled between the two sequential GPS fixes. Software applications can be programmed to differentiate a walking trip from a car or bus trip. However, this is not the case when distinguishing a car trip from a bus trip. The distance travelled in the bus trip is approximately the same as that travelled by car, and the distance between the two GPS fixes is almost identical. Owing to these and other similarities between car and bus trips, determining whether the trip mode is a car or a bus becomes rather challenging. Geographic information systems (GIS) data showing bus stops or routes can be utilised to determine whether the trip was taken by car or bus, but GIS data from transit agencies are often not available in a consistent format, may have issues with bus stop accuracy or may simply be out of date. Since the utility of mode identification techniques rely on specifically formatted GIS data from transit agencies and since specialised GIS software is limited, expensive and requires specific expertise to use and maintain, it is therefore advisable to find other means for mode detection techniques, independent of traditional GIS.

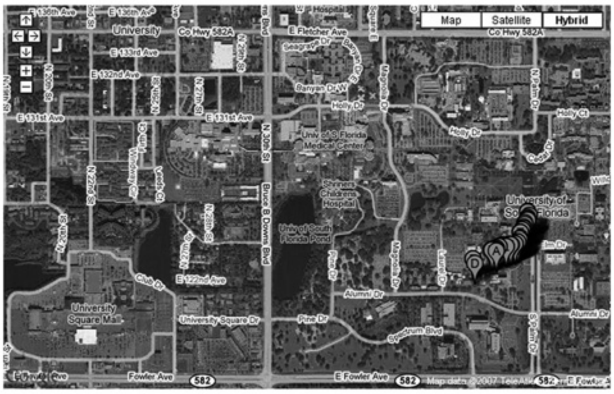
This paper focuses on the feasibility of using neural networks to automatically detect the mode of transportation from assisted GPS data collected using actual GPS-enabled mobile phones, through the standardised application-programming interface (API) for Java-enabled mobile devices, the JSR179 Location API [15]. This paper also explores how this method can be modified to detect the mode of transportation by examining only critical points (i.e. a minimum set of GPS fixes required to accurately reconstruct the user's path). Successful mode detection using fewer GPS data points enables the TRAC-IT mobile application to transmit fewer GPS fixes to a server, thereby saving costs incurred for data transfer, as well as conserving device and network resources including battery energy, network bandwidth and required storage space for collected data.

## 2 Related work

Ohmori *et al.* [16] implemented and tested a custom Java ME activity diary application on GPS-enabled mobile phones in 2006 in Japan for the sole purpose of travel behaviour data collection (i.e. the user does receive any incentive, such as traffic incident notifications, for their participation). Ohmori *et al.* concluded that users recorded activities more frequently in the mobile phone than in the paper diary, with less time delay or 'lag' in between when the activity was performed and recorded. Additionally, data handling and analysis time for the surveyor was considerably reduced for the mobile phone survey compared to the paper survey. However, several challenges remained before broad deployment of the software could be considered. First, the battery life of the phone was reduced to 5–6 h while the tracking application was running because of the extra energy consumption of the GPS receiver (set to record a position every 10 min), and wireless transmission of travel data back to the server. A GPS fix every 10 min is not frequent enough to yield high-resolution trip data such as the travel path or precise OD, and it is assumed that more frequent GPS calculations were not performed by Ohmori *et al.* since battery life would be decreased even further. Second, the financial cost of transferring location data over the cellular network to a server was approximately 900 yen (~\$9.18) for 1 day of data collection per respondent. This financial cost is likely another reason why GPS was not updated at a rate more frequent than once every 10 min. Finally, the limitations of the software used by Ohmori *et al.* prevented the user from making phone calls or sending emails or text messages while the tracking application was running. Users actually reported altering their behaviour as compared to their typical behaviour while participating in the survey; they reduced their travel because of the limited battery life of the phone, and reduced their communication with others since they could not place phone calls, text message or email while the tracking application was active. From a surveyor's point of view, this is not desirable since the observed travel behaviour may not be the same as the unobserved travel behaviour that would have otherwise occurred.




*a*



*b*


*c*

**Figure 1** *GPS trip data for different modes of transportation*

*a* Bus
*b* Walk
*c* Car

TRAC-IT was designed to improve upon critical limitations, such as those observed by Ohmori *et al.* in previous location-aware applications. First, TRAC-IT implements several 'smart' algorithms that execute on the phone to conserve battery energy [17]. A position recalculation management algorithm, or 'location-aware state machine', is used to



doi: 10.1049/iet-its.2009.0029

put the GPS in a sleeping state when frequent position recalculation is not needed (e.g. the participant is indoors or stopped for an extended period of time), and to transition to an active state when the user is moving. The state machine allows high-resolution travel behaviour data collection up to one GPS fix per second during travel, while automatically detecting when the user is stopped or indoors, and subsequently increases the amount of time between GPS fixes to save energy. A ‘critical point’ algorithm that pre-filters GPS data on-board the mobile device is also used to reduce the amount of unnecessary information transferred from the phone to the server, thus reducing impact on the battery as well as the financial cost of data transfer. Fig. 2 compares all GPS data points collected during a car trip (top), to the remaining GPS data points of the same trip after applying the critical point algorithm (bottom). The critical point algorithm uses attributes of sequential GPS fixes such as speed and heading to eliminate fixes that do not contribute path information. This technique is a dynamic variation on the concept of ‘line simplification’ in the area of GIS. For example, if a user is standing still (determined by the speed value of the GPS fix), only the first GPS fix recorded at that location would be considered a critical point, and the remaining points that duplicate that location information would be discarded until the user begins moving again. Similarly, if the user is travelling in a straight line, only the end points of the line (determined by the user’s change in direction) are considered critical, and the remaining points that fall on the line between the end points would be discarded.

The location-aware state machine is able to save significant energy. From benchmarking by the authors on a Sanyo Pro 200 mobile phone on the Sprint CDMA network, battery life is extended from 8 h (at a 4 s interval between GPS fixes), to over 14 h at a 30 s interval, and upwards of 41 h at the 5 min interval. On average, the critical point algorithm is able to filter out more than 80% of the total number of GPS fixes recorded during a trip, while retaining the critical points necessary to recreate the travel path. If the data transfer costs for a cellular network are charged by the byte, the cost to transfer a user’s travel behaviour is therefore reduced to only 20% of the original cost before using the critical point algorithm. (In the USA, unlimited data plans are now available on some cellular networks for a cost of ~\$10–15 per month per user, which should help reduce data collection and transfer costs for the surveyor. If the participant already subscribes to an unlimited data plan, then there would be no financial cost to the surveyor to transfer data from TRAC-IT to a server for that phone.) Benchmarking tests performed by the authors to evaluate energy savings of the critical point algorithm demonstrated that decreasing the frequency of wireless transmissions of location data from once every 15 s to once every 30 s, battery life for a Sanyo 7050 mobile phone on the Sprint CDMA network increases from approximately 9–17 h. If frequency is decreased further to one transmission every 60 s, battery life reaches approximately 30 h. Further detail about the methodology of the location-aware state machine and critical point algorithm, as well as the energy and cost benefits, can be found in Barbeau *et al.* [17].

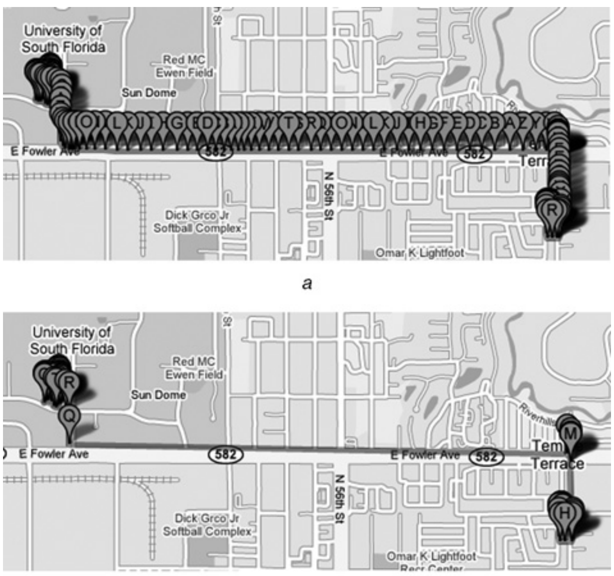



**Figure 2** *Illustration of a car trip where all GPS points against only critical points are detected*

In another key improvement over the system studied by Ohmori *et al.*, TRAC-IT has been designed to run as a background application on the participant’s phone. Therefore the participant can still make and receive phone calls, text messages and emails without any interference from the TRAC-IT application. When combined with the energy saving algorithms, TRAC-IT is capable of running in the background on a mobile phone, and providing personalised traffic incident alerts without significantly impacting the survey participant.

Byon and Abdulhai [18] have examined automatic mode detection using GPS and neural networks. However, that study used surrogate GPS data collected using stand-alone GPS units tethered to a laptop, instead of GPS-enabled cell phones. While Byon’s research is informative for general strategies in mode detection using neural networks, it does not accurately reflect the feasibility of mode detection using GPS data from mobile phones. Since characteristics of assisted GPS technology used in GPS-enabled mobile phones include increased sensitivity and a reduced time-to-first-fix, mobile phones can yield location data that are significantly different from data generated by traditional stand-alone GPS devices. Neural network performance is dependent on the characteristics of input data; therefore mode detection using assisted GPS data from GPS-enabled mobile phones must be analysed. Byon’s research also used attributes of GPS data such as instantaneous acceleration and horizontal dilution of precision (HDOP), properties which are not accessible


doi: 10.1049/iet-its.2009.0029

through the standardised API implemented in most GPS-enabled mobile phones, the JSR179 Location API [15]. Therefore new data attributes that are available in assisted GPS data obtained through the JSR179 Location API must be substituted as potential neural network inputs.

Byon and Abdulhai also examine how the frequency of location updates from the mobile phone impact the accuracy of mode detection. It is best to maximise the interval of time between location updates to minimise the impact on battery energy and to reduce the financial costs incurred for every transferred GPS fix. In their research, Byon and Abdulhai assume that the frequency is a static, fixed interval that cannot be dynamically modified, and that the cellular network controls the frequency of position updates. However, since the TRAC-IT mobile application is running on the mobile phone and utilising the JSR179 Location API, it is capable of intelligently pre-processing (i.e. filtering) location data using the critical point algorithm to send only GPS fixes that are required to reconstruct an accurate representation of the user's path to a server. The critical point method differs from a fixed update interval in that the fixed update interval will always report a position every $X$ seconds, even if that position information does not contribute to the knowledge of the user's path. The critical point algorithm provides a dynamic update method without a fixed location update interval, and is capable of providing GPS data points collected at strategic locations of the path instead of random locations governed by a static update interval.

This paper focuses on the analysis of assisted GPS data collected using the TRAC-IT GPS-enabled mobile phone application. It specifically investigates automatic travel mode detection with a neural network by utilising collected assisted GPS data. Since battery life of the cell phone is of utmost importance, and data transfer costs between phone and server can be significant, it is highly beneficial to develop data analysis techniques that accurately function by using only critical points, instead of requiring that the full GPS dataset be transferred from the phone to a server. This research therefore also evaluates the feasibility of mode detection based on a minimum set of GPS fixes that have been filtered using the critical point algorithm to discard non-critical data points on-board the cell phone, and send only critical points to the server. By sending only a subset of GPS data points to the server, significant device resources (e.g. battery life, data transfer costs etc.) can be saved during data collection, thereby significantly reducing the impact of passive data collection on the user and their cell phone.

## 3 Methodology

A neural network called a multi-layer perceptron was used for the research presented in this paper. Neural networks can extract subtle information from data that is often missed by humans or other analysis algorithms [19]. A neural network takes a series of inputs, processes them through hidden nodes and then generates an output that classifies the input data (Fig. 3). During the training of the neural network, the calculated output is compared to the known correct output for the training dataset being used. If the neural network's output for the test data is incorrect, the weights between the input nodes, hidden layer nodes and output nodes are adjusted accordingly in a method known as 'back propagation'. The training process is repeated until the accuracy of the network reaches a threshold defined by the user, or until a certain number of training iterations known as 'epochs', is completed. The weights, which represent the trained neural network, are then used by the network to evaluate future data for which the correct output is not known. Various settings for the neural network model can be adjusted to affect the training process including the number of hidden layers, the number of nodes in each hidden layer, the rate at which the weights are adjusted (i.e. the learning rate) and the amount of training (i.e. epochs).

The ability of a neural network to generalise conclusions for new data, which do not exactly match the training data, is one of the significant strengths of this type of artificial intelligence. Based on the chosen neural network settings and attributes of input data, the neural network can learn the subtle differences between car, bus and walking trips, and therefore automatically determine the mode of transportation for a new, previously unseen trip.

An open-source Java application that supports a variety of knowledge discovery functions, Weka, was used to determine the feasibility of automating mode detection with neural networks and assisted GPS data [20]. A multi-layer perceptron is programmatically created with particular settings that are further discussed in Section 4 of this paper. After the neural network is created, the inputs for each GPS trip are calculated and fed into the neural network. Next, a 10-fold cross-validation is performed to

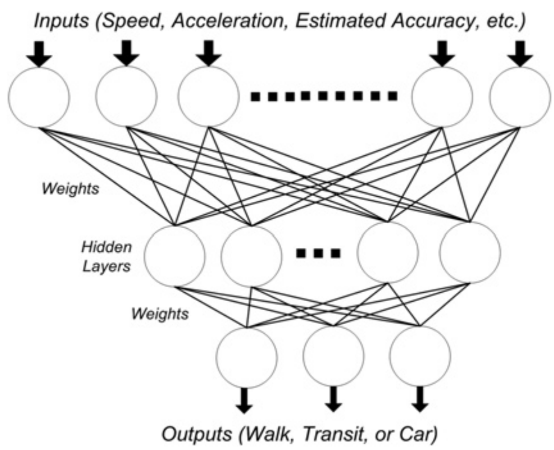



**Figure 3** *Neural network topology for mode detection*



doi: 10.1049/iet-its.2009.0029

train and test the neural network. The $N$-fold cross-validation randomly partitions a set of data with known correct outputs into $N$ folds, and then uses one fold as a testing dataset, while the remaining $N-1$ folds are used as the training dataset [21]. This process is repeated $N$ times with a different fold being used as the testing data for each iteration. The resulting $N$ accuracy measurements taken from the testing datasets are averaged to provide a realistic performance measure of expected accuracy on future unseen data. Once the 10-fold cross-validation is finished, the neural network outputs a trained model that can be used to evaluate new unseen data as well as the average accuracy of automatic mode detection for the evaluated datasets.

To collect the data evaluated in this paper, the TRAC-IT Java ME software was installed on Motorola i870 and i580 cell phones on the Sprint-Nextel iDEN network and set to record the GPS position of the cell phone user every 4 s. While members of the research team were travelling via different modes of transportation (i.e. walk, bus and car), each phone immediately sent each calculated position to the server so that the full set of all location data calculated during the trips was recorded. The research team made manual note of the mode of transportation (e.g. walk, bus or car) they were using via the 'active' portion of the travel survey in the TRAC-IT Java ME mobile application, and this information was used to mark the correct known mode of transportation for a given set of GPS data in the server database. By manually labelling the data with the known correct mode of transportation, the neural network can be trained by comparing the given output for the detected mode (e.g. bus) and the known correct classification (e.g. car), and adjusting the weights in the neural network by using back propagation if the two mode of transportation values do not match.

After all location data were collected using the GPS-enabled cell phones, the critical point algorithm was then executed on the server to post-process and label the subset of critical points, which appeared in the full GPS dataset. The result of this process is two datasets, one containing the entire representation of all location data recorded during a trip where the critical point algorithm was not used to pre-process GPS data on-board the cell phone. The other dataset contained the representation of the exact same trip using only critical points as if the critical point algorithm was used to pre-process GPS data on-board the cell phone. Since GPS data characteristics can differ between two trips, even if the same travel path is taken, this methodology is desirable over recording and comparing two different sets of GPS data (e.g. one trip with all GPS points and a second new trip with only critical points). Using the exact same GPS data to produce both full and critical points-only datasets ensured that the resulting differences in classification accuracy of the neural network were not attributed to variations in GPS characteristics for two different trips.

Since the neural network is completely dependent on the training dataset to learn to distinguish modes of transportation, it is important to examine the data attributes and statistics from the GPS trips that are used as input to the neural network. Furthermore, if mode detection can be successful when using statistics gathered from only critical points from a recorded trip instead of the full dataset of points, then significant mobile device resources (e.g. battery life, data transfer costs, storage space) can be saved by transferring only critical points from the cell phone to the server. Therefore attributes of both the full GPS trip datasets as well as the reduced critical-point datasets collected for experimentation were analysed for possible use as input to the neural network. This process of attribute selection for each dataset is described in the following subsections.

### 3.1 Neural network input for mode detection using all GPS points

It was desirable to select GPS data attributes with clear differences between modes to allow the neural network to easily identify a particular mode. Data attributes/measures used as inputs to the neural network for a particular trip when using the dataset containing all GPS points recorded during that trip included average and maximum speeds, estimated horizontal accuracy uncertainty, percent Cell-ID fixes, standard deviation of distances between stop locations and average dwell time. A rational for the inclusion of each measure is presented below:

- The average speed and the maximum speed were chosen as inputs since different modes of transportations will exert different average and maximum speeds. For example, the maximum speed of a car can be 80 miles/h, whereas a bus will not likely reach that speed.

- The estimated horizontal accuracy uncertainty is a measurement of the estimated confidence that the mobile phone places in a calculated GPS position. This value was chosen as an input since different modes of transportation have different values for estimated horizontal accuracy uncertainty. For example, a bus has the worst estimated horizontal accuracy uncertainty since it is essentially a metal box that obstructs the GPS signals. Conversely, a walking trip has the best-estimated horizontal accuracy uncertainty since typical GPS signal obstruction is caused by a bag or purse.

- Another important input for this algorithm is the percent Cell-ID fixes, which are the percentage of location fixes that refer to cellular signal coverage area instead of the GPS-calculated position of the phone. Cell-ID location fixes are obtained by the mobile phone when it cannot calculate a GPS fix. This percentage also serves as a rough measure of GPS signal quality.

- The standard deviation of distances between stop locations is also used as an input since different modes of transportation exhibit different types of values. For example, a bus trip is likely to travel less distance between stops than a car because of the repeated stops that a transit vehicle typically makes.

- Finally, the average dwell time is used as the last input. The average dwell time refers to the average length of time that a user is stopped each time their speed falls below a certain threshold. For example, the average dwell time is often higher for car or walking trips and lower for bus trips since most buses make quick stops to allow riders to board or exit the vehicle.

## *3.2 Neural network input for mode detection using critical points only*

As mentioned in the previous section, enabling automated mode detection while using only critical points from a dataset is important since efficient, location-aware mobile applications running on cell phones will only transfer critical points to a server for analysis. Therefore most of the input attributes mentioned in the previous section when using the full GPS dataset cannot be used, as there is not sufficient data to calculate the values when only critical points are available on the server. For example, to calculate the accurate standard deviation of stops and the average dwell time, all of the GPS fixes recorded during the trip are necessary. Additionally, no Cell-ID fixes are transferred to the server when using the critical point algorithm because the location of the cell tower does not contribute to the precise information of the user's travel path, and therefore the 'Percent Cell-ID fixes' attribute cannot be used. For these reasons, different input attributes to the neural network are required when only critical points for a trip are available in the GPS dataset on the server. The following input attributes are used for critical points:

- average acceleration;
- maximum acceleration;
- average speed;
- maximum speed;
- ratio of the number of critical points over the total distance of the trip;
- ratio of the number of critical points over the total time of the trip;
- total distance;
- average distance between critical points.

Two important input attributes are the average and maximum acceleration of a trip. Byon and Abdulhai note that acceleration is a key identifier of different types of modes, and therefore is a valuable attribute for automated mode detection [18]. Since instantaneous acceleration is not directly exposed to the application by the JSR179 Location API, the average acceleration must be calculated by dividing the change of velocity by the change in time of two consecutive GPS points. However, calculating average acceleration poses a problem when using the traditional critical point algorithm since by definition, critical points are not temporally or spatially near one another. Therefore a modified version of the critical point algorithm was created that records the velocity and time value for the GPS point either immediately after, or immediately before the current critical point. Having two velocity and time values per critical point allows the change of velocity and the change in time to be calculated to obtain the average acceleration for that time. The maximum average acceleration can be selected from the set of average acceleration values. The set of average acceleration measurements can be added and divided by the number of measurements to calculate the average acceleration for the entire trip.

It was concluded that recording the GPS fix immediately after the critical point was preferred over recording the fix immediately before the critical point, since positive acceleration values are preferred over deceleration. A traveller is most likely to decelerate just prior to the critical point fix, given that they may be changing directions. After the traveller makes the turn, the acceleration is more likely to be a higher positive value. This modified version of the critical point algorithm can be implemented so that the number of wireless transmissions to the server is not increased over the traditional algorithm. The transmission of a critical point can be delayed until the next GPS fix is calculated, and the velocity and time values for this second GPS fix can be packaged along with the critical point information that is then sent to the server.

As discussed when examining the full GPS dataset neural network input attributes, the average and maximum speed values are important since different modes of transportation exert different speeds. For these attributes, the cell phone can keep track of the statistics as it pre-processes all GPS fixes and then transfers the calculated values of average and maximum speed to the server at the end of the trip. This methodology allows the cell phone to transfer only critical points to the server during the trip while still providing simple statistics to the server based on all GPS data, without having to actually transfer the entire GPS dataset to the server. The ratio of the number of critical points over the total trip distance is another important input since different travel modes may have different ratio values. These ratios can be post-processed easily on the server based on the critical point dataset.

Table 1 shows the average number of critical points recorded using different modes of transportation for a sample of seven car, walking and bus trips. All data were collected using the TRAC-IT Java ME application on Motorola i870 and i580 phones on the Sprint-Nextel iDEN network. As the table describes, car trips are likely to have the highest critical point count since cars are more



doi: 10.1049/iet-its.2009.0029

**Table 1** Number of critical points for a sample set of trips

| # CP per trip | Car | Walk | Bus |
|---|---|---|---|
| trip 1 | 146 | 137 | 51 |
| trip 2 | 86 | 35 | 120 |
| trip 3 | 70 | 124 | 50 |
| trip 4 | 238 | 59 | 45 |
| trip 5 | 103 | 15 | 46 |
| trip 6 | 79 | 140 | 43 |
| trip 7 | 44 | 9 | 89 |
| average | 109 | 74 | 63 |

likely to make turns in a given trip. The second highest is the walking trips, followed by the bus trips, which show a fewer number of critical points. Taking the number of critical points of a trip and dividing it by the total trip distance shows a trend in ratios within the same travel mode since the number of turns in relationship to the total distance tends to be unique for each mode.

Fig. 4 shows the different ratio values for each transportation mode. A similar ratio, the number of critical points over the total time of the trip, is also used to train the neural network. Similarly, different ratio trends within the different transportation modes are seen in Fig. 5. These two ratios help the neural network to further learn the differences between car, bus and walking trips.

Since car and bus trips are the most difficult to distinguish from one another, it was best to find GPS data attributes for these two types of trips that are very different. One such input is the average distance (miles) between critical points (Fig. 6). There are significant differences in the average of the distance between different modes of transportation, and consistent differences between cars and buses. Buses tend to change their travel direction more often than cars, and therefore,

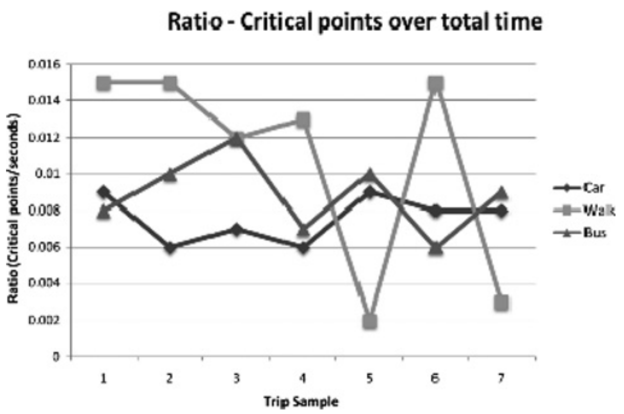



**Figure 4** *Number of critical points per second for sample trip set*

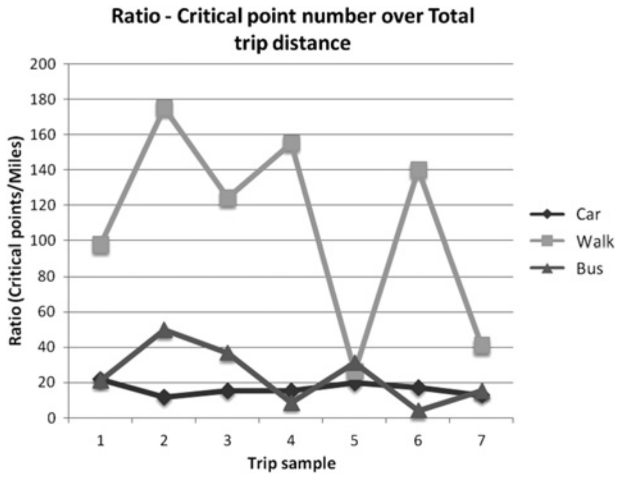



**Figure 5** *Number of critical points per mile for sample trip set*

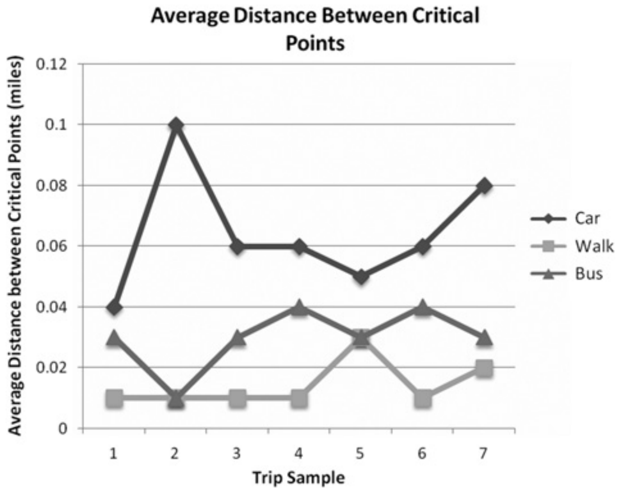



**Figure 6** *Average distance (miles) between critical points*

less distance is covered between critical points, while walking trips change travel direction more frequently.

Finally, the total distance (miles) is another important input for the neural network. As previously mentioned, different modes have different trip distances. For example, a walking trip distance will rarely exceed 3 miles, whereas the majority of car or bus trips are often more than 3 miles.

## 4 Performance evaluation

To train and test the multi-layer perceptron using 10-fold cross-validation, 114 trips were recorded in Tampa, Florida, using the TRAC-IT Java ME application on Motorola i870 and i580 phones on the Sprint-Nextel iDEN network. Assisted GPS data were gathered for 38 car, 38 bus and 38 walking trips. Usually, only critical points would be transferred from the mobile phone to a server, but for the tests, all GPS fixes were transferred to the server and the critical points algorithm was used to


doi: 10.1049/iet-its.2009.0029

mark the critical points in the database during post-processing, as described earlier. This process allows either all GPS points, or only critical points to be input into the neural network for the exact same trip.

GPS data for the 114 trips are first extracted from the database server and written into Weka's ARFF file format, which is then read by Weka and input into its neural network function for training and testing purposes. For all tests, Weka's auto-build feature was used to create the neural network topology with a single hidden layer that consists of a particular number of nodes. The number of hidden nodes is determined by the following equation

$$\text{Number of hidden nodes} = \frac{\text{Number of attributes} + \text{Number of possible classifications}}{2}$$

Since there are additional settings in the neural network that can be manipulated, it is important to evaluate the effect of different settings on the accuracy of mode detection. The following sections illustrate the performance gains achieved from utilising different neural network settings.

### 4.1 Evaluation with initial settings

The performance of the neural network is evaluated on the two sets of trip data (the all GPS points and only critical points). Default settings of the network include the learning rate (a number from 0 to 1) that represents the level of weight adjustment during each round of back propagation, and the training time that is the number of training epochs that the network should perform. The initial learning rate is set to 0.01 and the training time is set to 100 epochs.

The accuracies for mode detection using the initial neural network settings are 57.02% for all GPS points and 62.29% for only critical points.

### 4.2 Exploring different settings

Changing the multi-layer perceptron's settings helped improve the accuracy for mode detection as seen in Table 2.

As indicated by the table, setting the learning rate and the training time to different values affects the accuracy for mode detection while using the same set of inputs. A learning rate increase to 300 epochs has the highest accuracy of 91.23% for critical points, indicating that the neural network benefits from additional training over 100 epochs; however, over-training the network results in lesser performance on test data since the 'overfit' model cannot generalise well on unseen data. Additionally, using only critical points further aids the network in successful classification by eliminating GPS data that do not contribute to useful information about the mode of transportation.

**Table 2** Accuracy of mode detection with different multi-layer perceptron's settings

| Multi-layer perceptron's settings | | Accuracy | |
|---|---|---|---|
| The learning rate | The training time, epochs | All GPS points, % | Only critical points, % |
| 0.1 | 300 | 88.60 | 91.23 |
| 0.3 | 300 | 85.97 | 88.60 |
| 0.3 | 500 | 85.09 | 85.97 |
| 0.01 | 100 | 57.02 | 62.29 |

### 4.3 Correlation between input attributes and mode detection accuracy

This section explores how utilising different subsets of data attributes from the critical points-only dataset can help the neural network achieve a higher classification accuracy percentage. The purpose is to see if the number of input attributes for this algorithm can be reduced to eliminate noise that negatively affects the neural network's classification.

Table 3 shows how utilising different subsets of all possible attributes affects the classification accuracy of the neural network. The neural network settings found to be optimal were a learning rate of 0.1 and a training time of 300. Some subsets that include more input attributes actually degrade accuracy instead of improving it when compared to subsets with fewer input attributes. For example, the set of selected inputs in Subset F has an accuracy of 90%, but when the 'ratio of critical point number over total time' attribute is included as an input (Subset G), the classification accuracy of the neural network decreases. A similar situation occurs between the inputs used in Subsets D and E. This reduction in accuracy when new inputs are added can be a result of input noise that confuses the neural network instead of helping it learn. It is important to note that the highest accuracy (91.23%) was reached when utilising all selected inputs. Each attribute adds a specific piece of information related to characteristics of each mode, so that when all attributes are considered together, the mode is easier to identify. The second highest accuracy, 90.4%, was achieved when the ratio of critical point number over total time is removed from the input dataset.

The critical point algorithm effectively reduces the amount of noise input into the neural network, which results in a higher classification accuracy for transportation modes when the critical points-only dataset is used to train the neural network, instead of the full GPS dataset. By eliminating GPS fixes that do not directly contribute to knowledge of the travel path of the cell phone user, the critical point algorithm pre-filters the GPS dataset before




doi: 10.1049/iet-its.2009.0029

**Table 3** Accuracy of mode detection when using different subsets of data attributes as neural network inputs

| Different subsets of attributes | All possible attributes | | | | | | | | Average accuracy for given attribute subset, % |
|---|---|---|---|---|---|---|---|---|---|
| | Maximum speed | Average speed | Maximum acceleration | Average acceleration | # CP over total distance | # CP over total time | Total distance | Average distance between CP | |
| subset A | X | X | X | X | | | | | 86.0 |
| subset B | X | | X | | X | | | X | 86.0 |
| subset C | X | | X | | | | | X | 85.1 |
| subset D | X | | X | | X | | X | | 87.7 |
| subset E | X | X | X | X | X | | X | | 86.9 |
| subset F | X | | X | | X | | X | X | 90.4 |
| subset G | X | | X | | X | X | X | X | 89.4 |

the data are input into the neural network. Fortunately, this discovery means that mobile devices can utilise the critical point algorithm to pre-filter location data before these are sent to the server in order to reduce the amount of 'non-critical' information sent to the server, thereby significantly contributing to longer mobile device battery life and reduced costs of transmitting data over a cellular network.

### *4.4 Limitations of current state of research and future work*

To understand where future improvements can be made to automated mode detection using neural networks, it is important to understand the accuracy of the classification of each mode individually, to determine the most significant source of error. Table 4 shows the accuracy achieved for each individual mode of transportation, for the highest overall mode detection accuracy of 91.23%. These results demonstrate that bus trips are suffering from the highest number of incorrect classifications, at an 18.42% error rate. The car mode follows with 7.89% of trips classified incorrectly. It is clear that the neural network easily differentiates walking trips since none was incorrectly classified. For future research, efforts will focus on finding additional input attributes to better differentiate bus against car trips since this is the greatest potential area to improve

**Table 4** Accuracy of mode detection for each mode of transportation

| Mode of transportation | Average accuracy per mode, % |
|---|---|
| car | 92.11 |
| bus | 81.58 |
| walk | 100.0 |

the future performance of the neural network. As is evident from the results presented in this paper, selecting a proper set of inputs for the neural network and maximising the accuracy obtained with those inputs require some experimentation.

One limitation to the research presented in this paper is that the GPS data used to train and test the neural network were manually segmented by the cell phone user into trips, each of which contained a single mode of transportation. In other words, the user of the cell phone who was being surveyed indicated via input to the active diary portion of the TRAC-IT Java ME mobile application where the trip started (e.g. where they boarded a bus, where they started driving their car etc.) and where the trip ended (e.g. where they exited a bus, where they got out of their car etc.). Therefore for a given set of GPS trip data input into the neural network, that data do not contain GPS information from multiple modes of transportation. While manual segmentation is necessary to assess the feasibility of using neural network for learning automated transportation mode detection, in a completely passive survey the participant will not manually indicate where they are changing modes of transportation. Therefore research into automated trip segmentation for continuous streams of assisted GPS data is an area requiring future work. For example, if travel behaviour data were to be collected passively by a mobile software application such as TRAC-IT on a survey participant's cell phone, the resulting dataset for a day of data collection would be a completely un-segmented collection of GPS fixes representing the user's travel path for the entire day. This travel path would likely include multiple modes of transportation, since the user must walk to and from a bus or car. Before mode detection using neural networks could be performed on these data, the data should be divided into segments where each segment represents a single mode. To make automated mode detection scalable to thousands of users participating in a passive data collection effort, algorithms must be designed to automatically segment a set of GPS data into single-mode trips. The authors are currently examining automated trip segmentation via several methods, including clustering and other algorithms to identify points-of-interest. Once points-of-interest can be located, the user's daily travel path can be segmented according to these points-of-interest. Neural network accuracy for automated mode detection should increase as the precision of automated trip segmentation increases, since less noise (e.g. data from multiple modes within a single trip example used for training or testing) would exist for trips segmented at the correct location.

Since the TRAC-IT system is currently in a working prototype state, additional funding should be sought to test the system on a larger scale. By deploying the TRAC-IT mobile application to Java ME-enabled cell phones, a large amount of travel behaviour data could be captured over an extended amount of time. If real-time travel information services such as personalised traffic alerts are given to the user based on their real-time location, this additional information may serve as enough incentive for the user to consent to being tracked over years, instead of days as is normally captured by traditional travel surveys.

## 5 Conclusion

Next-generation transportation surveys will utilise GPS to collect trip data. Owing to their ubiquity, GPS-enabled mobile phones are developing into a promising survey tool that can also provide the participant with useful real-time services (e.g. traffic alerts for incidents along their predicted travel path) as an incentive to participate in travel surveys. As demonstrated in this research paper, automatic mode detection is feasible when utilising a neural network, and assisted GPS data collected via a mobile application such as TRAC-IT for GPS-enabled mobile phones. Furthermore, mode detection accuracy was actually improved when only a small subset of GPS coordinates required to re-create the user's path (i.e. critical points) were used as neural network input. Pre-filtering of GPS data using the critical point algorithm effectively reduced the amount of noise in data fed into the neural network and improved proper classification of the mode of transportation. By transferring only critical points instead of the entire GPS dataset from a phone to a server for analysis, TRAC-IT reduces the amount of data sent over the cellular network, thereby saving battery energy, data transfer costs, network bandwidth and storage space. A new set of data attributes for neural network input had to be developed for the critical points-only datasets since the reduction in available GPS data makes the calculation of some attributes used for the full GPS datasets impossible. The highest accuracy accomplished for mode detection using 10-fold cross-validation was 91.23% for the critical points-only dataset using a neural network-learning rate of 0.1 and training time of 300 epochs. When broken down by mode, the neural network correctly predicted 92.11% of the car trips, 81.58% of the bus trips and 100% of the walking trips for this test series.

Future efforts will focus on trying to identify additional data attributes that will better differentiate car against bus trips. A limitation of the research presented in this paper is that the GPS data used for training and testing of the neural network were manually segmented by the survey participant into trips, each consisting of a single mode of transportation. Future work should focus on identifying methods of automatically segmenting assisted GPS datasets representing an entire day of travel behaviour into single-mode trips. Once GPS data are automatically segmented, automated mode detection can occur on passively recorded trips which have not been manually segmented by the survey participant. Additionally, a larger deployment of the TRAC-IT Java ME mobile application to the public is desirable to assess scalability issues for the system, as well as generate larger datasets. These datasets can be used for further training and validation of automatic mode detection using neural networks, as well as the advancement of the general field of travel behaviour data collection and analysis.

## 6 Acknowledgments

This work was supported in part by the National Science Foundation under grant number 0453463, the Florida Department of Transportation and the United States Department of Transportation through the National Center for Transit Research under grant numbers BD-549-24 'Testing the Impact of Personalised Feedback on Household Travel Behaviour (TRAC-IT Phase 2)' and BD-549-35 'Smart Phone Application to Influence Travel Behaviour (TRAC-IT Phase 3)'.

## 7 References

[1] MURAKAMI E., WAGNER D.P., NEUMEISTER D.M.: 'Using global positioning systems and personal digital assistants for personal travel surveys in the United States'. Int. Conf. Transport Survey Quality and Innovation, Grainau, Germany, 1997, available at: http://gulliver.trb.org/publications/circulars/ec008/session_b.pdf, accessed July 2004

[2] MURAKAMI E., WAGNER D.P.: 'Can using global positioning system (GPS) improve trip reporting', *Transp. Res. Pt C*, 1999, **7**, (2/3), pp. 149–165

[3] GOULIAS K., ROSSI T.: 'Data needs for innovative modeling transportation'. Workshop Summary, Research Circular E-C071: Data for Understanding Our Nation's Travel National Household Travel Survey Conf., Transportation Research Board, 2004, pp. 56–59



doi: 10.1049/iet-its.2009.0029

[4] STOPHER P.: 'Collection and processing of data from mobile technologies'. Paper submitted to the Eighth Int. Conf. Survey Methods in Transport: Harmonization and Data Comparability, Annecy, France, May 2008

[5] WOLF J., LEE M.: 'Synthesis of and statistics for recent GPS-enhanced travel surveys'. Paper submitted to the Eighth Int. Conf. Survey Methods in Transport: Harmonization and Data Comparability, Annecy, France, May 2008

[6] AULD J., WILLIAMS C.: 'Activity-travel surveying using GPS technology'. Presentation to the New Technology Subcommittee at the Transportation Research Board 87th Annual Meeting, Washington, DC

[7] Federal Communication Commission (FCC). 'Enhance 911 – Wireless Services.' Retrieved 10/8/2006 at: http://www.fcc.gov/911/enhanced/, accessed June 2008

[8] VAN DIGGELEN F.: 'Indoor GPS theory & implementation'. IEEE Position, Location & Navigation Symp., 2002, pp. 240–247

[9] Qualcomm: 'Qualcomm announces MSM6125 chipset with features to enable doubling of system capacity and advanced position-location capabilities'. Available at: http://www.qualcomm.com/news/releases/2005/051003_msm6125.html, accessed October 2005

[10] Telespial Systems Inc.: 'Super Trackstick Tech Specs'. Available at: http://www.trackstick.com/products/supertrackstick/supertrackstick.html, accessed June 2009

[11] KVM Australia: 'KVM 8 Meg tracking Bluetooth GPS: GPS-BTT-08'. Available at: http://www.kvm.com.au/store/product.aspx?ProductID=2177&CategoryID=387, accessed June 2009

[12] QStarz: 'Super 51-CH performance Bluetooth GPS travel recorder: BT-Q1000'. Available at: http://www.qstarz.com/Products/GPS%20Products/BT-Q1000.html, accessed June 2009

[13] GlobalSat: 'DG-100 GPS data logger'. Available at: http://www.qstarz.com/Products/GPS%20Products/BT-Q1000.html, accessed June 2009

[14] BARBEAU S., LABRADOR M., GEORGGI N., WINTERS P., PEREZ R.: 'TRAC-IT: a software architecture supporting simultaneous travel behavior data collection and real-time location-based services for GPS-enabled mobile phones'. Proc. National Academy of Sciences' Transportation Research Board 88th Annual Meeting, January 2009, Paper #09-3175

[15] Sun Microsystems, Inc.: 'Java specification request (JSR) 179: location API for J2ME™'. Available at: http://jcp.org/en/jsr/detail?id=179, accessed June 2008 (© Sun Microsystems, Inc., 2007)

[16] OHMORI N., NAKAZATO M., HARATA N., SASAKI K., NISHII K.: 'Activity diary surveys using GPS mobile phones and PDA'. Proc. National Academy of Sciences' Transportation Research Board 85th Annual Meeting, January 2006, Paper #06-3039

[17] BARBEAU S., LABRADOR M., PEREZ A., WINTERS P., GEORGGI N., AGUILAR D., PEREZ R.: 'Dynamic management of real-time location data on GPS-enabled mobile phones'. UBICOMM 2008 – The Second Int. Conf. Mobile Ubiquitous Computing, Systems, Services, and Technologies, Valencia, Spain, September/October 2008

[18] BYON Y., ABDULHAI B.: 'Impact of sampling rate of GPS-enabled cell phones on mode detection and GIS map matching performance'. Transportation Research Board Annual Meeting 2007, Paper #07-1795

[19] HAYKIN S.: 'Neural networks: a comprehensive foundation' (Prentice-Hall, 1998, 2nd edn.)

[20] University of Waikato: 'Weka 3: data mining software in Java'. Available at: http://www.cs.waikato.ac.nz/ml/weka/, accessed June 2008

[21] KOHAVI R.: 'A study of cross-validation and bootstrap for accuracy estimation and model selection'. Proc. 14th Int. Joint Conf. Artificial Intelligence, August 1995