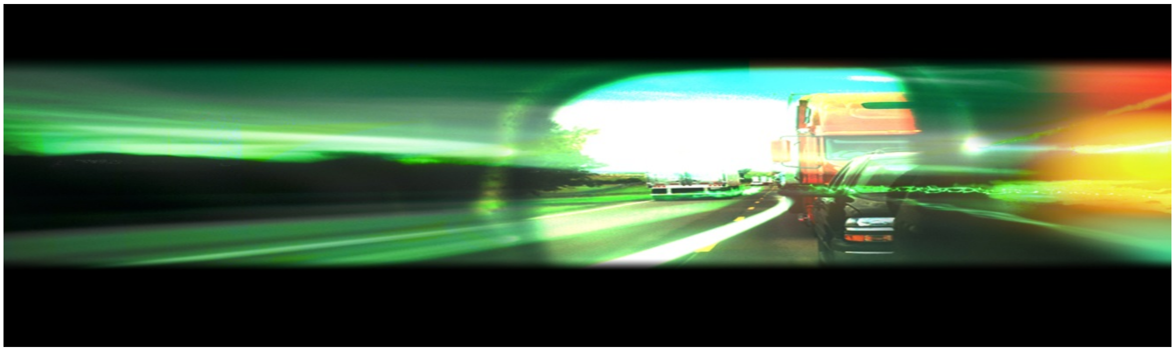
SMARTPHONE-BASED SOLUTIONS TO MONITOR AND REDUCE FUEL CONSUMPTION AND CO2 FOOTPRINT

Final Report



Dr. Mecit Cetin, Ilyas Ustun, Dr. Tamer Nadeem, Dr. Duc Nguyen
Old Dominion University

Dr. Hesham Rakha
Virginia Tech

06/2016

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

1. Report No.

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Smartphone-based Solutions to Monitor and Reduce Fuel Consumption and CO2 Footprint

5. Report Date
June, 2016

6. Performing Organization Code
KLK900-SB-001

7. Author(s)
Cetin, Dr. Mecit; Ustun, Ilyas; Nadeem, Dr. Tamer; Nguyen, Dr. Duc; and Rakha, Dr. Hesham.

8. Performing Organization Report No.
N16-01

9. Performing Organization Name and Address

Transportation Research Institute (TRI)
Old Dominion University
4111 Monarch Way, Suite 204
Norfolk, VA 23506

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
DTRT12GUTC17

12. Sponsoring Agency Name and Address

US Department of Transportation
Research and Special Programs Administration
400 7th Street SW
Washington, DC 20509-0001

13. Type of Report and Period Covered
Final Report: Jan 1, 2014–Jan 31, 2016

14. Sponsoring Agency Code
USDOT/RSPA/DIR-1

15. Supplementary Notes:

16. Abstract

Smartphones equipped with GPS and several low-energy sensors (e.g., gyroscope, compass, and accelerometer) can provide a medium to collect probe data. As smartphone users navigate the transportation networks, their travel modes and trajectories can be inferred to estimate fuel consumption and CO_2 footprint. The specific goals of the proposed research are: (i) Develop new algorithms to estimate the mode of travel (walking, biking, train, car, bus, etc.) and operating mode of a vehicle (e.g., idling) based on low-energy sensors available within smartphones; (2) Evaluate the effectiveness of FC and CO_2 estimation from probe vehicles at various market penetration levels; and (3) Develop shortest paths algorithms for finding eco-friendly routes. To achieve these goals, various methodologies are developed and tested with both simulation and field data. For example, machine learning algorithms (e.g., support vector machines) are developed to predict the travel mode and to detect whether a vehicle has stopped. The results show that the travel mode can be detected accurately, about 94% on average, when considering all five travel modes within the sample data. Using the accelerometer data only, the results show that the models can accurately detect the times at which the vehicle stops and moves during its journey. To predict the impacts of probe vehicle market penetration on estimating fuel consumption, simulation data are created for an intersection. Lastly, the shortest path (SP) algorithm for static networks are modified so that the algorithm can find the SP for minimizing both the travel time and fuel consumption for a given network. Together, the models and algorithms developed in this study can be integrated to support various mobility and environmental applications.

17. Key Words
Smartphones, Machine Learning, CO2, Fuel Consumption.

18. Distribution Statement

Unrestricted; Document is available to the public through the National Technical Information Service; Springfield, VT.

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
37

22. Price
…

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The value of trajectory- or probe vehicle-based information is very well-known as such information allows accurate characterization of vehicle and traffic dynamics, which ultimately enable various mobility and environmental applications (e.g., eco-routing) to reduce Fuel Consumption (FC) and emissions. Smartphones equipped with GPS and several low-energy sensors (e.g., gyroscope, compass, and accelerometer) can provide a medium to collect probe data. As smartphone users navigate the transportation networks, their travel modes and trajectories can be inferred to estimate fuel consumption and CO2 footprint.

As smart-phone market is expanding rapidly, various applications supporting travelers' decision making have been developed. Most of these applications pertain to access or dissemination of traffic congestion information and transit schedules and routes. However, there is very limited work on investigating the use of smartphones for monitoring and providing emission and FC information. The overall goals of this project are to develop smartphone-based solutions and algorithms to estimate FC and CO_2 footprint so that feedback to a driver can be provided for a given multi-modal trip. The specific goals of the proposed research are:

1. Develop new algorithms to estimate the mode of travel and operating mode of a vehicle (e.g., idling, accelerating) based on low-energy sensors (e.g., gyroscope, compass, and accelerometer) available within all smartphones. Even though researchers used cell phones for trajectory estimation before, these relied on GPS which has the disadvantages of low accuracy in urban canyons and depleting the phone battery quickly.
2. Evaluate the effectiveness of fuel-consumption and CO_2 estimation from probe vehicles at various participation or market penetration levels.
3. Develop shortest paths algorithms for finding eco-friendly routes in near real-time in large-scale transportation networks.

The proposed research relies on the capabilities available within smartphones. A collection of robust algorithms is developed to extract useful information from the ***low-energy sensors*** while making minimal or no use of the GPS sensor.

To achieve these goals, various methodologies are developed and tested with both simulation and field data. For example, machine learning algorithms (e.g., support vector machines) are developed to predict the travel mode and to detect whether a vehicle has stopped. The results show that the travel mode can be detected accurately, about 94% on average, when considering all travel modes (bike, car, walk, run, bus) within the sample data. Using the accelerometer data only, various models are built to predict when a vehicle stops and when it is in motion. When tested on field data, the results show that these models can accurately detect the times at which the vehicle stops and moves during its journey. To predict the impacts of probe vehicle market penetration on estimating fuel consumption, simulation data are created for an intersection. The results show that estimation error drops dramatically as the level of market penetration increases from zero to approximately 20% beyond which the error improves only marginally. Lastly, the shortest path (SP) algorithm for static networks are modified so that the algorithm can find the SP for minimizing both the travel time and fuel consumption for a given network. Together, the models and algorithms developed in this study can be integrated to support various mobility and environmental applications including Dynamic Low Emissions Zones Concept of the AERIS program. They can also be used to build applications to give feedback to the traveler in terms of FC and CO2 emissions for each completed trip.

DESCRIPTION OF PROBLEM

According to the newest statistics, the transportation sector accounted for 27% of our nation's greenhouse gas (GHG) emissions in 2011 [1], almost all in the form of CO_2 from fossil fuel combustion. In addition, billions of gallons of fuel are wasted every year due to congestion. To address these challenges, transportation decision makers at all levels need data to understand how emissions vary over the network links by the time of day. To support that, this project aims to develop new solutions to enable collection of vast amount of dynamic travel data (e.g., operating modes) in large networks by leveraging the rich data from low-energy sensors within the smartphones while making minimal or no use of the GPS sensor. In addition, the proposed smartphone application will provide useful feedback to the user in terms of accurate FC and CO_2 emissions for each completed trip. If the user takes different routes at different times of the day between the same origin-destination points, he/she can then assess the relative fuel consumptions and CO_2 emissions for the alternative travel options and make trip decisions while considering these factors. Last but not least, new shortest path algorithms will be developed for finding eco-friendly routes in near real-time in large-scale transportation networks. Overall, the algorithms and solutions developed in this project will support various mobility and environmental applications including Dynamic Low Emissions Zones Concept of the AERIS program.

Research Approach: The proposed research relies on the capabilities available within smartphones. There are numerous motivating factors for this: (i) The proliferation of mobile devices (e.g., more than 50% of subscribers now have smartphones) with ever-increasing computing, sensing, and communication capabilities; (ii) The ability for rapid deployment and ease of software updating with the current App-store system; (iii) The high-power consumption of GPS-based sensing which hampers reliance on GPS-based sensing; (iv) The low accuracy of GPS in urban canyons; (v)The ability to collect a large-amount of data at a very low cost; and (vi) The ability to collect travel data across all modes with smartphones. Furthermore, the low energy sensors are always on within the phone and can provide data at high frequency (e.g., 20Hz). Even though researchers used cell phones for trajectory estimation before, these relied on GPS, which has the aforementioned disadvantages inhibiting large-scale usage.

A collection of robust algorithms is developed to extract useful information from the ***low-energy sensors*** while making minimal or no use of the GPS sensor. For example, Figure 1shows variation in accelerometer data by the mode of transport. By developing robust and intelligent algorithms that can assimilate noisy and vast data efficiently, it is possible to determine mode of travel, vehicle operating mode (e.g., idling, accelerating), and the amount of time spent within each mode. Such information, which currently does not exist at a large scale, will be invaluable for accurately estimating GHG and other emissions. Furthermore, by relating such data to links and nodes of a transportation network, which can be accomplished by minimal GPS points (i.e., turning GPS on just for several seconds) combined with the information obtained from low-energy sensors, the system operators can determine areas with inefficient operations and higher emissions.

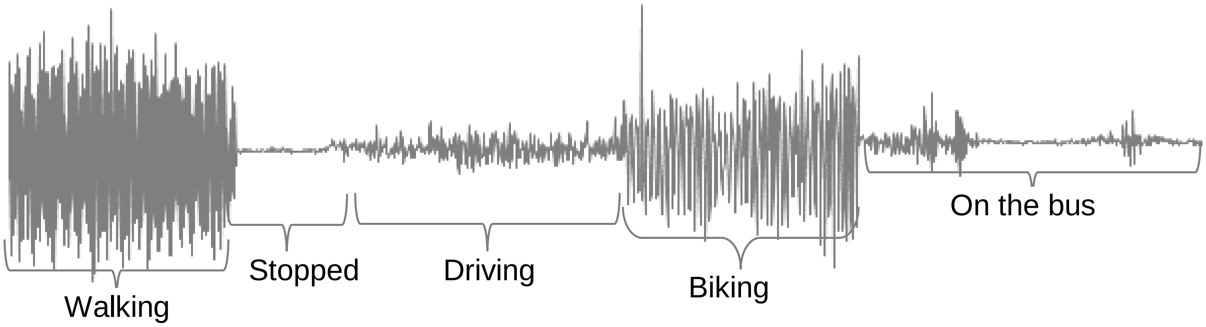


Figure 1 Sample accelerometer data collected while using different modes of travel

Finally, the research team has developed new shortest paths algorithms for finding eco-friendly routes in near real-time for large-scale transportation networks. For these eco-friendly routes to have a measurable impact on the environment, it is critical that these routes can be determined as quickly as possible, preferably in (near) real-time, e.g. by an in-vehicle route navigation system. To this end, the research team has designed improved data structures to more efficiently store and retrieve the massive amount of pertinent traffic network data, such as road network topology and emission information. In addition, novel strategies are also developed to exploit the increasingly common multi-core processors. Particularly, new methods are developed to divide our shortest path algorithm over multiple processors, in order to make use of parallel computing in reducing the computational time.

APPROACH AND METHODOLOGY

Smartphone Sensors

The sensors such as accelerometers, and gyroscopes are micro electro mechanical systems (MEMS) which are embedded within smartphones [2]. The smartphone accelerometer sensor, for example, is a sensor with x, y, and z axes orthogonal to each other as shown in Figure 2. The advantages of a smartphone is that it can log the data obtained from sensors, process them, and send and receive data through various channels (e.g. Wi-Fi, Cellular Network, Bluetooth). The ubiquitous use of smartphones by mobile subscribers qualify them to be a viable method for data collection.

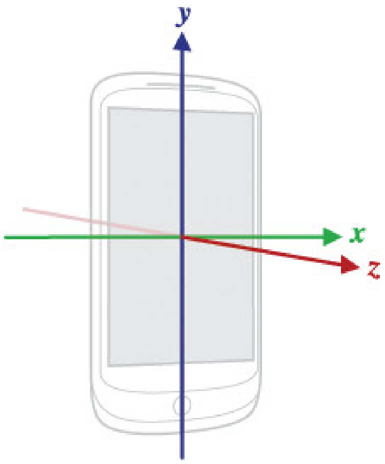


Figure 2 Smartphone sensor axes directions.

Many studies have used Global Positioning System (GPS) data for classification purposes. However, several limitations are associated with the use of GPS sensors. These limitations include: GPS information is not available in shielded areas (e.g. tunnels) and the GPS signals may be lost especially in high dense locations, which results in inaccurate position information. Moreover, the GPS sensor consumes significant power that sometimes users turn it off to save the battery [3, 4].

This project focuses on developing detection models using machine learning techniques and data obtained from smartphone sensors including accelerometer, gyroscope, and rotation vector, without GPS data. Transportation mode detection and vehicle motion detection are explained next.

Classification Algorithms for Detecting Vehicle Stops

Here we are focused on developing machine learning algorithms to extract useful traffic information from crowdsourced data. In particular, high-resolution accelerometer data collected by smartphones onboard vehicles are analyzed, and advanced classification algorithms are developed to reliably detect vehicle stops (e.g., at traffic signals). Support Vector Machines (SVMs), Hidden Markov Models (HMMs), and Changepoint Detection Methods (CDMs) are employed to develop reliable algorithms for stop detection. These algorithms are optimized based on the field data collected by a custom Android application. The results demonstrate that both SVM-HMM and CDM models are effective in detecting stops with minimal false alarms. Overall, this research demonstrates the feasibility of collecting useful performance measures at arterials which are important for improving system operations

Figure 3 shows a sample acceleration and speed data measured via an on-board diagnostics (OBD) device. Since the objective is to detect stops, the speed data are mapped to two discrete states: motion (state *M*) and stationary (state *S*). The state is stationary when the OBD speed (or GPS speed) is below a specific threshold (e.g., 2 mph), and is motion otherwise. The machine learning algorithms aim to take the raw acceleration data and estimate the true state.

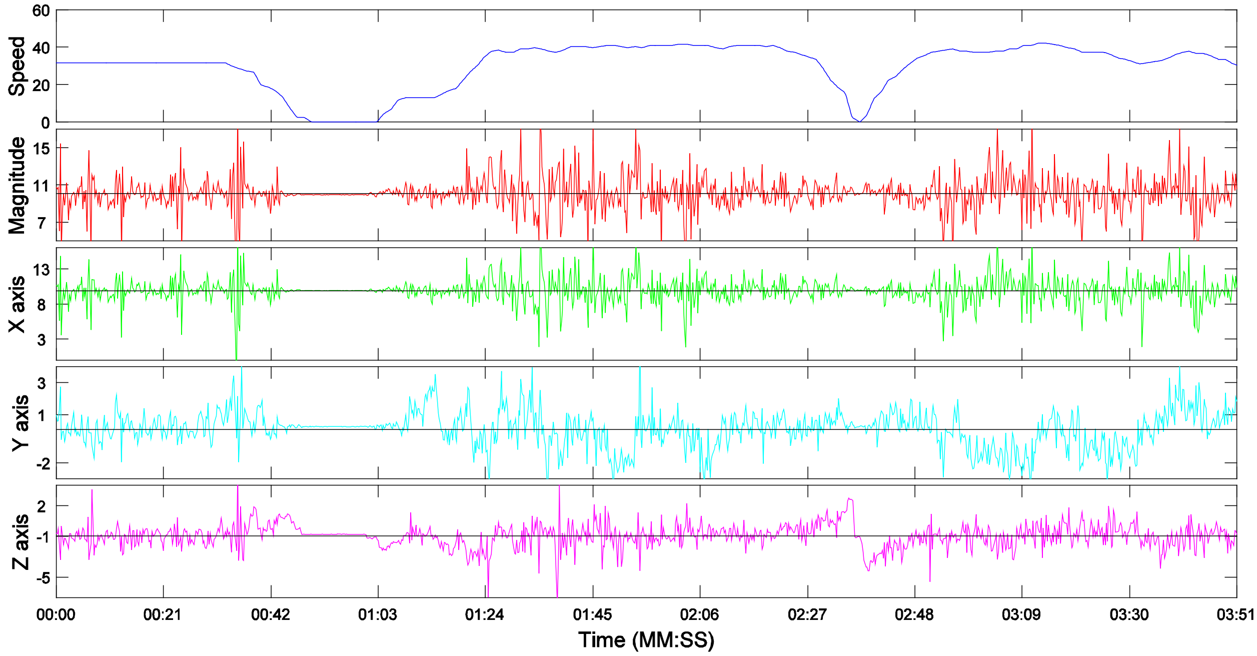


Figure 3 Vehicle speed and the three-axis accelerometer data collected by the smartphone for a trip.

Since the main objective of this research is to detect whether a vehicle is in motion (state *M*) or stationary (state *S*), at any given time from the time-series data, the vehicle can be in either one of the two discrete states (*M* or *S*) which implies that binary classification method is needed. Overall, there are two potential approaches to estimate whether the vehicle is in state *M* or state *S* from the onboard smartphone accelerometer data. These two options are:

1. Periodic analysis of the time series data: In this case, the time series data (e.g., accelerometer data) are analyzed periodically (e.g., every second or several seconds) to determine whether the vehicle is in motion or has come to a stop. Essentially, data from each (short) interval are treated independently of other intervals and a decision is made via any binary classification algorithm (e.g., SVM, neural networks). In this paper, algorithms based on SVMs are developed to periodically estimate the "state" of the vehicle. These estimated states are further processed through a Hidden Markov Model (HMM) to account for the correlation between consecutive time periods. This method is abbreviated as SVM-HMM and may be more suitable for online or real-time application as explained below.

2. Analysis of complete or segments of time series data: In this case, the entire time series data are analyzed at once to estimate the time instances when the vehicle stops. Basically, the changepoints or breakpoints when any state transition occurs (from M to S or vice versa) are predicted. To find such points in the accelerometer data, algorithms based on Changepoint Detection Methods (CDMs) are developed to estimate the state of the vehicle as also explained below.

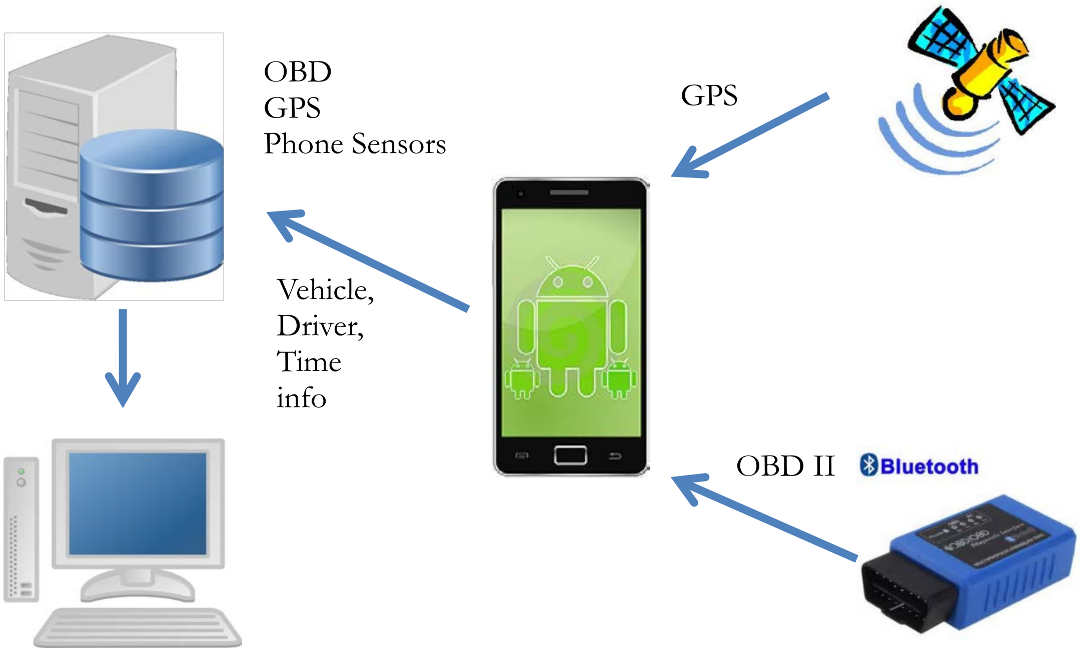


Figure 4 The overall view of the data collection system

Data Collection: Field data are collected by a smartphone, and an on-board diagnostics (OBD) device. The OBD device used is capable of transmitting data via Bluetooth to the smartphone. An Android application is developed which records the data from the smartphone sensors, logs GPS readings and the speed data transmitted from the OBD. The overall scheme of data collection and processing is shown in Figure 4. Each sensor has its own data sampling rate, and the app is set to log data from each sensor at the highest rate allowed. From the field data collected, it is found that the sampling rate for accelerometer sensor can be as low as 1 sample per second and as high as 238 samples per second, with the majority at 15 samples per second. In general, the more the forces exerted on the phone, the higher the sensor activity is. The GPS and OBD data collection rate is 1 sample per second. In order to fix the time intervals between samples to a certain duration, these separate datasets from accelerometer, GPS, and OBD are first interpolated with a common start and end time and are resampled with a chosen sampling rate, and then appended together which yields the raw data. The rows represent each sample point, and the columns represent each sensory data of that sample. The OBD speed data are used as ground truth for detecting when the vehicle stops and starts moving and for model training and testing purposes. The data collection process involved normal driving on arterial streets with signalized intersections.

The magnitude of the accelerometer data or total acceleration is computed from each accelerometer sample to allow for classification of the data with any arbitrary phone orientation as shown below.

$$acc_{Tot} = \sqrt{\left(acc_x{}^2 + acc_y{}^2 + acc_z{}^2\right)}$$

The accelerometer sensor works on the principle of forces being exerted on a rigid body causing acceleration; hence it will measure gravity even when the phone is completely stationary. Since it is almost impossible for a phone to be perfectly set in a vehicle such that one axis will be affected by gravity; one axis by motion; and one axis by lateral movements alone, phone orientation correction algorithms need to be applied to mitigate the effect of gravity on each direction and align one of the phone axis with the motion direction of the vehicle. In order to eliminate this phase, and also make the vehicle motion detection simpler and applicable to any orientation, total acceleration proves to be a convenient method. The total acceleration is defined as the square root of the sum of squares of acceleration values measured in each direction. Several features such as mean, range, standard deviation and percentile scores are extracted from the acceleration magnitude dataset. These features are used as inputs to the classification techniques.

The results are summarized graphically in Figure 3. Here, the blue line represents the true observation, green dashed line represents the estimation obtained by applying SVM-HMM, and the red dotted line is the estimation from Changepoint Detection methods. The outputs for 7 test trips are shown. When the line for a given trip is in the lower level it means standstill, while the upper level means the vehicle is in motion. Both SVM-HMM and CDM models are found to perform well on the data analyzed in this research.

This research demonstrates how alternative machine learning algorithms can be developed to extract useful traffic information from crowdsourced data. In particular, high-resolution accelerometer data collected by smartphones onboard vehicles are analyzed, and advanced classification algorithms are developed to reliably detect vehicle stops. Support Vector Machines (SVMs), Hidden Markov Models (HMMs), and Changepoint Detection Methods (CDMs) are employed for algorithm development. These algorithms are trained and optimized based on the training data. The SVM-HMM model applies the classification rule sequentially to the data whereas the CDM takes a segment of data and estimates the breakpoints to separate the data-points collected when vehicle is in motion from those when it is stationary. Both SVM-HMM and CDM have two key parameters which are optimized by a grid search performed based on the training data from seven trips.

SVM-HMM and CDM models are tested on the field data; the results demonstrate that both models are effective in detecting the stops with minimal false detections. Overall, the results demonstrate the feasibility of extracting useful performance measures at arterials which are important for improving system operations. The model developed here can be the basis for developing algorithms to detect stops from large accelerometer data produced by thousands of smartphones onboard vehicles in the traffic stream. Future work can include applying the developed algorithms to larger datasets collected by various types of smartphones and vehicles.

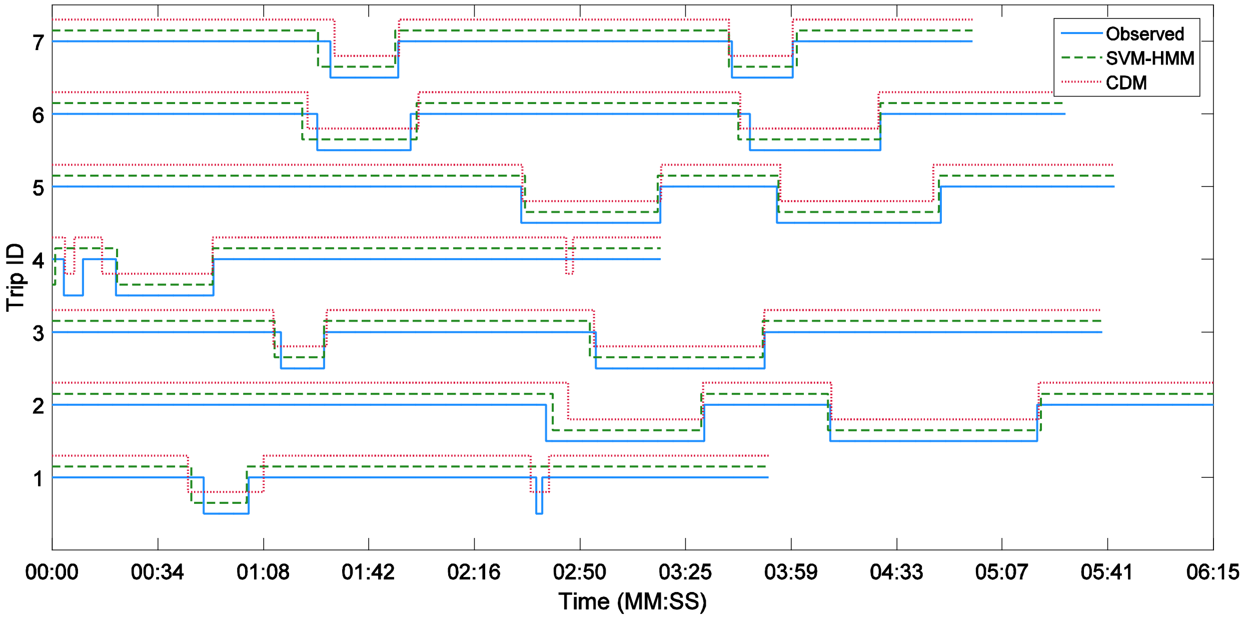


Figure 5 Performance of the SVM-HMM and CDM models when tested on new datasets.

Transportation Mode Recognition

This research adopts different supervised learning methods from the field of machine learning to develop multi-class classifiers that identify the transportation mode, including: driving a car, riding a bicycle, riding a bus, walking, and running. Methods that were considered include K-Nearest Neighbor (KNN), Support Vector Machines (SVMs), and tree-based models that comprise a single Decision Tree (DT), Bagging (Bag), and Random Forest (RF) methods. For training and validating purposes, data were obtained from smartphone sensors including accelerometer, gyroscope, and rotation vector sensors. K-fold Cross-Validation as well as Out-of-Bag error was used for model selection and validation purposes. Several features were created from which a subset was identified through the minimum Redundancy Maximum Relevance (mRMR) method. Data obtained from the smartphone sensors were found to provide important information to distinguish between transportation modes. The performance of different methods was evaluated and compared. The Random Forest (RF) and Support Vector Machine (SVM) methods were found to produce the best performance. Furthermore, an effort was made to develop a new additional feature that entails creating a combination of other features by adopting a Simulated Annealing (SA) Algorithm and Random Forest (RF) method.

Transportation mode detection can be considered an activity recognition task in which data from smartphone sensors are utilized to infer the mode of transportation. Knowledge of the transportation mode can be useful in several applications:

1- Knowing the transportation mode is an essential part of urban transportation planning, which is usually investigated through questionnaires, travel diaries and telephone interviews [5, 6]. These are expensive, erroneous, limited to a specific area, and not up-to-date [3].

2- Knowing the transportation mode can provide accurate information of an individual's carbon footprint and the amount of calories burnt in case of walking or biking [7].

3- Knowing the mode of transportation and speed can be utilized to provide users with real time information such as congestion along the route, or specific ads targeted at specific mode travelers.

In detecting the mode of transportation several techniques are used including K-Nearest Neighbor (KNN), Support Vector Machines (SVMs), and tree-based models that comprise a single Decision Tree (DT), Bagging (Bag), and Random Forest (RF) methods to identify transportation modes using data obtained from smartphone sensors.

The research carried out about mode detection is more comprehensive and complete to previous research in that it considers both motorized and non-motorized modes of travel. It also does not require the travelers maintain a fixed location for their phone. GPS is avoided as well which is almost always used in other papers.

Data Collection

A smartphone application was developed for the purpose of data collection for this part of the research. The application stores the data coming from smartphone sensors including GPS, Accelerometer, Gyroscope, and Rotation Vector at the highest possible frequency. To collect the data, ten employees at Virginia Tech Transportation Institute (VTTI) were asked to carry a smartphone (two devices were used: a Galaxy Nexus and a Nexus 4) with the application installed on it on multiple trips. They were asked to select the travel mode they intend to use before starting the logging process, and then using the application buttons they were able to start and stop data logging. Although smartphones can be carried in other places, to make sure the data collection is less dependent on the sensor positioning, the travelers were asked to carry the smartphone in different positions that they normally do such as in pocket, in palm, in backpack, and different places inside car (e.g. on front right seat, coffee holder alongside of the driver) as they reported after the data collection. However, the amount of time that was spent for different positions were unknown since the participants were not asked to collect data in a particular position for a certain amount of time and the reason was to make the data collection as natural as possible. Data collection was conducted on different workdays (Mon through Fri) during working hours (8 AM to 6 PM) on different road types with different speed limits (i.e. car mode on roads with 15, 25, 35, 45, and 65 mph; bus mode on roads with 15, 25, 35, and 45 mph; bike mode on roads with 15, 25, and 35 mph) in Blacksburg, Virginia. Thirty minutes worth of data for each mode per person were collected. The original data frequency was about 25 Hz (for accelerometer, gyroscope, and rotation vector sensors), but the data from different sensors were not synchronized. Thus, in order to ensure that the data were gathered at identical sampling rates, linear interpolation was first applied to the data similar to [4] to produce continuous data sets and then the data were re-sampled at the desired rate (rate of 100 Hz was applied). Since the original frequency of 25 Hz was not a constant rate (i.e. a constant frequency was not possible to set for collecting data), the choice of 100 was made to make sure no information is lost. Furthermore, a low pass filter was used for noise reduction. In total, 25 hours of data (30 minutes per mode per person) were stored and used for training and testing purposes. In other words, total of ten travelers collected 30 minutes of data for each mode that equals (30x10x5)/60 = 25 hours.

Several features were extracted from the data collected. Out of the many features that were calculated the most important ones are shown in Table 1. Here *a* represents accelerometer data, and g represents gyroscope date from the smartphone. A total of 80 features were identified as the most relevant features by mRMR method. The importance of the features was assessed based on two measures: (1) Mean Decrease Accuracy that shows how the detection accuracy is decreased if a feature was excluded, averaged over all trees, and normalized by the standard deviation of the differences in accuracy and (2) Mean Decrease Gini that shows how a single feature contributed to decrease the Gini index over all the trees. Since the two measures determine the feature importance in different ways the identified features by the two measures are different. While both measures have been used in the literature, there have been arguments concerning the preference for one measure over another. It is recommended that the first method (i.e. Mean Decrease Accuracy) is more suitable for causal interpretations. More details about the arguments and some contradictions regarding these measures can be found in [8].

Table 1. Important Features

No.	Feature Name	No.	Feature Name
1	$spectralEntropy(a_x)$	11	$mean(a_z)$
2	$range(a_y)$	12	$iqr(\dot{a}_x)$
3	$max(a_y)$	13	$var(\dot{g}_x)$
4	$max(g_y)$	14	$min(\dot{a}_y)$
5	$min(g_y)$	15	$range(\dot{a}_x)$
6	$range(\dot{g}_x)$	16	$energy(a_x)$
7	$spectralEntropy(a_y)$	17	$range(g_x)$
8	$max(a_z)$	18	$mean(g_z)$
9	$mean(\dot{g}_x)$	19	$std(\dot{a}_y)$
10	$min(\dot{a}_z)$	20	$spectralEntropy(g_x)$

Research within this area has made use of several artificial intelligence tools such as Fuzzy Expert Systems as in [9], Decision Trees as in [3-7, 10], Bayesian Networks as in [6, 10], Random Forests as in [6], Naïve Bayesian techniques as in [6, 7], Neural Networks as in [11, 12], and Support Vector Machine (SVM) techniques as in [7, 10, 13-16], of which the Decision Tree and SVM methods were used the most. To improve the model performance, other techniques were also combined with machine learning methods such as Discrete Hidden Markov Models as in [7] and Bootstrap aggregating as in [17]. Other than AI tools, statistical methods were also applied such as the Random Subspace Method in [18]. Some studies have used additional information from Geographic Information System (GIS) maps as in [6, 9, 19, 20]. However, GIS data is not always available, and also this approach may not be suitable for real-time applications because it mostly relies on the knowledge of the entire trip with respect to the GIS features such as bus stops, subway entrances, and rail lines.

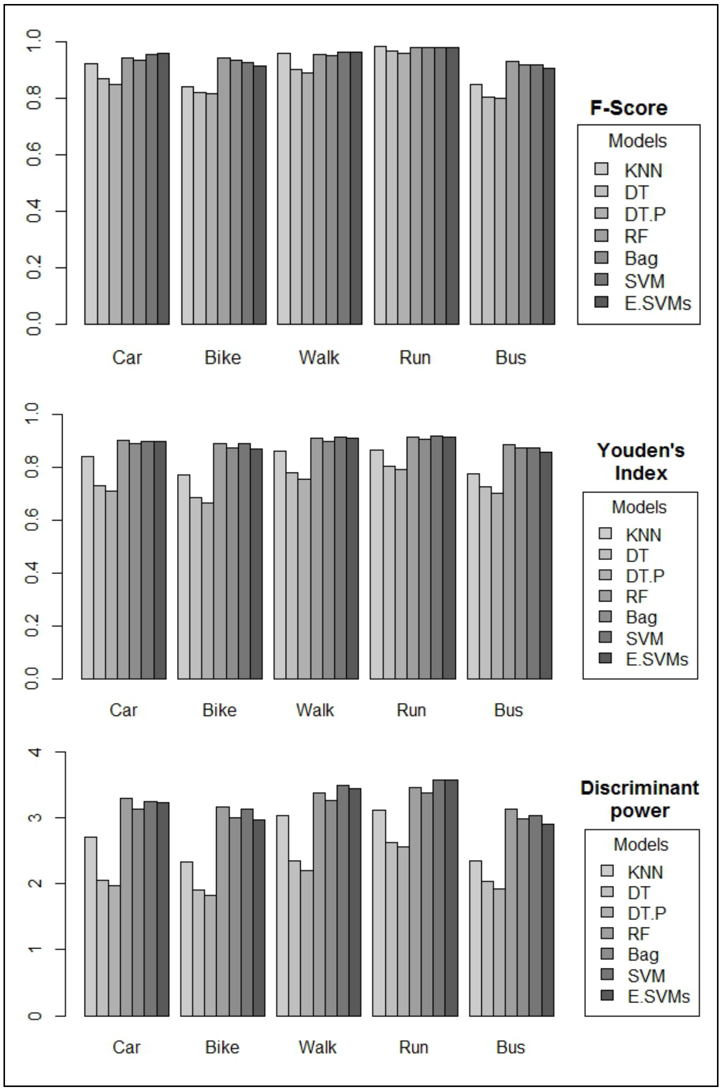


Figure 6 Model comparison results for mode identification

The performance of the models was evaluated using four metrics, namely: the overall accuracy, the F-Score, Youden’s index, and the Discriminant Power (DP). The overall accuracy is calculated by dividing the total number of correct detections by the total number of test data. The F-Score is a combined measure of the Recall and the Precision. The Youden’s index is a measure to assess the ability of a model to avoid failure. The discriminant power shows how well a model discriminates between different classes by summarizing sensitivity and specificity of the model; the model is a poor discriminant if DP <1, limited if DP <2, fair if DP <3, good – in other cases. The sensitivity and specificity assess model performance on a single class, and are equivalent to the recall. By definition, assuming two classes (positive and negative) sensitivity is exactly the same as the Recall measure. Specificity is also the same metric but for the negative class. **FIGURE 6** illustrates a visual comparison between the models using different performance measures.

Different classifiers were developed using machine learning techniques to identify different transportation modes including bike, car, walk, run, and bus. In training and testing the classifier, data were obtained from smartphone sensors such as accelerometer, gyroscope, and rotation vector which were found to have important information for the purpose of mode recognition. A time window of one second was chosen, so the model can fit in a broader range of applications. For each method, parameters that needed to be optimized were examined to conduct a complete model selection. K-fold Cross-Validation and Out-Of-Bag error were used for model evaluations. In addition, some performance measures such as the F-Score, Youden’s index, and Discriminant Power were applied to assess model performances on the individual modes. Considering misclassification rates, the car and bus modes were the most difficult ones to distinguish, as would be expected. Even using more complex models such as SVM and RF, the car mode was misclassified as the bus mode in about 4-6% of the time. The Random Forest method was found to produce the best overall performance, which is shown in Table 2. However, for specific modes (i.e. walk and Run), the SVM outperformed the RF method. Several features were created and examined; among which 80 features were identified using the mRMR method as the most relevant feature.

Other than some statistical measures of dispersion (e.g. range, max, variance, etc.), spectral entropy and energy were among the most important features.

Table 2. Confusion matrix - Random Forest

Random Forest		Actual					
		Bike	Car	Walk	Run	Bus	**Precision**
Predicted	Bike	95.47	1.46	2.63	0.97	2.29	93.06
	Car	0.37	93.84	0.12	0.05	4.47	94.93
	Walk	2.93	0.13	96.23	1.59	0.12	95.24
	Run	0.03	0.00	0.40	96.81	0.00	99.55
	Bus	1.19	4.57	0.63	0.58	93.12	93.02
	Recall	95.47	93.84	96.23	96.81	93.12	

Feature combination was investigated by employing SA algorithm and the RF method in an iterative mechanism. However, no significant improvement was obtained. Due to collecting data in a naturalistic way, inevitable situations with very similar data might have occurred, which can be the cause of the error.

Some recommendations for future directions that applies to this and similar research problems include: adding more data, applying approaches to examine the data as a sequence, considering more transportation modes (e.g. metro), and conducting error analysis to gain some insights about where different models fail to correctly classify the data and consequently incorporate that knowledge into the models to enhance the detection performance.

Distributed Learning: An Application to Transportation Mode Identification

When dealing with Machine Learning problems, the traditional way is to collect some relevant data and develop a single model in a centralized system based on the entire dataset. However, the data can be divided and different models can be developed on different portions of the data. Having a distributed system rather than a centralized system has been an active research topic in the computer science field, but other fields have also shown interest. There are several reasons why a distributed approach can be beneficial over a centralized approach, as follows [21-24]:

1. Dealing with a large amount of data in a centralized system may be too difficult to handle;
2. Data sources may be physically dispersed, and therefore too expensive to be directed to a centralized system;
3. Sometimes the data from various sources cannot be shared due to privacy, security, and data ownership issues; and
4. Sometimes it is more efficient to have learning activities in parallel.

In such conditions, a desired approach is to design a knowledge acquisition system that can analyze parts of the data wherever available and then the analysis results can be transmitted, if needed. In other words, the knowledge can be acquired from different data parts and if required the results can be aggregated [23, 24]. Furthermore, specifically in the transportation domain, recent methods of data collection such as probe vehicles have been proposed to collect traffic data as a cost-effective alternative to the more traditional approaches such as loop detectors and video cameras [25]. In fact the combination of traditional data collection methods (on-road sensors) with these new methods (on-board sensors) provide high quality datasets to be utilized [26]. In the new methods, instead of the infrastructure (e.g. loop detector, video cameras, etc.), individuals (e.g. probe vehicles, smartphones, etc.) collect the data. As a result, using these methods, there is an opportunity to analyze the data for each individual in a distributed manner and if required the analysis results can be aggregated.

The data that were used in this part of the project were from a previous effort in which a smartphone application was developed to collect the required data from smartphones [27, 28]. To collect the data, the transportation mode (Car, Bike, Bus, Walk, and Run) should be selected before starting the logging process, and then the application stores the data coming from smartphone's sensors including GPS, Accelerometer, Gyroscope, and Rotation Vector at the highest possible frequency. Data collection was carried out by five individuals. While the app collected the data from the aforementioned sensors, the data from the GPS sensor were not used in this study. About 150 minutes of data were gathered by each person while using different transportation modes. The data were equally gathered amongst transportation modes (i.e 30 minutes for each mode). A total of 750 minutes of data were stored and used for model development. There was no restriction on placing the smartphone (i.e. attaching the smartphone to part of the body) while collecting the data so the approach becomes independent of the device placement.

To compare the proposed approach against the traditional approach, machine learning methods namely Random Forest (RF) and Support Vector Machine (SVM) were applied. A single Random Forest model was developed using the entire dataset to represent the traditional approach. Also, to assess the proposed approach, five RF models were developed, each of which was based on the corresponding data obtained from each individual. To implement the RF method, the R software and RandomForest packages were used [29, 30]. SVM is known as a large margin classifier, which means when classifying data, it determines the best possible decision boundary that provides the largest possible gap between classes. This characteristic contributes to a higher confidence in solving classification problems. To implement SVM, the LibSVM library of SVMs was applied. For multiclass classification, considering K classes, LibSVM applies one-against-one method in which $K(K-1)/2$ binary models are built. Among these, LibSVM chooses the parameters that achieve the highest overall performance [31]. To be able to evaluate how the proposed approach performs compared to the traditional approach k-fold cross validation error rate and out of bag error were used when assessing the SVM and the RF models, respectively.

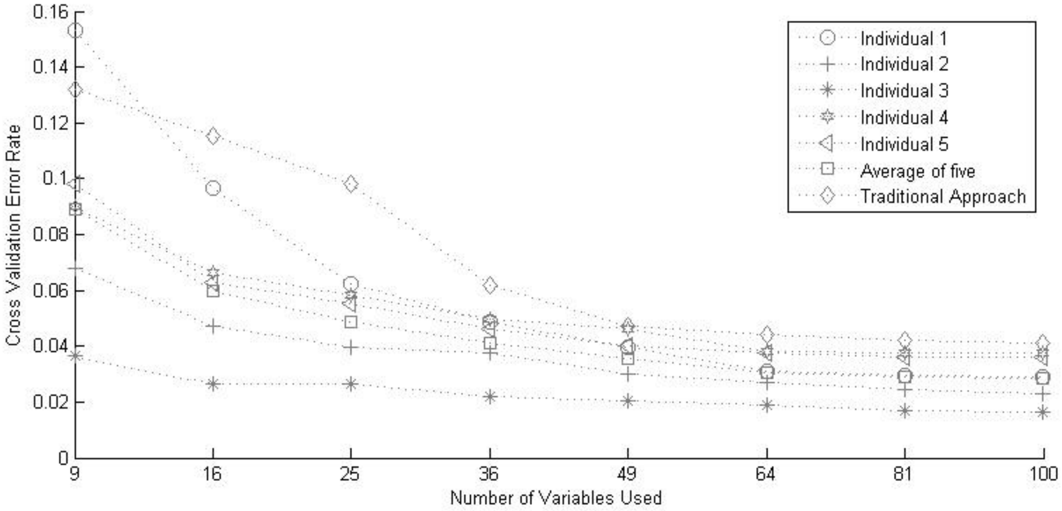


Figure 7: Accuracy comparison - SVM models

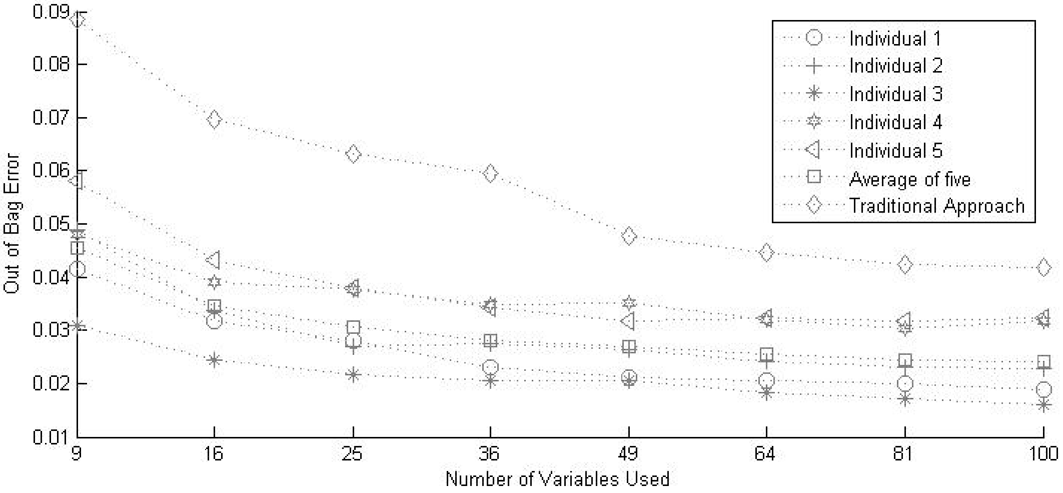


Figure 8: Accuracy Comparison - RF models

A mode identification application was used to illustrate how the proposed approach performs compared to the traditional approach. Models developed using both SVM and RF methods in the proposed approach contributed to more accurate models compared to the models in the traditional approach. Looking at the error rates of all models as shown in Figure 7 and Figure 8, all individual models as well as the average model resulted in a lower error rate compared to the model obtained from the traditional approach except in only one case when using SVM and the number of variables were 9. In this special case, it seems that the number of variables is not sufficient to develop a good model as high error rates were produced. Consequently, this case can be excluded simply because not enough information is provided to the models. It can be seen that when using 36 variables or above, the SVM models achieved low error rates which shows these models have good performance. Thus, at these points (above 36 variables) comparison can be drawn. Similarly, RF models resulted in more accurate models when the proposed approach was adopted as opposed to the traditional approach. As shown in Figure 8, the proposed approach always led to a lower error rate. Using both methods (i.e. SVM, RF), the more variables (features) were included the more accurate models were achieved. Furthermore, when having more variables, the performance of the traditional approach becomes closer to that of the proposed approach. However, the traditional approach always produces a higher error even when using 100 variables. In fact, considering all models, not much benefit is gained by including more than 49 variables.

The accuracies obtained using the proposed approach were not significantly higher than those of the traditional approach, but even small improvements are considered vital in certain applications (e.g. safety). More accurate models were obtained using the proposed approach even though a number of factors were in favor of the traditional approach, namely; a larger dataset was used in the traditional approach, and the five individuals, who collected the data, had somewhat similar characteristics as all were of similar age and same gender.

Estimating Fuel Consumption and Carbon Footprint from a Probe Sample

In order to explore how the number of probes or market penetration of probe vehicles impact the estimation of total fuel consumption and carbon footprint, a simulation study is carried out. Even though the use of probe vehicles as a traffic data source has been investigated before, for example in the context of travel time estimation [32, 33], queue length estimation [34-37], there has been limited effort at investigating the use of probe data for estimating fuel consumption levels. To estimate fuel consumption and CO2 emissions, a scenario is considered and developed where vehicles travel through a signalized intersection. More specifically, a systematic approach is presented to predict the fuel consumption level using vehicle trajectory data and (possibly) the make and model of each probe vehicle. The analysis is based on microscopic traffic simulation data, and an instantaneous fuel consumption model developed at Virginia Tech: the Virginia Tech Comprehensive Power-Based Fuel Consumption Model (VT-CPFM). The VT-CPFM model utilizes instantaneous power (computed from instantaneous speed and acceleration measurements) as an input variable to estimate fuel consumption levels [38]. The model can be calibrated using publicly available fuel economy data (i.e., EPA's published city and highway fuel ratings for specific vehicles). Thus, the calibration of model parameters does not require gathering any vehicle-specific data. In summary, the specific goal of this effort is to develop procedures to estimate the intersection-wide fuel consumption level and carbon footprint using a sample of probe vehicle trajectories. This approach will allow for the identification of environmental hotspots for the development of emission mitigation strategies.

Modeling fuel consumption requires specifying the vehicle mix in the traffic stream. This study does not attempt to model the vehicle mix for a real-world setting. However, the selected and modeled vehicle types represent enough variation in terms of fuel consumption rates which allow for the generalization of the study conclusions. Moreover, since CO2 emissions are directly proportional to fuel consumption, the results are presented only in terms of fuel consumption.

The overall objective of this study is to assess the relationship between sample size of probe vehicles and the accuracy of the estimated Fuel Consumption (FC) at a signalized intersection. It is assumed that probe vehicles provide their trajectories after they travel through the intersection. Based on these trajectories, the total FC is calculated using the VT-CPFM. Given the complexities and nonlinear nature of the problem, a simulation-based approach is followed to investigate the sample size and accuracy tradeoff in estimating the FC.

In order to generate the necessary data, a simple network is created in the INTEGRATION software [39]. The reason the INTEGRATION software was used is because it explicitly models vehicle dynamics by considering the various tractive and resistive forces acting on the vehicle. The software generated a dataset containing both vehicle trajectories as well as fuel consumption levels for each vehicle based on VT-CPFM. Figure 9 shows sample trajectories created from the simulation model as well as the road network being simulated (on the right side of the figure).

The total fuel consumed by all vehicles within the analysis period is computed, and is denoted by F_t. If the fuel consumed by each vehicle is known then, F_t can be found by simply summing over all vehicles. The fuel consumption of a particular vehicle traveling over a road segment depends on many factors (e.g., vehicle type, drive cycle, road grade). However, two of the key factors are vehicle type (or make and model) and the trajectory or drive cycle (these two terms are used interchangeably in the paper) it follows. Both of these factors are considered in the analysis. The roadway is assumed to be flat but this could easily be changed in future analyses.

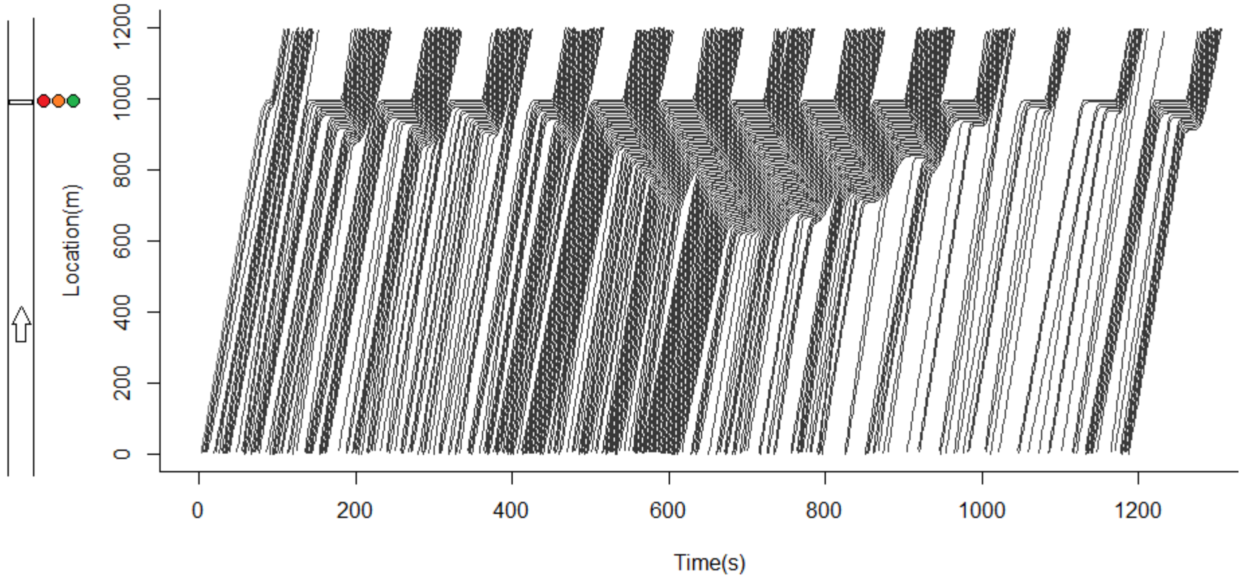


Figure 9: Simulated vehicle trajectories

In order to construct the vehicle trajectory dataset, specific make/models were identified. In this paper, six vehicle types are modeled, namely: Honda Civic, Toyota Camry, Honda Accord, Ford F150, Ford Escape, and Toyota Sienna. The simulated traffic consists of these six vehicle types because they were identified as the top-sold vehicles in their category in the USA. For each one of the vehicles, the parameters needed for the simulation of the vehicles in the INTEGRATION software and the VT-CPFM were obtained from the various automotive websites. With known parameters and a given trajectory, the fuel consumption and CO_2 emissions for a specific vehicle were computed on a second-by-second basis. Figure 10 shows the variation in the travel times, FC and CO_2 emissions for the six vehicle types that completed their trips and exited the simulation model. Each point on Figure 10-b represents the total fuel consumed by a specific vehicle. Since all vehicles traveled through the same single-lane road, vehicles exiting at around the same time experienced similar traffic conditions. It is clear from the figure that there is significant variation in the trip FC across the six vehicle types.

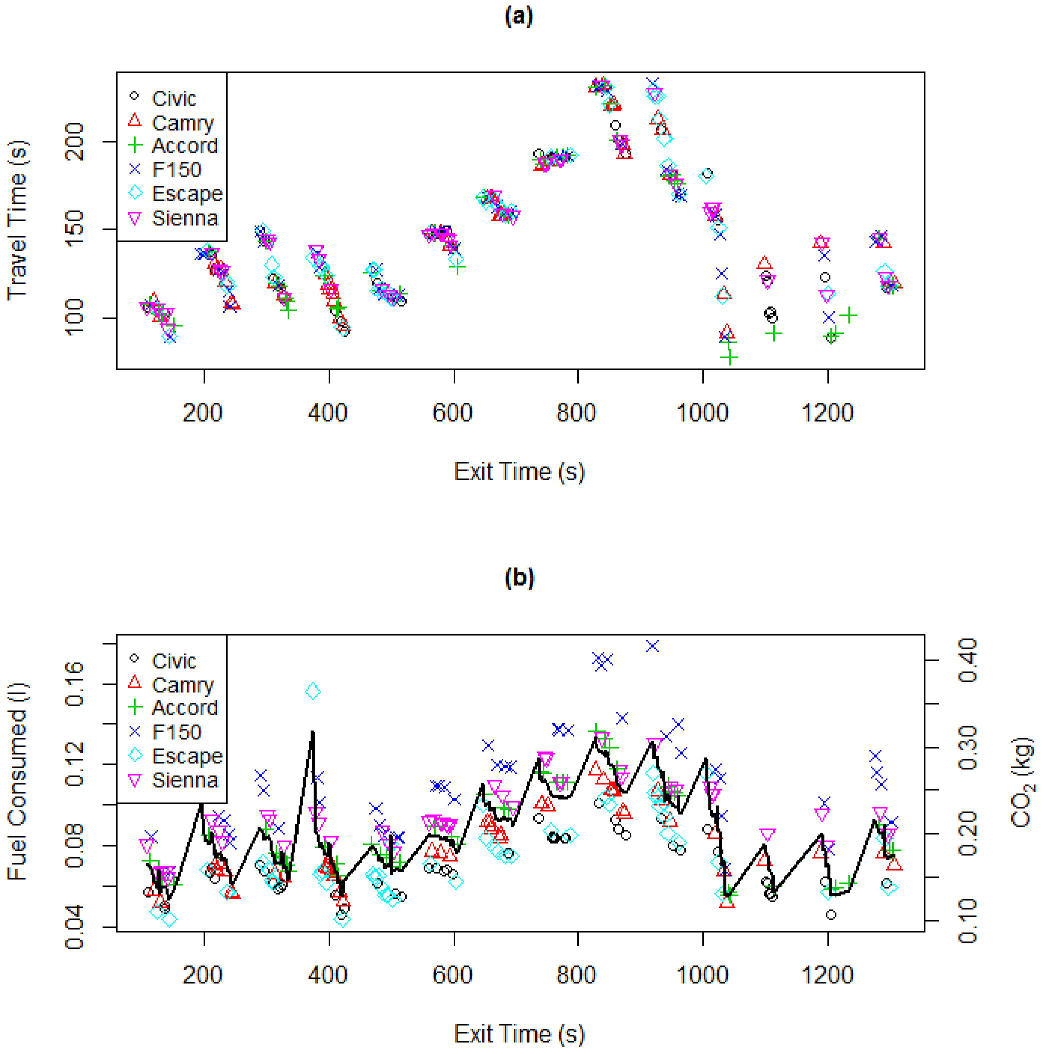


Figure 10: Travel time (a), and fuel consumed & CO_2 emitted (b) for completing the trips by vehicle types

Estimation Methods

The estimation methodology presented here involves making a prediction or estimate for the total fuel consumption for all vehicles within an observation period. It is assumed that the total number of vehicles in the observation period is known as well as the number of vehicles between two consecutive probe vehicles. In essence, it is assumed that there is a vehicle counter either at the entry or exit point of the network shown in Figure 9. It is possible to relax this assumption and estimate the volume from the probe data as well but this will introduce another level of uncertainty and complexity into the analysis. Furthermore, for the given network, one can use data from a traffic counter and probe vehicles and make use of traffic flow models to predict the number of stops, queue dynamics, etc., and use the outputs from these models to estimate the fuel consumption level and CO_2 emissions. These additional complexities are not included here; but instead the focus is on the investigation of the market penetration rate and how it affects the estimated total fuel consumption levels. These complexities are left for a future exercise.

As mentioned in the previous section, if the vehicle trajectory and its type are known then, its FC is assumed to be determined without any error from the VT-CPFM model. Therefore, there are only two sources of error: misrepresentation of the vehicle type and errors in the construction of the vehicle trajectory. When only a sample of vehicles is observed (i.e., probe vehicles), the data from this sample is used to compute an estimate for the total fuel consumed. In this study, two scenarios are considered in terms of the availability of data from the probe vehicles:

- Both trajectory and vehicle type are provided; or
- Only trajectory is provided.

Under each condition, an estimate for the other vehicle parameters needs to be made since not all vehicles are probes. Two estimation methods are presented and evaluated. In the first method, the fuel consumed by probe vehicles is calculated first, and the result is extrapolated to the rest of the traffic by using the observed proportion of probes as the extrapolation factor. In the second method, the trajectories of two consecutive probes are used to approximate the trajectories of non-probe vehicles in between these vehicles. Both methods are presented below.

Method 1: Simple Extrapolation: This method applies a simple factor to predict the total fuel consumption by all vehicles (denoted by $\widehat{F}_t$) and has two variants as follows:

$$\widehat{F}_t = F_p * \frac{N}{n} \tag{1}$$

$$\widehat{F}_t = \widehat{F}_p * \frac{N}{n} \tag{2}$$

Where, *n* is the total number of observed probes, *N* total number of all vehicles in the analysis period, F_p is the actual total FC for all probes when the vehicle make/model is known whereas $\widehat{F}_p$is the estimated FC based on the FC level of an *average* vehicle (vehicle of unknown type). In other words, in the first case (Equation 1), probe vehicles provide both their trajectories and make/model, whereas in the second case (Equation 2) only their trajectories. Defining an "average" vehicle type is a critical aspect of the analysis as will be discussed later on.

Method 2: Interpolation Based on Trajectories: Since traffic dynamics change over time, it will be useful to incorporate the variation in probe vehicle trajectories into the estimation process. In the second method, from the trajectories of two consecutive probes, two fuel consumption values (one for each trajectory) are calculated for an "average" vehicle. These two FC values are then averaged to estimate the FC for the non-probe vehicles in between the two probes. Like Method 1, this method has two variants:

$$\widehat{F}_t = F_p + \sum_{1}^{n-1} N_{(j,j+1)}\widehat{F}_{j,j+1} + N_{(1)}\widehat{F}_1 + N_{(n)}\widehat{F}_n \tag{3}$$

$$\widehat{F}_t = \widehat{F}_p + \sum_{1}^{n-1} N_{(j,j+1)}\widehat{F}_{j,j+1} + N_{(1)}\widehat{F}_1 + N_{(n)}\widehat{F}_n \tag{4}$$

Where the second term in each equation gives the estimated fuel consumed by all non-probe vehicles that have travelled in between probes *j* and *j*+1. Other terms are as follows:

$N_{(j,j+1)} =$ the number of all non-probe vehicles counted in between probes *j* and *j*+1.

$N_{(1)} =$ the number of all non-probe vehicles counted before the first probe vehicle is observed

$N_{(n)} =$ the number of all non-probe vehicles counted after the last probe vehicle is observed

$\hat{F}_{j,j+1}$ or $\hat{F}_j$ = the estimated fuel consumed by a vehicle of unknown make/model predicated on probe vehicle *j*'s and (*j*+1)'s trajectories or only *j*'s trajectory (for the last two terms in Equations 3 and 4). These are calculated as follows:

$$\hat{F}_{j,j+1} = \left(\sum_{k=1}^{m} \alpha_k T_k^j + \sum_{k=1}^{m} \alpha_k T_k^{j+1}\right)/2 \qquad (5)$$

$$\hat{F}_j = \sum_{k=1}^{m} \alpha_k T_k^j \qquad (6)$$

Where,

α_k = proportion of vehicles of type *k* in the population

T_k^j = fuel consumed by vehicle type *k* under the trajectory of probe vehicle *j*.

These estimation methods are labeled as follows for easy referencing:

- M1a: Equation 1 – simple extrapolation while knowing make/model of probe vehicles
- M1b: Equation 2 – simple extrapolation without knowing make/model of probe vehicles
- M2a: Equation 3 – use interpolated trajectories and the known make/model of probe vehicles
- M2b: Equation 4 – use interpolated trajectories without knowing make/model of probe vehicles

All methods except M1a require a definition of an average vehicle (or more precisely the average fuel consumption for a given trajectory) as it is needed when the vehicle type is not known. How an average vehicle is defined will impact the estimation results, and may introduce a bias. In this context, the definition of an average vehicle depends on its fuel consumption; and the average should be defined such that an artificial bias is not introduced. This can be achieved if the breakdown of vehicle types is known. If there are *m* types of vehicles in the population and the proportion of each type *k* is known (α_k's in Equation 6) then, the average vehicle will be the one that consumes the fuel predicted by Equation 6 under an arbitrary trajectory scenario. In other words, the fuel consumption of the average vehicle is the convex combination of the FC of each potential vehicle type.

As explained before, the points on Figure 10-b represent the total fuel consumed by the vehicle to complete its trip through the signalized intersection. In addition to the FC by the individual vehicles, the figure also has a solid line that represents the average fuel consumption. This line is created by applying Equation 6 to each trajectory. In essence, it captures the fuel consumption of an average vehicle for the given trajectories.

In the simulation scenarios described below, it is assumed that the proportion of each type *k* is known (α_k's in Equation 6) so that an average FC can be determined for a vehicle trajectory when its actual type is not known. Without making this assumption (or using some arbitrary average) it will not be possible to implement methods M1b and M2b since under these cases the make/model of probe vehicles is not available. However, it should be understood that defining this average requires prior knowledge about the vehicle mix in the traffic stream, and brings accurate/unbiased average fuel consumption rates into the estimation models.

Simulation Results

In order to show the application of the methods above and to test their performances in estimating fuel consumption, a simple network was constructed in microscopic simulation software INTEGRATION to generate the needed trajectory data. The network was composed of a single lane roadway of 1 km length upstream of the traffic signal and 200m downstream of the traffic signal. The roadway parameters were: a free-flow speed of 50 km/h; a speed-at-capacity of 40 km/h; a base saturation flow rate of 1800 veh/h/lane, and a jam density of 167 veh/km/lane. Variability in driver behavior was captured using a speed coefficient of variation of 0.1 (standard deviation normalized by the mean space-mean speed). The signal timings were: cycle length of 90 seconds with a 45 s effective green time and a 3 s lost time. Initially a traffic demand of 1146 veh/h was loaded onto the network for a duration of 10 minutes followed by a lower demand of 570 veh/h for another 10 minutes. Six types of vehicles are simulated as mentioned before, each with the same proportion (i.e.., 1/6) in the traffic stream. The simulation is run until all vehicles exit the network. Figure 9 shows the resulting trajectories.

In performing the analyses, the following steps were followed:

1. Simulate vehicles in the INTEGRATION software and generate their trajectories
2. Calculate the FC for each vehicle using its trajectory and known type as inputs to the VT-CPFM
3. Calculate the average FC for each trajectory using Equation 6
4. Randomly select probe vehicles from the population of vehicles

5. Implement the four estimation methods
6. Calculate the percentage error or the difference between the actual total FC (F_t) and the estimated fuel consumption ($\hat{F_t}$) as: $(\hat{F_t} - F_t)/F_t * 100$.

The steps outlined above are applied to the trajectory data shown in Figure 9. The random selection of probes (step 4) is repeated 500 times and the probe percentage is varied from 2% to 100%. The results from each one of the four estimation methods are summarized graphically in Figure 11, Figure 12, and Figure 13. In Figure 11, boxplots show the variation in the estimation results for all four models at four market penetration levels: 2%, 5%, 10%, and 15%. In each plot the vertical axis shows the percentage error (defined in step 6 above). As the market penetration of probes is increasing from 2% to 15%, the variation in error decreases significantly for all models. In all models, the median is very close to zero and the boxes are symmetric about the zero line indicating that the models produce unbiased estimates. From these plots, it is also evident that Method 2 produces more accurate results compared to Method 1. However, it is not easy to reach a conclusion between the performances of the two variants of each method.

The results are also presented in terms of the standard deviation of errors in Figure 12 for all four methods and all probe percentage levels. From this chart, it is also clear that Method 2 has lower variation and hence better accuracy relative to Method 1. Perhaps contrary to the expectations, knowing the make and model of probe vehicles does not improve the estimation accuracy as the results for M1a and M2a are not better than their counterparts (i.e., M1b and M2b). The error (or variation) is higher when, for probe vehicles, FC is calculated based on the true vehicle types rather than the average type defined before (compare Equations 1 & 2, and Equations 3 & 4). Although this result may seem counterintuitive at first, it is theoretically valid since the average FC values in a random sample (of trajectories) have lower variation (and by design, using the average FC levels produces unbiased results). This implies that for FC estimation one does not need to know the actual vehicle types of the probes in addition to their trajectory data provided that vehicle mix in the traffic stream is known reliably. This conclusion is valid independent of the number of vehicle types present in the mix due to the theoretical reasons mentioned above.

Another important observation from Figure 11 and Figure 12 is the fact that the total FC can be estimated with a reasonable accuracy (e.g., 5% stdev for the error) with low market penetration rates (5-10%). Obviously, this is based on the simulated data. It will be interesting and useful to repeat this for more realistic vehicle mix with more vehicle types in a future study.

Finally, Figure 13 shows the 95th percentile and 5th percentile bounds for Methods M1b and M2b. If the acceptable error range is required to be ±5% with 90% confidence then, approximately a level of market penetration (LMP) of probes of 9% will be sufficient if M2b is used or 15% if M1b is used. Under M1b, probe vehicles report only their total fuel consumption (not anything else). However, under M2b they provide their trajectories as well. This additional information allows the required market penetration to go down by 6% in this example.

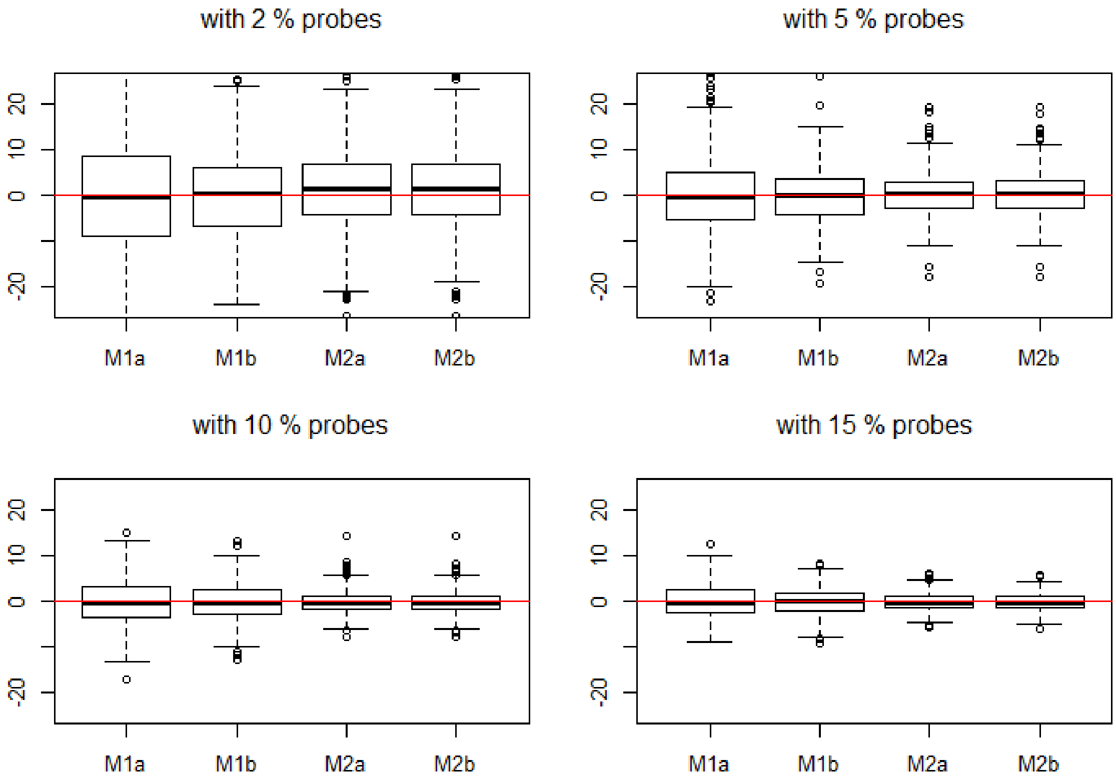


Figure 11: Variation of estimation error as a function of probe market penetration (500 replicas at each level).

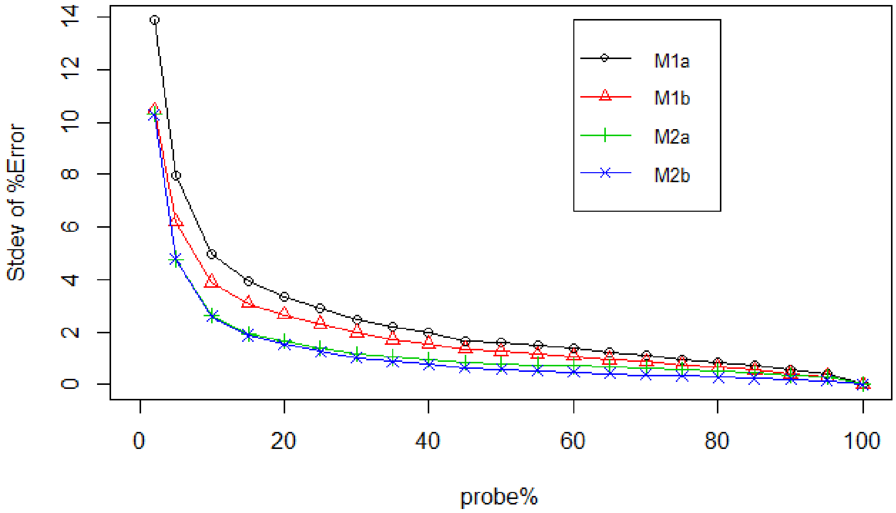


Figure 12: Variation in standard deviation of error as a function of probe vehicle market penetration

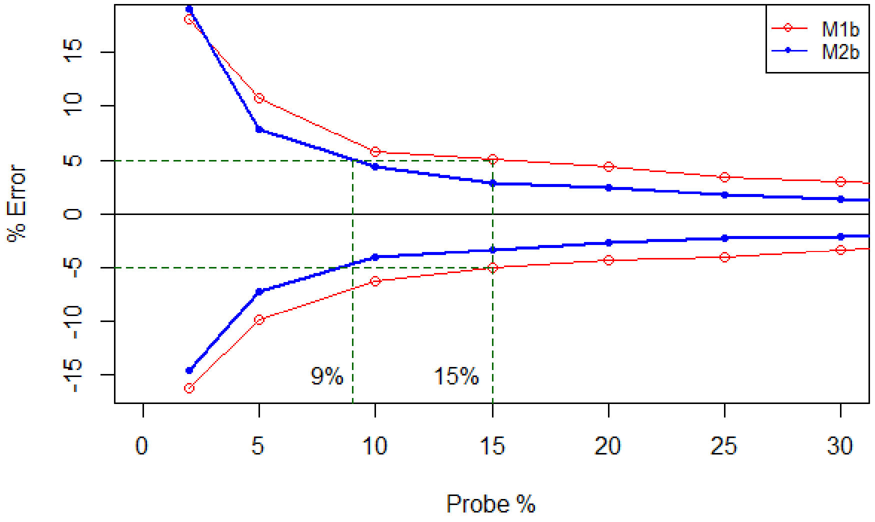


Figure 13: Error bounds for the two methods based on simulation data

The results above demonstrate that the second method provides more accurate results as it makes use of the probe trajectories to infer traffic conditions for the non-probe vehicles. It is shown that probe trajectories (i.e., second method) provide more valuable information for the estimation for high levels of congestion. In addition, the analyses show that the standard deviation of the estimation error drops dramatically as the level of market penetration increases from zero to approximately 20% beyond which the error improves only marginally. Finally, it is shown that when the true vehicle mix is known a priori, the fuel consumption can be estimated effectively without the need to know the make and model of the probe vehicles reporting their trajectories. This result is important for the design and implementation of such systems in the future.

Alternative Algorithms to Find Shortest Path

The problem of finding the Shortest Paths (SP) to simultaneously optimize the total "travel time" and/or "fuel emission" can be formulated and solved in different ways. The simplest way to handle such above mentioned problem is to formulate the "objective function" as a linear combination of total travel time, and total fuel emission allowed, such as:

Minimize OBJ = α * "total travel time" + (1-α) * "total fuel emission"

where α is in [0.00, 1.00] and can be specified by the user.

The original Dijkstra SP algorithm can be utilized to obtain the optimal solution. The above formulation/strategy can also be incorporated (with "minor modifications to the traditional Dijkstra SP algorithm") for handling cases where the real-life traffic data is "occasionally updated", such as the "travel time, and/or fuel emission" for each link of a given network can be varied (rather than having constant values), depending on the specific time of a particular day.

On the other hand, the above mentioned problem can also be formulated as minimizing the total "travel time" (or "fuel emission"), subjected to the total "fuel emission" (or "travel time") allowed. In other words, here we have the so-called SP with an "added constraint". Since this problem belongs to the class "SP with added constraint", traditional SP algorithms, such as Dijkstra algorithm, or Label Correction Algorithm (LCA) etc. can't be directly applied. However, the SP with added constraint can still be formulated and solved based on the following 4 different strategies/algorithms, such as:

Strategy/Algorithm 1: Based on the ideas of finding successive SP (or finding the k-th SP) **without** any added constraint. The 1-st, or the 2-nd, or the 3-rd, or the 4-th etc. SP whichever happens to satisfy the added constraint will be selected as the optimal (or near optimal) solution of the SP with added constraint problem.

Strategy/Algorithm 2: Based on the idea of using the “Lagrange multiplier” in engineering optimization, for which the “added constraint” will be included in the Objective function as a “penalty term”.

Strategy/Algorithm 3: Based on the idea of formulating the “SP with an added constraint” as a Linear Integer Programming (LIP) problem, where some sort of “Branch & Bound” strategies need be applied for computational efficiency.

Strategy/Algorithm 4: Based on the idea of using “Genetic Algorithm (GA)”, with special procedures for “cross-over and/or mutation” operations.

In this research, the shortest path (SP) Dijkstra algorithm for “Static” networks have been re-visited and modified so that the modified Dijkstra algorithm can (i) find the SP to minimize the “travel time” and/or “fuel consumption” for a given network, (ii) finding successive (1st, 2nd, 3rd) or kth SP, (iii) without and with added constraint on “travel time” and/or “fuel consumption”. Several small-scale (academic) networks, as well as real-life/large-scale networks have been used to demonstrate the practical applications of the proposed algorithms and its associated software, under MATLAB computer environments.

For the SP with constraint problems, numerical results obtained from this study have indicated that (a) strategy 1 (based on the idea of successive kth SP), strategy 3 (based on the LIP formulation), and strategy 4 (based on GA) yield the correct optimal solutions. However, strategy 2 (based on Lagrange multiplier formulation) gives near (sub-optimal) solutions (b) in terms of computational (wall-time) efficiency, and for large-scale (real-life) networks, strategy/algorithm 3 (based on LIP formulation) seems to be the fastest. Efforts are underway, however, to further refine the GA formulation, so that clock-time can be significantly reduced in both “serial” and “parallel” computing environments.

CONCLUSIONS

In this project, various methodologies and algorithm are developed and tested with both simulation and field data to support smartphone-based solutions for monitoring and reducing fuel consumption and CO_2 emissions. Machine learning algorithms are developed to predict the travel mode and to detect whether the vehicle is in motion or stopped based on the smartphone sensor data. The results show that the travel mode can be detected accurately, about 94% on average, when considering all travel modes (bike, car, walk, run, bus) within the sample data. The vehicle stop detection is challenging based on noisy accelerometer data, but as shown previously with the hybrid model of SVM-HMM and the changepoint detection algorithms, nearly all the points were detected with acceptable proximity. The common feature that was most important for these algorithms was the variance in the accelerometer data, which helped distinguish whether the vehicle is in motion or stopping.

To predict the impacts of probe vehicle market penetration on estimating fuel consumption, simulation data are created for an intersection. The results show that estimation error drops dramatically as the level of market penetration increases from zero to approximately 20% beyond which the error improves only marginally. Lastly, the shortest path (SP) algorithm for static networks are modified so that the algorithm can find the SP for minimizing both the travel time and fuel consumption for a given network.

Together, the models and algorithms developed in this study can be integrated to support various mobility and environmental applications including Dynamic Low Emissions Zones Concept of the AERIS program. They can also be used to build applications to give feedback to the traveler in terms of FC and CO2 emissions for each completed trip.

ACKNOWLEDGEMENTS

This study was funded by the TranLIVE University Transportation Center (UTC), a consortium of five universities, funded by the US DOT’s Research and Innovative Technology Administration (RITA).

REFERENCES

1. EPA, *DRAFT Inventory of U.S. Greenhouse Gas Emissions and Sinks: 1990-2011*, U.S.E.P. Agency, Editor. 2013, U.S. Environmental Protection Agency: Washington, DC.
2. Susi, M., V. Renaudin, and G. Lachapelle, *Motion Mode Recognition and Step Detection Algorithms for Mobile Phone Users.* Sensors, 2013. **13**(2): p. 1539-62.
3. Widhalm, P., P. Nitsche, and N. Brandie. *Transport mode detection with realistic Smartphone sensor data*. in *2012 21st International Conference on Pattern Recognition (ICPR 2012), 11-15 Nov. 2012*. 2012. Piscataway, NJ, USA: IEEE.
4. Manzoni, V., et al., *Transportation mode identification and real-time CO2 emission estimation using smartphones*. 2010, Technical report, Massachusetts Institute of Technology, Cambridge.
5. Yu, X., et al. *Transportation activity analysis using smartphones*. in *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*. 2012.
6. Stenneth, L., et al. *Transportation mode detection using mobile phones and GIS information*. in *19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM SIGSPATIAL GIS 2011, November 1, 2011 - November 4, 2011*. 2011. Chicago, IL, United states: Association for Computing Machinery.
7. Reddy, S., et al., *Using Mobile Phones to Determine Transportation Modes.* Acm Transactions on Sensor Networks, 2010. **6**(2).
8. Neville, P.G., *Controversy of Variable Importance in Random Forests.* Journal of Unified Statistical Techniques, 2013. **1**(1).
9. Biljecki, F., H. Ledoux, and P. van Oosterom, *Transportation mode-based segmentation and classification of movement trajectories.* International Journal of Geographical Information Science, 2013. **27**(2): p. 385-407.
10. Zheng, Y., et al., *Learning transportation mode from raw gps data for geographic applications on the web*, in *Proceedings of the 17th international conference on World Wide Web*. 2008, ACM: Beijing, China. p. 247-256.
11. Gonzalez, P.A., et al., *Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks.* Iet Intelligent Transport Systems, 2010. **4**(1): p. 37-49.
12. Byon, Y.J., B. Abdulhai, and A. Shalaby, *Real-Time Transportation Mode Detection via Tracking Global Positioning System Mobile Devices.* Journal of Intelligent Transportation Systems, 2009. **13**(4): p. 161-170.
13. Zhang, L., M. Qiang, and G. Yang, *Mobility transportation mode detection based on trajectory segment.* Journal of Computational Information Systems, 2013. **9**(8): p. 3279-3286.
14. Nham, B., K. Siangliulue, and S. Yeung, *Predicting mode of transport from iphone accelerometer data*. 2008, Tech. report, Stanford Univ.
15. Nick, T., et al. *Classifying means of transportation using mobile sensor data*. in *Neural Networks (IJCNN), The 2010 International Joint Conference on*. 2010. IEEE.
16. Bolbol, A., et al., *Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification.* Computers, Environment and Urban Systems, 2012.

17. Zheng, Y., et al., *Understanding transportation modes based on GPS data for web applications.* ACM Transactions on the Web (TWEB), 2010. **4**(1): p. 1.
18. Nitsche, P., et al., *A strategy on how to utilize smartphones for automatically reconstructing trips in travel surveys.* Procedia-Social and Behavioral Sciences, 2012. **48**: p. 1033-1046.
19. Gong, H., et al., *A GPS/GIS method for travel mode detection in New York City.* Computers, Environment and Urban Systems, 2012. **36**(2): p. 131-139.
20. Lester, J., et al., *MobileSense-Sensing modes of transportation in studies of the built environment.* UrbanSense08, 2008: p. 46-50.
21. Weiß, G. *A multiagent perspective of parallel and distributed machine learning*. in *International Conference on Autonomous Agents: Proceedings of the second international conference on Autonomous agents*. 1998.
22. Hall, L.O., et al., *Learning rules from distributed data*, in *Large-Scale Parallel Data Mining*. 2000, Springer. p. 211-220.
23. Caragea, D., A. Silvescu, and V. Honavar, *Decision tree induction from distributed heterogeneous autonomous data sources*, in *Intelligent Systems Design and Applications*. 2003, Springer. p. 341-350.
24. Khedr, A.M., *Learning k-nearest neighbors classifier from distributed data.* Computing and Informatics, 2012. **27**(3): p. 355-376.
25. Herrera, J.C., et al., *Evaluation of traffic data obtained via GPS-enabled mobile phones: The< i> Mobile Century</i> field experiment.* Transportation Research Part C: Emerging Technologies, 2010. **18**(4): p. 568-583.
26. Leduc, G., *Road traffic data: Collection methods and applications.* Working Papers on Energy, Transport and Climate Change, 2008. **1**: p. 55.
27. Jahangiri, A. and H. Rakha. *Developing a Support Vector Machine (SVM) Classifier for Transportation Mode Identification by Using Mobile Phone Sensor Data*. in *Transportation Research Board 93rd Annual Meeting*. 2014.
28. Jahangiri, A. and H. Rakha, *Applying Machine Learning Techniques to Transportation Mode Recognition using Mobile Phone Sensor Data.* ieee transactions on intelligent transportation systems, 2014.
29. R Core Team, *R: A Language and Environment for Statistical Computing*. 2014, R Foundation for Statistical Computing.
30. Liaw, A. and M. Wiener, *Classification and Regression by randomForest.* R news, 2002. **2**(3): p. 18-22.
31. Chang, C.-C. and C.-J. Lin, *LIBSVM: a library for support vector machines.* ACM Transactions on Intelligent Systems and Technology (TIST), 2011. **2**(3): p. 27.
32. Cetin, M., G.F. List, and Y. Zhou, *Factors affecting minimum number of probes required for reliable estimation of travel time.* Transportation Research Record: Journal of the Transportation Research Board, 2005. **1917.**: p. 37-44.
33. Ferman, M.A., D.E. Blumenfeld, and X. Dai, *An analytical evaluation of a real-time traffic information system using probe vehicles.* Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, 2005. **9**: p. 23-34.
34. Ban, X., P. Hao, and Z. Sun, *Real time queue length estimation for signalized intersections using travel times from mobile sensors.* Transportation Research Part C: Emerging Technologies, 2011. **19**(6): p. 1133-1156.

35. Cetin, M., *Estimating Queue Dynamics at Signalized Intersections from Probe Vehicle Data: A Methodology Based on Kinematic Wave Model.* Transportation Research Record: Journal of the Transportation Research Board: Journal of the Transportation Research Board, 2012. **2243**: p. 164-172.

36. Comert, G. and M. Cetin, *Queue length estimation from probe vehicle location and the impacts of sample size.* European Journal of Operational Research, 2009. **197**: p. 196-202.

37. Comert, G. and M. Cetin, *Analytical Evaluation of the Error in Queue Length Estimation at Traffic Signals From Probe Vehicle Data.* IEEE Transactions on Intelligent Transportation Systems, 2011. **12**(Copyright 2011, The Institution of Engineering and Technology): p. 563-73.

38. Rakha, H.A., et al., *Virginia Tech Comprehensive Power-Based Fuel Consumption Model: Model development and testing.* Transportation Research Part D-Transport and Environment, 2011. **16**(7): p. 492-503.

39. Van Aerde, M. and H. Rakha, *INTEGRATION © Release 2.40 for Windows: User's Guide – Volume I: Fundamental Model Features*. 2013, M. Van Aerde & Assoc., Ltd.: Blacksburg.