Transportation Mode Classification

by

Peter Braun

A thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada

March 21, 2018

Thesis advisor | Author

Prof. Dr. Carson K. Leung | **Peter Braun**

Transportation Mode Classification

Abstract

The increasing amount of digital data in urban research has drawn attention in *urban data mining*. In urban research (*e.g.*, travel studies in urban areas), researchers who conduct paper- and telephone-based travel surveys often collect biased and inaccurate data about their participants' movements. Although the use of GPS trackers in travel studies improve the accuracy of exact participant trip tracking, the challenge of labelling trip purpose and transportation mode still persists. The automation of such a task would bring benefits to travel studies and other applications that rely on contextual knowledge (*e.g.*, current travel mode of a person). In my M.Sc. thesis work, I focus on transportation mode classification. In particular, I develop a system that improves classification accuracy of ground transportation modes (*e.g.*, bus, car, bike, or walk), when compared with existing systems, by uniquely using GPS-, accelerometer-data and bus stop locations in one system. To elaborate, I design new classification features based on *Dwell Time History* and a *Window History Queue*, which uses previously encountered data to increase the classification accuracy of current data. The resulting system remains as a semi real-time classification system, and it achieves a high classification accuracy of 98.5%. Additionally, I explore the performance of classifiers by training with different combinations of GPS-, accelerometer- and GIS data.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank Prof. Dr. Carson K. Leung for supporting me throughout my journey as a graduate student. He has been a great mentor, advisor, and teacher to me and always helped me with any problems I encountered. Through him, I was first introduced to data mining and had the chance to learn a lot about the field, in particular frequent pattern mining. He has also given me the chance to explore different data mining topics, including the field of urban data mining and transportation mode classification.

I would like to thank my thesis advisory committee members, Dr. Jonathan D. Regehr, P.Eng. (Civil Engineering) and Dr. Yang Wang (Computer Science), for your support and time to read and examine my thesis and defence. I also thank Dr. M. Rasit Eskicioglu for chairing my thesis defence.

I would also like to thank my lab mates and friends from the Database and Data Mining Lab, and other labs, Hao Zhang, Richard K. MacKinnon, Fan Jiang, Zhao Han, Oluwafemi A. Sarumi, Mrigank Rochan, and many more that I worked with and have helped me. I am honoured to have made such great new friends over the years and I appreciate their support and encouragement. I also thank my fiancée, Youlan Wang, who has been by my side throughout all those years.

I thank Frank Franczyk, president of Persen Technologies Inc. (PERSENTECH), for his support and encouragement throughout my studies and for providing me with a modified OttoFleet Mobile™ tracking application.

Last but not least, I want to thank my parents for their support, love and encouragement. They have been there for me throughout all my endeavors.

Peter Braun

B.C.Sc. (Honours), The University of Manitoba, Canada, 2012

The University of Manitoba
March 21, 2018

Chapter 1

Introduction

Data mining aims to extract "implicit, previously unknown, and potentially useful information" from a large amount of data (Fayyad et al., 1996). With the increasing amount of digital data in urban research, the field of *urban data mining* (Behnisch & Ultsch, 2008; Sokmenoglu et al., 2010; Andrienko et al., 2016, 2017) has also developed and been given more attention. *Urban data mining* aims to discover implicit, previously unknown and potentially useful knowledge from data related to urban problems for solving urban issues.

In urban research, specifically travel studies in urban areas, researchers often have been using paper- and telephone-based travel surveys. Those surveys can often be biased and contain inaccurate data about their participants' movements (Murakami et al., 2004; McGowen & McNally, 2007). Participants tend to under-report short trips and irregular trips. Additionally, car trips are often reported to be shorter than they are and public transit trips are reported to be longer than they are (Stopher, 1992; Ettema et al., 1996).

Commute diary studies are another approach of collecting data about people's daily commutes but they also have been shown to be prone to errors. When people are asked to use a diary to keep track of their commutes, they often forget to record their commutes throughout the day, and when trips are recorded at the end of the day then diary studies can inherit the same problems as paper- and telephone-based travel surveys (Stopher & Wilmot, 2000; Maat et al., 2004). For that reason commute, diary studies can also be a mental burden to study participants and cannot be used long term (Schönfelder et al., 2002; Schlich & Axhausen, 2003). It has also been shown that, because of the mental burden, people's willingness to record trips accurately throughout the day declines with each day of participation, and thus the commute diaries become less accurate with each passing day (Arentze et al., 2001).

To avoid data collection problems that paper-based travel survey and commute diaries have, there is now a shift towards the use of GPS trackers to collect more objective commute data from participants. Studies with such GPS trackers in the early 2000's have confirmed the problems related to paper-based travel survey and commute diary studies, and they have also shown that GPS-based travel surveys are more accurate (Wolf et al., 2001; Gleave & GeoStats, 2003; Forrest & Pearson, 2005; Ohmori et al., 2005).

Although the use of GPS trackers in travel studies improve the accuracy of exact participant trip tracking, the challenge of labelling trip purpose and transportation mode still persists. Nowadays, with the use of electronic trackers, researchers are often dealing with a large amount of movement trajectories that are collected by participants of a study who use GPS trackers and sometimes various other sensors like Bluetooth, WiFi, accelerometers, barometers, *etc.* Manually trying to segment trajectories into origin-destination segments and further segmentation by transportation mode is a very labour intensive task and most likely infeasible if performed on a large dataset (Biljecki et al., 2013). The automation of such a task would bring obvious benefits to travel studies and also other applications that rely on contextual knowledge such as current travel mode of a person. A contextual use for transportation would be for example when a person drives to a new shopping mall with the car. While the person is driving, a device like a smartphone could recognize that the person is driving in a car and give a notification about the current estimated time of arrival, assuming the phone knows the destination based on previous user interaction or saved frequently visited locations. Another application for transportation mode classification is also the automatic trip transportation mode labeling for trip histories similar to Google Maps Timeline™ (Google Inc., 2018a) (requires a Google account to view), which keeps track of a user's location history and automatically classifies trips with the major transportation mode. In the Google Maps Timeline™ example, we can observe that those classifications are not very accurate and need a lot of corrections by the user. It also does not track when transportation modes were changed. A more accurate algorithm or system would be much more helpful and interesting.

As previously mentioned, travel surveys that use GPS tracking either use standalone GPS devices or smartphones to record a person's movement. Nowadays, smartphones are very ubiquitous, offering an opportunity to use them as tracking devices for travel surveys. This obviously reduces study costs since no extra hardware needs to be bought or rented for participants. Only a smartphone application needs to be developed or bought, which can be much cheaper, especially for large scale surveys. The disadvantage of using smartphones in travel studies is that high frequency GPS logging (*e.g.*, 1-5 Hz) consumes a lot of battery power, which can be a burden on the participants or even might affect them negatively if they have to recharge their phones multiple times a day. Another problem with smartphones is that the sensor data logged through custom applications is not consistent across phones because smartphone manufacturers use different internal hardware, such as GPS and accelerometer chips and different operating system, such as Android™ and iOS™, which additionally have different APIs and internal implementations of data retrieval from sensors, making it more difficult to write apps that log consistently across different device platforms.

A standalone tracking and logging device for participants of travel surveys would be able to log sensor data more reliably and consistently because developers and engineers have full control over the device and the hardware, and software platforms are the same on every device. Such loggers can be used to simply log data to local device storage and then be collected for data retrieval. Some devices could also connect via Bluetooth™ to a smartphone application on a participant's phone and collect data this way on regular intervals, where data could be further processed and the users could be prompted with surveys. In such a case, transportation mode classification could happen on a smartphone, reducing computational burden on the logger device, allowing for a cheaper architecture that requires weaker processing units and hence potentially decrease power consumption and thus increase battery life. The system proposed in this thesis could work well for all scenarios described above — smartphone logging (with online and offline classification) and standalone logging devices.

In general, there are two different types of approaches of transportation mode learning and two ways for classification, *offline-* and *online-learning*, and *offline-* and *online-classification* (Su et al., 2016). In *offline learning*, a classification model is trained after collecting an entire dataset of movement data. Thus, the classification model training is usually performed on a server. The classifier can then be deployed on a workstation or a server for *offline classification* or on mobile devices for *online classification* of new data. In *online learning*, a classification model is trained after collecting some (but not necessarily the entire set of) movement data. Thus, the training is usually performed on a portable device, such as a smartphone. Furthermore, in *online learning*, the initial classifier can be trained offline first (*i.e.*, on a server or workstation) and then the mobile device continues training the model online on the mobile device, which has the advantage that each mobile device will have a classifier customized for individuals.

The two approaches, *online-* and *offline-classification*, use different techniques to perform the classification. *Offline classification* uses mostly some form of expert system, where trip traces get split into smaller but still fairly large sub-segments and then are classified separately. Afterwards, the classified sub-segments get stitched back together. The work of Biljecki *et al.* (2013) is a good example of that. The segmentation of the traces usually happens by defining a lot of general rules, like dwelling places, and average speed changes, but to make segmentation more and more accurate it is also necessary to define lots of segmentation rules based on discovered edge cases. For classification, fuzzy-logic algorithms are used that will deduce transportation mode for sub-segments based on computed parameters. This kind of approach has the advantage of becoming quite accurate for a certain dataset but might not be easily transferable to a different dataset from a different environment. It also requires a lot of programming of predefined rules by the developer of the system for the segmentation and classification to succeed. *Online classification* usually uses a different approach in terms of segmentation and classification. Here, not much thought is and cannot be put into segmentation of the data. Trip data is simply divided into small windows of data. Based on the real-time requirements, the window length could be 1 second or even 5 minutes long. Then each window's transportation mode is classified by calculating feature parameters, and a machine learning model is trained based on those parameters. Then the trained model could be deployed on either a portable device like a smartphone or a remote server, where data can be classified in real-time and provided to the user. The work of Stenneth *et al.* (2011) is a good example of online classification. This kind of approach has the advantage of reducing the amount of programming work for the developer to develop special rules that will help the system deduce the transportation mode. One can simply calculate generic parameters based on the data collected, and train a model that will figure out the subtle differences and patterns in the data. Another advantage is that there is not much logic and work necessary to segment trip traces. Once all windows are classified, labelled and stitched together the complete trip sub-segments for each transportation mode will reveal themselves to a user. The disadvantage of online classification is that it is more difficult to increase accuracy without sacrificing model generalization for new datasets and environments. A model can easily be overfitted. Usually, it helps to get bigger windows to make a more accurate classification but that would result in missing the exact time and place, where transportation modes changed. It also would result in slower real-time updates. Obviously, online classifiers could also be used in offline classification by simply using the classifier on preexisting data on a server or data that is sent to a server for classification.

In my approach of transportation mode classification, I use the basis of an *online classification* approach but also use multiple windows of previously encountered data in order to increase classification accuracy. *Online classification* focuses on one window at a time during processing and classification. To the best of my knowledge, existing academic works in this area have not taken advantage of previously encountered data windows in order to increase the classification accuracy of the currently processed window of data. In my work, I design new classification features based on *Dwell Time History* (Sections 4.8 to 4.9) and a *Window History Queue* (Section 4.10), where *Dwell Time History* focuses on a trip's dwell time at bus stops and the *Window History Queue* focuses on summarizing data from previously encountered data windows.

In my thesis work, I extract implicit, previously unknown, and potentially useful information from a large amount of *movement data* for solving urban problems. More specifically, based on people's movements, tracked by GPS, and accelerometers, augmented with GIS data (bus stop locations), I extract the transportation modes people use in an urban area. To achieve the goal, I designed and implemented a system for transportation mode classification of trip sub-segments using:

1. GPS trajectories,

2. accelerometer data, and

3. limited GIS data (*i.e.*, bus stop locations)

The classification approach in this system takes advantage of the techniques mainly used in *online classification*, which are easier to program and do not require an extra initial and complicated segmentation phase. In addition, I use multiple windows of previously encountered data in order to increase the classification accuracy of the currently processed window, whereby the system still remains a semi real-time classification system.

1.1 Thesis Statement

The data collected in travel studies based on paper and online surveys can be biased, often contain wrong information, and are sparse due to participants' subjectivity and poor memory recollection (McGowen & McNally, 2007). Electronic mobile trackers open up the opportunity to improve the quality of travel study data, specifically for trip segmentation and classification of transportation modes for trip segments. *For my MSc research, the goal is to improve accuracy of transportation mode classification, when compared with existing systems. My key contribution is to develop a transportation mode classification system that, to the best of my knowledge, uniquely uses limited GIS information (i.e., bus stop locations), accelerometer data, and GPS data in one system during the classification process to improve general transportation mode classification accuracy.*

Questions to be answered in this thesis:

1. Does adding bus stop location-based features to a feature set based on accelerometer and GPS data improve real-time window-based transportation mode classification?

2. Is it possible to calculate features based on previously seen data for a single trip and use it to improve real-time window-based transportation mode classification?

1.2 Thesis Organization

My thesis is structured in the following way. I provide more background information in Chapter 2. I review related works in the field of transportation mode classification in Chapter 3. I show advantages and disadvantages of existing works and how it relates to my thesis research. At the end of the chapter I introduce my ideas and the problems I solve in my thesis research.

In Chapter 4, I describe how I collected my mobility dataset using two mobile applications, that logged GPS and accelerometer data, which is used in my transportation mode classification system. I give an overview of the dataset, the data flow and the system architecture, including feature generation and classifier training. I also show my newly developed features, which are used to improve transportation mode classification for real-time transportation mode classification, and could also be used for offline classification. Bus stop locations are the main source for my new features, which are *dwell-time* features at bus stop locations and trip-based *dwell-time history* features. I also show a *window history queue* that keeps track of previous data to help improve classification accuracy for a given window of data.

In Chapter 5, I describe my testing methodology for classifying transportation modes on my mobility dataset, and present the classification test results, analysis and interpretation, and the impact on classification accuracy of my new features. Based on my results I also draw some conclusions about the power consumption of classifiers based on different dataset combinations.

Finally, in Chapter 6, I give my research conclusions and discuss future work that I would like to do to extend my study on transportation mode classification.

Chapter 2

Background

In this chapter, I briefly describe and explain some important terms, techniques and tools that I used in my thesis work.

2.1 Haversine Distance

The *Haversine Distance* (Cajori, 2007) is the Haversine version of the great-circle distance between two points on a sphere. The distance is calculated with the following formula:

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos\left(\varphi_1\right)\cos\left(\varphi_2\right)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

where:

- d, is the distance be two points A and B on a sphere,
- r, is the radius,
- φ_1, is the latitude of point A,

- φ_2, is the latitude of point B,
- λ_1, is the longitude of point A, and
- λ_2, is the longitude of point B

I use the *Haversine Distance* in my thesis to calculate the distance between two geographical coordinates, which is used as a feature in my classifier training.

2.2 Random Forest

Random Forest (Breiman, 2001) is a machine learning model based on *Decision Trees*, used for regression and classification of data. Multiple decision trees are trained on subsets of feature parameters from the training dataset. The resulting ensemble of decision trees is then used make a combined decision in the classification or regression phase of the provided data. *Random Forest* is the model I am using in my thesis.

2.3 Stratified 10-fold Cross-validation

To test the *Random Forest* (Breiman, 2001) classifier in my thesis, I use a *stratified 10-fold cross-validation* (Geisser, 2017) testing strategy with a 50-50 partition split. K-fold cross-validation is a common strategy used in machine learning. Such a test also works well when not a big amount of data is available. A 10-fold cross-validation test splits the dataset into 10 equal partitions. Each data partition will be used to train a classifier and test it. In my thesis I chose a 50-50 split for each partition, *i.e.*, the first half of the partition is used for training the classifier and the second half is used for testing it. Stratified cross-validation

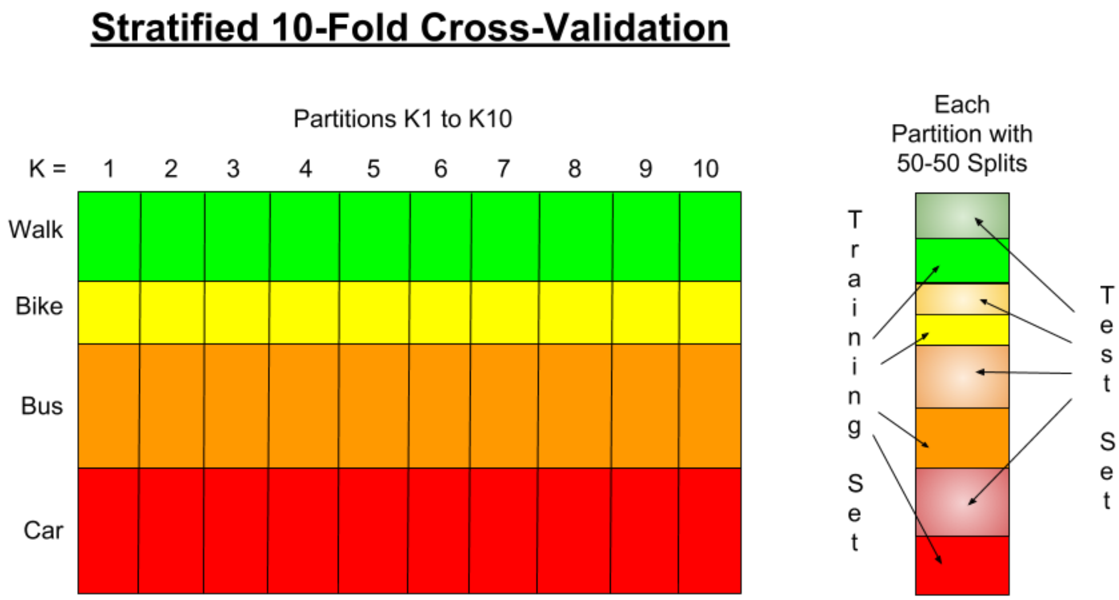


Figure 2.1: A Stratified 10-Fold Cross-Validation example, where each partition splits each class data portion into half for training and testing.

test takes an equal ratio of data from each class of data, ensuring that each partition has similar amounts of data and uses data of all classes of the dataset. An example of a stratified 10-fold cross-validation with a 50-50 partition split is depicted in Figure 2.1.

2.4 Accuracy, Precision, Recall, and F1-score

Accuracy, *Precision*, *Recall*, and *F1-score* (Chou, 1969; Blair, 1979; Powers, 2007) are calculated ratios used in machine learning and pattern recognition to measure the performance of created model.

Precision is calculated as follows, and represents the ratio of numbers of correctly predicted positive transportation mode labels to the total number of positive predictions performed:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

Recall is calculated as follows, and represents the ratio of numbers of correctly predicted positive labels to the total number of labels that are truly positive and the ones that should have been labeled as positive:

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

Accuracy is calculated as follows, and represents the ratio of numbers of correctly predicted labels to the total number of predictions performed:

$$Accuracy = \frac{TruePositives + TrueNegatives}{TruePositives + TrueNegatives + FalsePositives + FalseNegatives}$$

F1-score uses *Precision* and *Recall* and can be interpreted as a weighted average for *Precision* and *Recall*.

F1-score is calculated as follows:

$$F1\text{-}score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

2.5 Summary

In this chapter, I provided some background information that is relevant to the remainder of this thesis. For example, the *Haversine distance* measures the distance between two geographical coordinates on the spherical Earth. The *Random Forest* model uses multiple decision trees for classifying data based on their *Haversine distance* and other features discussed in my thesis. To test the Random Forest model, a *stratified 10-fold cross-validation* strategy can be used. It splits the data into 10 partitions during cross-validation. Precision, recall, accuracy, and F1-score measures the performance of the classification when using the Random Forest model with the *stratified 10-fold cross-validation* strategy.

Chapter 3

Related Works

In this chapter, I describe related works relevant to my study and the understanding of transportation mode classification. Towards the end of this chapter, I provide a summary table of the related works (Table 3.1) capturing techniques and data used and their classification accuracy achievements.

Most research works related to *transportation mode classification* require at a minimum GPS or Accelerometer data for creating a classification model. Much research exists, where researchers used some combination of data from different sensors, like GPS and accelerometers, and other modern smartphone sensors (*i.e.*, barometer, magnetometer, *etc.*). Some researchers also augmented GIS data to their sensor datasets, but none of them uniquely combined GPS, accelerometer, and GIS data, in one study, the way I did.

In this chapter, I grouped the related works by the type of dataset research used in their studies, namely:

1. GPS

2. Accelerometer
3. GPS and GIS
4. GPS and Accelerometer
5. Multiple Sensors

Some of the following works may not necessarily be directly related to my work but may help to understand some of the progression and inspiration of newer work and future work, including mine.

3.1 GPS

Zheng *et al.* (2008; 2010) used supervised *Decision-Trees* machine learning and graph-based post-processing after machine learning classification to classify transportation modes from GPS data only. Their system achieved an accuracy of 76%. Their system was capable of classifying four modes of transportation, “walk”, “bike”, “bus”, and “car”. Zheng *et al.*’s approach focused on classifying trips after data for an entire trip was collected. They used common GPS features used in other works for their machine learning classifier and used a “change point-based” graph method for segmenting their trips prior to classification, which is based on location point density calculations. Post-processing based on probabilities of transportation mode switching helped them to further improve their system’s classification accuracy significantly. The classifier developed by Zheng *et al.* is appropriate for scenarios, where entire trip data was already collected. Hence, it would not work for real-time classification.

3.2 Accelerometer

Hemminki *et al.* (2013) focused, in one of their works, on classifying transportation modes ("stationary", "walk", "bus", "train", "metro", "tram") only with the use of accelerometer data. Their developed system achieved an accuracy of 80.1%. The data was collected with various smartphones logging data at 60-100Hz. To classify transportation modes, three different classifiers were trained with a combination of AdaBoost and Hidden Markov Model for three different classes of modes. A major feature Hemminki *et al.* developed is "gravity estimation". It estimates the gravity acceleration vector from the data frame's accelerometer data and then can be used to align with the gravity direction in order to develop better features along the true vertical axis. Those gravity estimation-based features accounted for about 20% of their total classification accuracy 80.1%.

Shafique and Hato (2015) also used accelerometer data only. Specifically, they used only accelerometer data in their study and applied multiple machine learning algorithms to perform transportation mode classification. Their results showed that Random Forest was the most accurate algorithm to classify transportation mode. To extract accelerometer data features, they developed an equation to calculate the "k-point average" and an equation for "average resultant acceleration". Trips are segmented into windows during data collection, but only averages, minimum and maximum acceleration values where recorded. Those windows were then later grouped for average calculations, which are used for the training and classification process. The authors tested various window group sizes ranging from 125 seconds to 625 seconds. They found that the higher the window group sizes the higher the accuracy was. Their study had two limitations:

1. Even the minimum window group size of over 2 minutes used for the classification

process is relatively long and might not be suitable for real-time classification.

2. The devices were mounted to a fixed position during their data collection phase, which restricts real world applications.

With a fixed accelerometer the acceleration data is much more noise free and the point of gravity is always well known. This is most likely why their method achieved such a high accuracy of 99.8% with only average calculations.

In my study, I do not restrict logging device positioning and orientation, which would allow for a more practical application for real-time classification, where application users do not worry about device orientation. The average calculations for average resultant acceleration are a good idea for a basic set of features, but more features are probably needed if the data is more noisy as in my case.

Hemminki *et al.*'s (2013), and Shafique's and Hato's (2015) works showed that it is possible to build a good classifier only based on accelerometer data and that it has a strong contribution in the classification process. This is why I also included accelerometer data in my system.

3.3 GPS and GIS

Chung and Shalaby (2005) developed a system for GPS-based travel surveys. Their system classified four transportation modes, "walk", "bike", "bus", and "car". The authors used a dataset collected in Toronto, where participants wore a proprietary GPS logging device, collecting 60 trips at 1Hz. The system achieved the transportation mode classification by using a rule-based algorithm and map-matching (Greenfeld, 2002) algorithm to detect the exact roads people moved on. Based on GPS data and road network data, simple rules, such as, speed ranges, GPS location clustering, public transit route matching, *etc.*, could be programmed to enable transportation mode detection. Chung *et al.*'s system achieved a 92% classification accuracy in their tests. Their approach works well if road network data and bus stop locations are available but the developed system is dependent on those GIS data. In case of lack of such data their classification model might not yield a high accuracy. Additionally, their system only works for cases where classification can be performed after a trip has finished, which means that it could not be used for scenarios where real-time classification is preferred.

When dealing with raw GPS data from travel survey studies, one challenge is to automate segmentation of GPS traces for origin and destination, and for different transportation modes. To deal with this challenge, Biljecki *et al.* (2013) developed a method to segment GPS traces into "journeys" by looking for idling clusters with long dwelling times, and they further divide journeys into separate transportation mode segments by looking at short dwellings at idling clusters, quick changes in direction and speed, and closeness to bus stop locations. After segmentation, the segments are classified with a hierarchical fuzzy logic expert system that uses GPS and GIS information and has a hierarchical classification technique, which reduces the number of false positives by falling back to a more generic classification, when there is too much ambiguity between two class instances. The system Biljecki *et al.* (2013) developed was for users to analyze trips on a map and classify them based on the expert system's suggestion. The system also looks at the entire trip data to perform segmentation and segmentation mode classification. This method can sometimes be more accurate and is suitable for situations where classification is not needed in real-time.

Nonetheless, I adapted some of the ideas into my window-based real-time classification system. Biljecki *et al.* looked at dwell times and distances to public transit stations and bus stops, which helped them to segment trips and classify those longer trip segments. In the next paragraph, I describe another related work that also used some public transit information calculate features for the classification model. I adapted those ideas to fit a window-based classification approach.

Stenneth *et al.* (2011) built a real-time transportation mode classification system that uses GPS and GIS information. The GIS information included rail and bus routes, bus stop locations, and real-time bus locations. To perform the classification, they used the *Random Forest* (Breiman, 2001) supervised machine learning algorithm to identify a person's current transportation mode. They tried various other common machine learning algorithms to compare their performance and concluded that Random Forest outperforms the other ones. In Stenneth *et al.*'s work, trip segmentation was performed by splitting data into relatively small windows of 30 seconds, where each window contains two data points. The data was submitted from a mobile app to a server, where the classification was performed. One of the features Stenneth *et al.* calculated, is the current distance to a bus stop within a window of data, which contributed a little bit to the classification accuracy but not significantly.

In my research, I seek to improve upon the idea of calculating features based on bus stop locations proximity. I think the reason why bus stop distance alone does not greatly impact classification accuracy is because if a person drives in a car along a bus route, the algorithm would calculate similar feature values for a car and a bus, or maybe even a bike, which in most cases probably caused the ambiguity. We need to produce a feature that better captures the driving behaviour of a bus. Cars usually do not frequently stop at bus stops, and if they do then usually not for a long time. Buses, on the other hand, frequently stop at bus stops for a longer time. Based on that notion, I use dwell time at bus stops for my transportation mode classifier, which helps with removing more ambiguity between modes of transportation.

Chung and Shalaby's (2005) and Biljecki *et al.*'s (2013) studies, both developed a classification for non-real-time use where they have the advantage to look at a relatively large array of data of each trip in order to better segment a trip into smaller segments, which could then be more accurately classified. In a windowed segmentation approach, such an advantage is lost. In a windowed real-time classification approach people also only looked at one window's data at a time to do the classification.

In my research, I also take advantage of previously seen data just like in a non-real-time approach. To achieve that, I developed two features that would allow for that: the "Window History Queue" that allows me to add summary data from previous windows to the current window, and the "Dwell Time History" feature that keeps track of stops and dwell times from the start of a trip to the current data window. This kind of technique adds some of the advantage of a non-real-time approach to the real-time classification approach.

Liang *et al.* (2017) used GPS data and bus stop locations in their work. They utilized Random Forest for their classifier to classify transportation modes of trip segments, which they segmented after data for a full trip was retrieved. Liang *et al.* developed seven new features based on velocity, acceleration and behaviour of people, which includes a "Bus Stop

Rate". The bus stop rate includes distance measurements near bus stops at a close to zero velocity, which helped the classification model to distinguish better between cars and buses. Since they are classifying long segments, which are potentially single mode segments, the bus stop rate feature can be calculated for the entire segment. Liang *et al.* (2017) achieved a classification accuracy of 86.5%, which was about 1.6% better than a similar study they compared to. It is also a non-real-time application, and the system segments a whole trip into potential single mode segments

In my case, where I only have small 4 second windows, the bus rate feature would work well for a particular window of data but for the following windows, when the bus is not near a bus stop, the feature would not have an impact anymore. My new features based on bus stop locations also use similar measurements to their bus stop rate feature, but in order to take advantage of the knowledge in all related windows I developed features that represent the knowledge throughout a trip.

3.4 GPS and Accelerometer

Like other studies, Ellis *et al.* (2014) used the Random Forest classification to successfully perform transportation mode classification with a relatively high accuracy. Their dataset contained both GPS data and accelerometer data. To perform their classification, Ellis *et al.* segmented their recorded trips in to windows of 1 minute length. For each window, they calculated 49 features based on the GPS data and the acceleration data and then trained their classification model based on those windows. Here, the algorithm does not make the assumption that there should always be a walking mode between stages of changes transportation modes. When using a window of 1 minute length, that assumption will not be very practical in some cases where it takes less than 1 minute to switch modes of transportation. Ellis *et al.* also used a post-processing phase after the model classification phase, where they used a moving average voting algorithm, using 5 windows to catch and correct misclassified cases where modes rapidly changed. This post-processing phase made a significant accuracy improvement in the authors' experiments.

The basis of my transportion mode classification system is very similar to the one in Ellis *et al.* (2014), except that I also use bus stop location features and other new features, as described above, to reduce ambiguity between private and public transportation modes. Their work lists features commonly used across transportation mode classification research. In Chapter 4, I mention which features I use in my classification system.

3.5 Multiple Sensors

Much research related to transportation mode classification tends to focus on using only a small number of sensors to aide in mode classification. Popular combinations of sensors used in research are described above in this chapter. There is some research that involves using a combination of many senors, such as GPS, accelerometer, gyroscope, magnetometer, barometer, and even air humidity sensor, that are available in today's modern smartphones. The more sensor modules are used in a smartphone the more the power consumption increases, especially if a high sampling rate is used to collect data from them. For real-world smartphone applications that are supposed to run many hours per day as a background task to collect data, it is not practical for users because it would deplete the phone's battery too quickly causing great inconvenience. If power consumption is not a great concern then research like Su *et al.* (2015) produced shows that with the utilization of more smartphone sensors it is possible to get a very high transportation mode classification accuracy, of 96.4%, without using GIS data. In their work, the authors were able to classify transportation modes such as “walk”, “jogging”, “bike”, “car”, “bus”, and “metro”. They used a two-step classification method to achieve classification on a dataset that was logged with smartphones at a 5Hz sampling rate and segmented into 13 seconds windows. In the first step a Bayes Net classifier identifies “non-wheeled” vs. “wheeled” transportation mode for a given window and then in the second step another Bayes Net classifier identifies more granular classes of transportation modes (“walk”, “jogging”, “bike”, “car”, “bus”, and “metro”).

My work focuses only on a small number of sensors, namely accelerometer and GPS, because of power consumption issues and practicality. One advantage of using a small amount of sensors is that less data is logged and analyzed. This benefits the training phase of a classification system and the real-time classification. For the training phase less data needs to be stored and less features need to be calculated. This makes the process of retraining a model faster. Similarly, for the classification phase, less features need to be calculated on the tracking device (*e.g.*, on a smartphone, or a dedicated tracker) and less memory is needed for classifying transportation mode in real-time, which in turn reduces power consumption on the tracking device, as well. Let us consider a scenario where a country’s transportation department wants to perform a nation-wide travel survey with selected participants. Such a study could have more than 60,000 participants. Not every participant would have smartphones that could be used for travel logging, or participants simply do not want to have the inconvenience of additional power consumption on their personal device. One solution would be to buy smartphones for such studies and lend them to participants, but even cheaper smartphones might be too expensive for a big travel survey.

With memory footprint and power consumption in mind, if only a few sensors are necessary to perform reliable transportation mode detection, then it could be feasible to manufacture a dedicated logging device for such travel studies. Such a device could be cheap and small enough that it would be more affordable given a transportation department's budget and it would be better for participants because they do not need to worry about using their personal devices for the study.

3.6 Summary

In this chapter, I have discussed works that are indirectly and directly related to my work. The research shows a common framework for building a system for transportation mode classification. In general, researchers collected a combination of GPS-, accelerometer- and GIS-data. The trip data is then segmented into smaller legs or small windows, and then each segment is classified. Researchers have used two different approaches for the classification. The first one is a rule based classification system (Chung & Shalaby, 2005), where manually developed rules and thresholds are used to determine the transportation modes of a trip. The second approach uses machine learning to train a classification model to recognize transportation modes based on developed features (Zheng et al., 2008, 2010; Stenneth et al., 2011; Biljecki et al., 2013; Hemminki et al., 2013; Ellis et al., 2014; Shafique & Hato, 2015; Su et al., 2015; Liang et al., 2017). Using a machine learning model avoids the need to figure out rules and thresholds for the classification phase, which is very labour intensive. I have noticed that the Random Forest machine learning model seemed to perform the best in those related works, and that is why I decided to use it for my study, as well.

We could see that existing works of real-time transportation mode classification only focused on classifying one window at a time without using information from previously seen data (Stenneth et al., 2011; Hemminki et al., 2013; Ellis et al., 2014; Shafique & Hato, 2015; Su et al., 2015). In my thesis, I use information from previous data and use it to calculate features for the current window of data to be classified. This adds some of the advantages from works that process and classify entire trip data after its collection in a non-real-time setting (Chung & Shalaby, 2005; Zheng et al., 2008, 2010; Biljecki et al., 2013; Liang et al., 2017).

Hemminki *et al.*'s (2013), and Shafique's and Hato's (2015) works showed that accelerometer data has a strong contribution in the classification process, which is a reason why I also included accelerometer data in my system.

After examining works that used bus stop locations (Stenneth et al., 2011; Biljecki et al., 2013), I saw that there was more potential to develop additional features based on bus stop locations. That is why I developed new bus stop location-based features for my study.

Table 3.1 shows a summary of all related works mentioned in this chapter.

Table 3.1: A summary of related transportation mode classification works and their classification accuracies, grouped by their data used in the studies.

Authors	**Data Used**	Data Origin	Methods	Classification Accuracy
Zheng *et al.* (2008; 2010)	**GPS**	28 cities in China, USA, South Korea, and Japan	Decision-Trees, graph-based post-processing, non-real-time	76%
Hemminki *et al.* (2013)	**Accelerometer**	Finland (Helsinki), Japan (Tokyo, Amnon, Shirakami, Hiroshaki), Germany (Frankfurt, Saarbrücken), and Luxembourg	AdaBoost and Hidden Markov Model, gravity estimation, real-time	80.1%
Shafique and Hato (2015)	**Accelerometer**	Japan (Niigata, Matsuyama, Gifu)	Random Forest, average resultant acceleration features, real-time, fixed device orientations, very high data sampling rate	99.8%

Chung and Shalaby (2005)	**GPS and GIS**	Canada (Toronto)	Rule-based algorithm and map-matching, non-real-time	92%
Stenneth *et al.* (2011)	**GPS and GIS**	USA (Chicago)	Random Forest, real-time offline classification	93.5%
Biljecki *et al.* (2013)	**GPS and GIS**	Netherlands (Amersfoort and Delft)	Hierarchical fuzzy logic expert system, segmentation based on clustering and distance to bus stop locations, non-real-time classification	91.6%
Liang *et al.* (2017)	**GPS and GIS**	Microsoft GeoLife GPS Trajectories	Random Forest, Bus Stop Rate Feature, non-real-time	86.5%
Ellis *et al.* (2014)	**GPS and Accelerometer**	USA (San Diego)	Random Forest, moving average voting post-processing, real-time	91.9%

Su *et al.* (2015)	**Multiple sensors**	USA (New York City)	Two-step classification using Bayes Net classifiers, real-time online classification	96.4%

Chapter 4

Transportation Mode Classification using GPS-, Accelerometer-, and GIS-Data

In this chapter, we answer the following two questions:

1. Does adding bus stop location data based features to a feature set based on accelerometer and GPS data improve real-time window based transportation mode classification?
2. Is it possible to calculate features based on previously seen data for a single trip and use it to improve real-time window based transportation mode classification?

4.1 Overview

My system to classify transportation modes consists of several stages:

1. Dataset collection,

2. Trip Segmentation,

3. Features Extraction,

4. Model Construction and Data Classification,

The end result of the system will be segmented trips (or trip windows), where each segment is labelled with the predicted transportation mode the person used for the period of the segment.

4.2 Dataset Collection

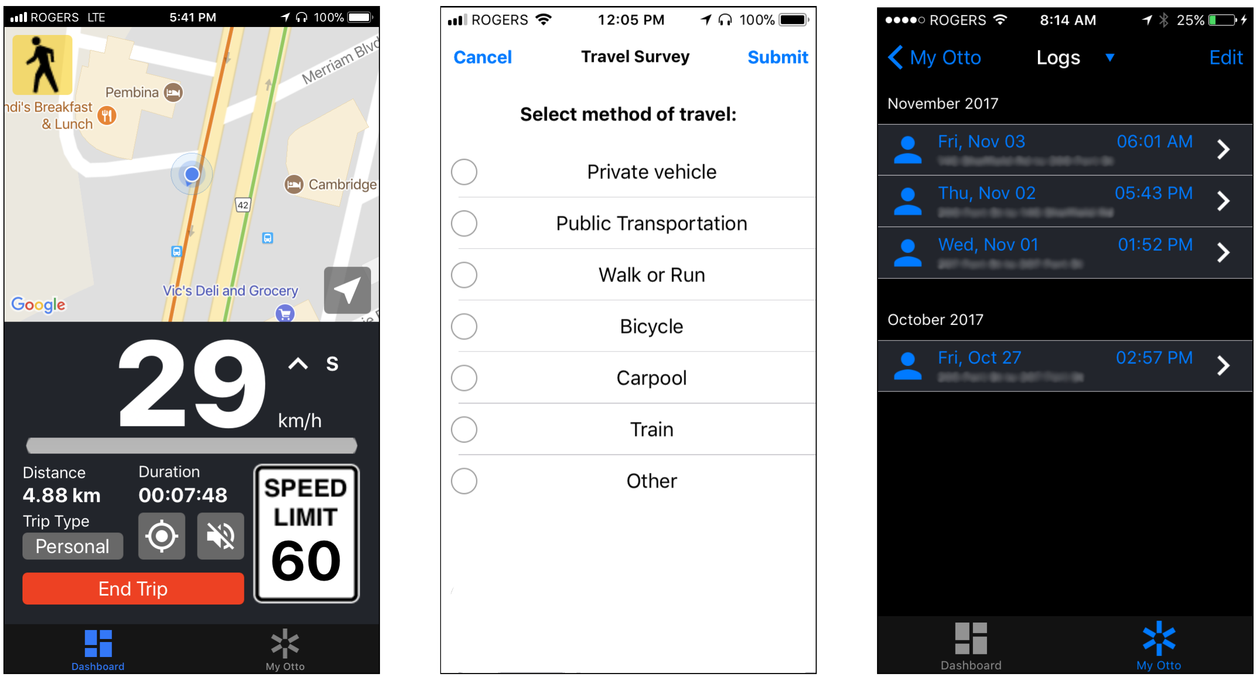
To develop the trip segmentation and transportation mode classification algorithm I need to have the following kind of datasets:

- Trip traces (GPS locations) and trip accelerometer data.
- Winnipeg bus stop locations.

PERSENTECH (2018) already developed an iOS application, "OttoFleet", that has the ability to keep track of a user's movement, *i.e.*, log GPS data at a 1 Hz sampling rate. They also already added the feature to collect accelerometer data at a 10 Hz sampling rate. Figure 4.1 shows the main screens of the application. Figure 4.1a shows the "current trip" screen of the application, where users can see their current trip information, such as speed, alerts and a map. To start a new trip or trip leg the user simply ends a trip and a new trip will automatically start. After a new trip or trip leg starts, the user will be presented with a popup of a list of transportation modes, as seen in Figure 4.1b. Figure 4.1c shows the screen listing a user's saved trip logs, where they can review trip information and delete logs they do not want anymore. Trip logs did not get uploaded directly from the "OttoFleet" application to a server and had to be collected manually for this study.

I also developed a PhoneGap app for iOS and Android, which keeps track of a user's movement, Figure 4.2. Figure 4.2a shows the start screen of the application, where users can manage their recorded trips, start new trip recordings and upload existing trips. If they did not want certain trips to be uploaded or if they mislabelled some trips they could freely delete them. Figure 4.2b shows the transportation mode selection popup that would appear when the user starts a new trip. They could also open the popup anytime from the current-trip screen when they are switching transportation modes. Figure 4.2c shows the screen for current trip information after a user started a new trip. The GPS sampling rate was set to 1 Hz and the accelerometer sampling was at 22 Hz. Intuitively, having different people collect data for my study should help to increase the variability of the dataset, so that the dataset contains different edge-cases, unique travelling patterns and other characteristics related to things like road conditions or traffic flow. Stenneth *et al.* (2011) used six people to collect their dataset over a period of three weeks. I and three other people collected commute data using different transportation modes, using the "OttoFleet" and my own developed application.

Users of the tracking application can, on demand, anonymously and securely, upload saved data directly from their smartphone to a server hosted by me, which only I can access. Data collected with the "OttoFleet" application was transferred manually to the server by me for processing. The data is stored in MongoDB, which has basic geo-spatial query capabilities. I used Node.js for the data management and python with scikit-learn to prepare data and analyze data and train a machine learning classifier for transportation mode classification.

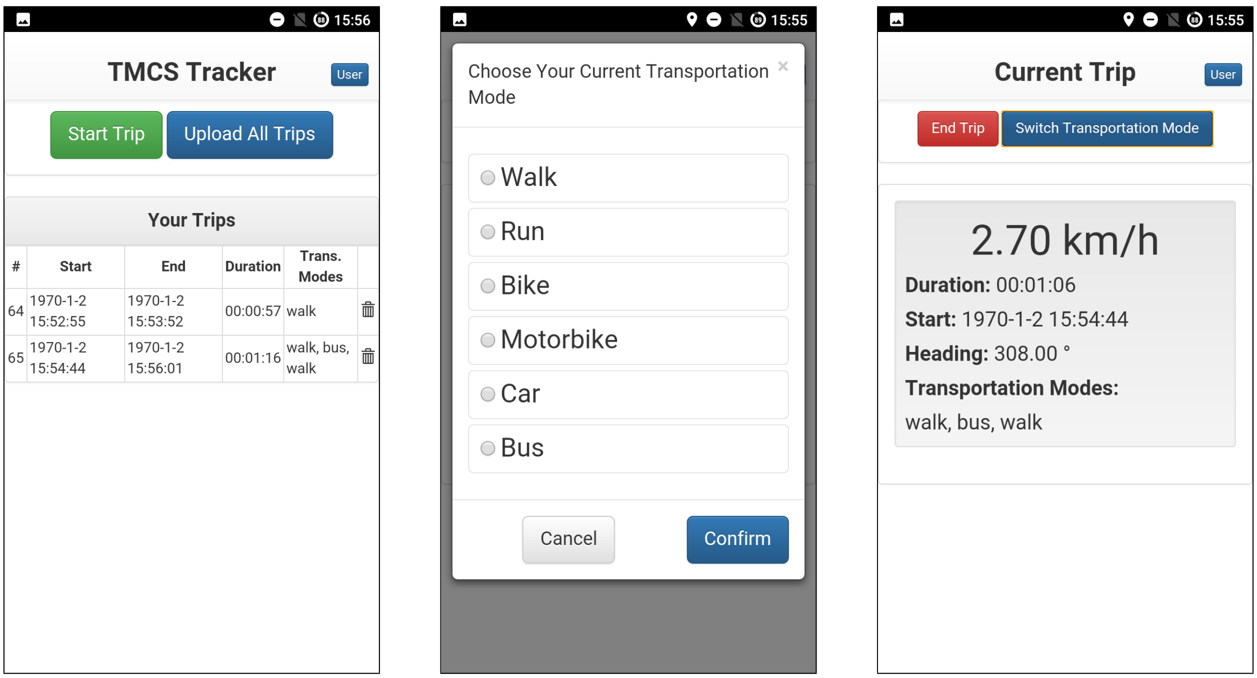


(a) The start screen and information relating to the user's current trip.

(b) Users can select a transportation mode at the beginning of a trip. Trip legs are represented by individual trips.

(c) Users can see a list of their stored trips.

Figure 4.1: Screenshots of the tracker application developed by Persentech for collection of GPS and Accelerometer data.



(a) The start screen and the user's list of trips.

(b) Users can select a transportation mode at the beginning of a trip and during a trip when transitioning between different modes.

(c) Users can see some information relating to their current trip.

Figure 4.2: Screenshots of the tracker application developed by me for collection of GPS and Accelerometer data.

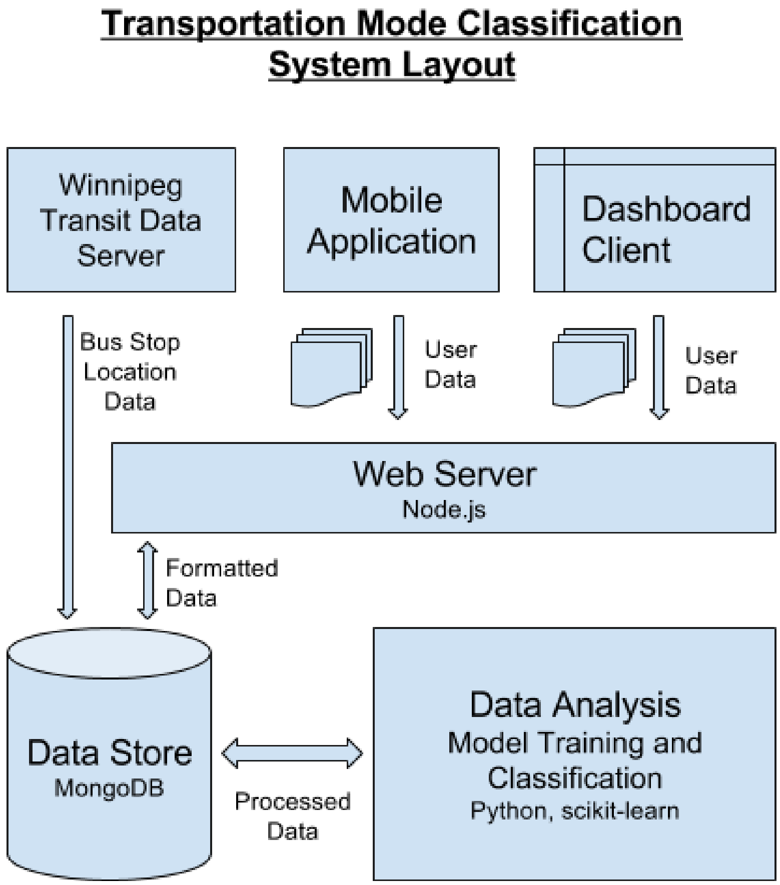


Figure 4.3: The system overview diagram showing the flow of data from mobile applications and a web dashboard into the server, where it is stored and further used for transportation mode classification

Figure 4.3 shows the system overview. Clients (*i.e.*, mobile application and web dashboard) connect to the Node.js web server, which prepares and consolidates the trip data received from the clients and stores everything in MongoDB. Bus stop locations from the Winnipeg Transit API is queried by a cron job and stored in MongoDB, too. A data analysis process, developed in python, queries trip data and bus stop locations in order to build a transportation mode classifier.

Figure 4.4 shows the data analysis process overview. The process queries all raw trip data stored in MongoDB, which includes segments of GPS and accelerometer data together with their transportation mode labels. The data is split up into small windows, which is described in Section 4.3. Later features are extracted from the data windows and stored as vectors for each corresponding window as well as a corresponding vector with transportation mode labels, which are used for the classifier training. Some features in the vectors are based on bus stop locations, which are queried from MongoDB for each GPS point in a data Window during the features extraction phase. The features vectors are used by scikit-learn to train a transportation mode classifier based on the Random Forest model. After the classifier is trained it is tested on trip data that was not used during the training phase of the classifier. The results are trip data windows labeled with transportation modes.

The total amount of data obtained is 488 trips, totalling 226 hours. Trips were recorded throughout the year 2016. Hence, the dataset represents trips with different weather and road conditions from summer and winter times. Users of the application also recorded, which transportation modes they used at the time of commute by using the application's survey feature. A survey prompts the user at the beginning of every trip with the following possible answers: "Walk", "Bicycle", "Car", and "Bus". That results in a dataset of trip data labelled with four classification labels provided by the users. In the classification system, four corresponding classification labels with the same names are used for the results.

Adding GIS data in addition to GPS and accelerometer data to train a transportation mode classification model could potentially improve the accuracy of the model but using GIS data adds another dependency for a transportation mode classification algorithm. If the GIS data used is more difficult to obtain then the trained classification model will not

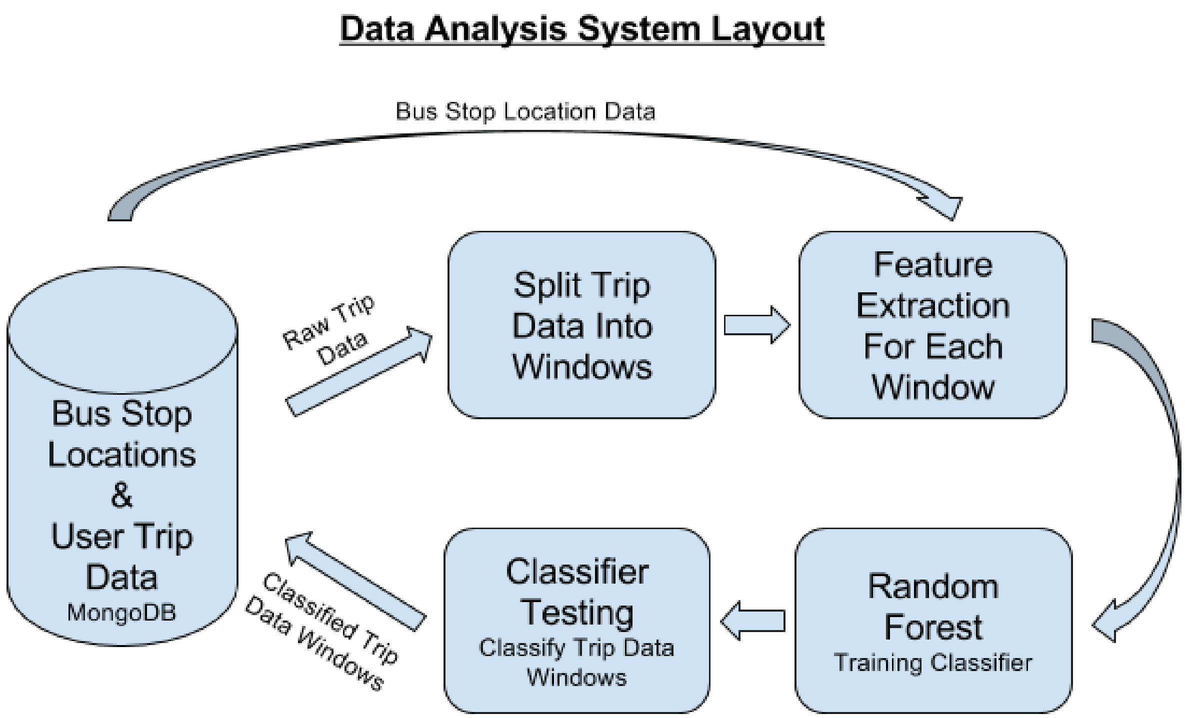


Figure 4.4: The diagram shows an overview of the transportation mode classifier training process, which uses trip data, such as, GPS data and accelerometer data, and bus stop locations.

be usable in cities where the dependent GIS data is not available. For example, real-time bus location data is not as available as bus stop location data. For that reason, to avoid unavailability of GIS data needed for the classification model, the only GIS data used in this work are bus stop GPS location data. These data are available from Winnipeg Transit and can be easily retrieved from the Winnipeg Transit Open Data Web Service API (Winnipeg Transit, 2018). The bus stop location data were collected during the time we collected the trip traces data to ensure that I have the most relevant GPS locations. To collect that dataset, a script was written to automatically retrieve the bus stop locations in Winnipeg during the period of trip data collection. I retrieved a total of 5129 bus stop locations for the city of Winnipeg during the period of the data collection.

4.3 Trip Segmentation

A trip is simply a collection of data points collected during a person's entire commute from origin to destination (*e.g.*, from home to work). The data collected in this study is GPS and accelerometer based data. Each trip can further be segmented into smaller legs, *e.g.*:

1. from home to the departure bus stop,
2. from departure bus stop to destination bus stop, and
3. from destination bus stop to office.

When data for a trip is collected the trip needs to be segmented in some way into such legs where transportation modes changes occur. In the previous example we see three legs, where

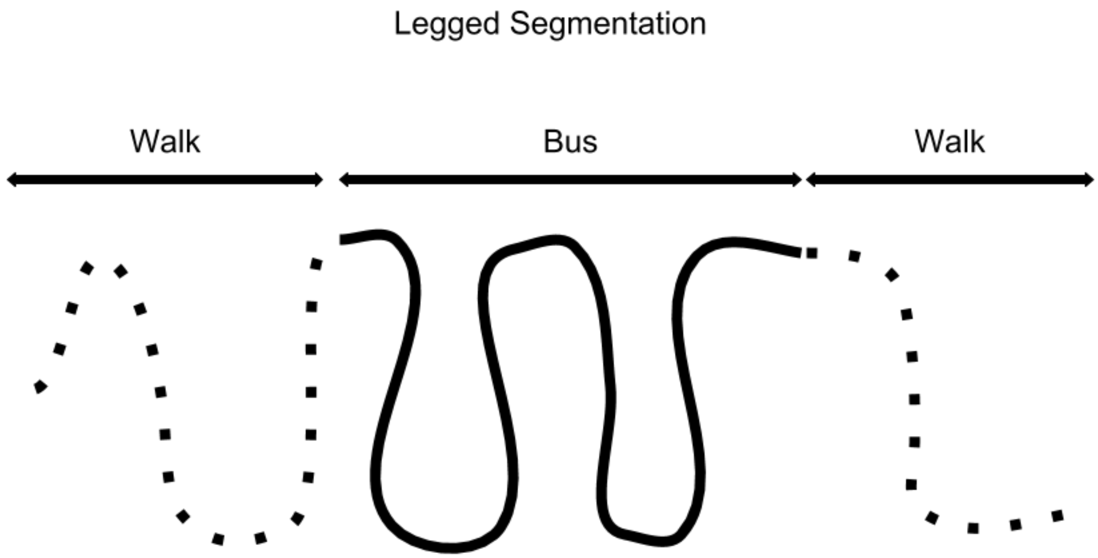


Figure 4.5: A trip trajectory example segmented into separate legs by transportation mode.

each leg uses a different transportation mode, 1. Walk, 2. Bus, 3. Walk, which is depicted in Figure 4.5.

The system designed in this thesis automatically achieves the trip segmentation by simply classifying the transportation mode of each window of data. This way a user of the classifier could theoretically easily recognize the different legs of a trip because they are labelled with the transportation mode. The data of a trip is divided into windows of data of equal size. Read more about the determined appropriate window size in Section 5.4. So, instead of trying to segment trips into legs right away, the system separates data into windows of equal time intervals, which then get classified one-by-one (see Figure 4.6 for an example). When splitting data into windows, it is important not to to mix data points labelled with different transportation modes in one window. Whenever a transportation mode change occurs the different data needs to be assigned to different windows, because training a machine learning

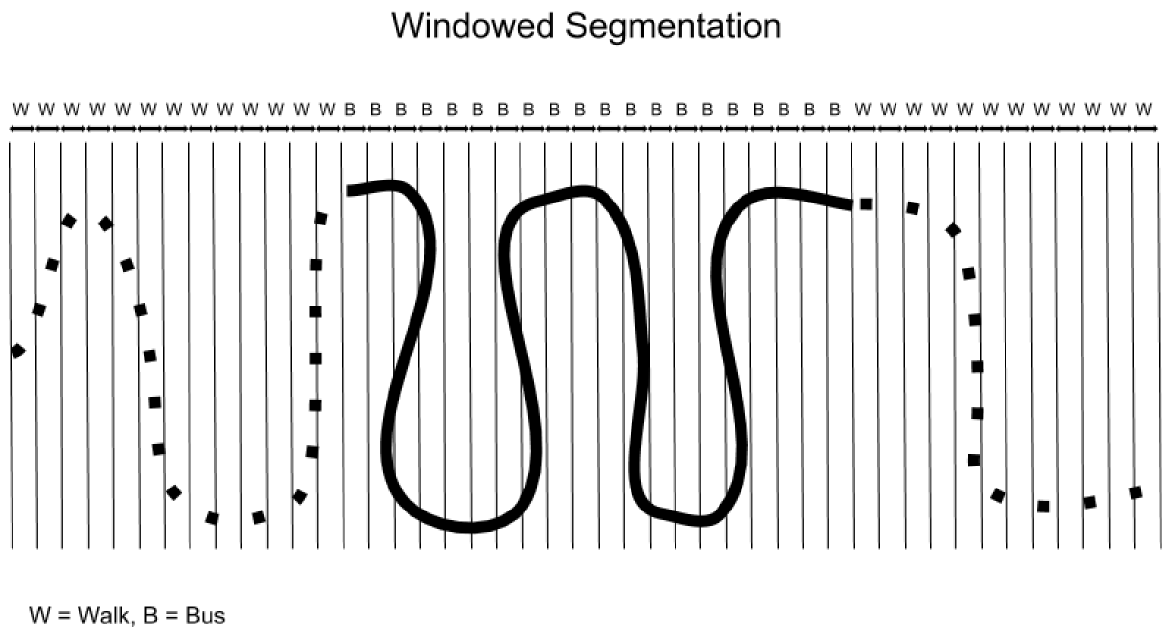


Figure 4.6: A trip trajectory example segmented into windows of equal time intervals, labelled with transportation modes.

model with wrongly labelled data could significantly reduce classification performance. Although, when the window size is very small, the effect of having mixed transportation modes in one window is theoretically not as significant as for bigger windows. Segmenting the data into small windows has the benefit that classification can be performed in real-time, *i.e.*, as soon as enough data has been collected to fill a new window. Once every window is classified with a transportation mode then the user simply needs to concatenate the windows/trip segments, which now have transportation mode labels, and present each label on a map with a colour-scheme for different transportation modes. Once the trip is rendered on a map the user can easily identify the different legs of the trip by simply looking at the different colours of the trip. The window segmentation approach is a commonly used technique used in other existing real-time classification systems, as described in Chapter 3.

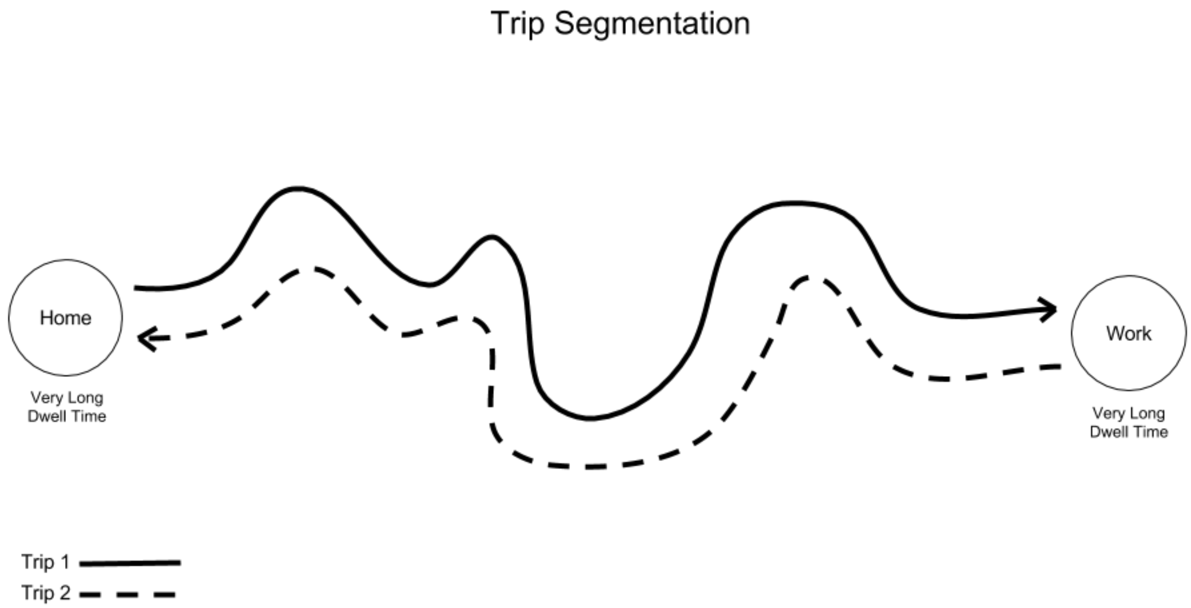


Figure 4.7: A trajectory example segmented into two trips based on long dwell times.

Another segmentation problem is the matter of segmenting data into individual trips in the case where a user has data that is not separated into trips yet. This problem can easily overcome by searching the entire data for very long dwell times. Usually, when people commute, they tend to have very long dwell times at their origin and destination places. To detect such long dwell times the user needs to define an appropriate minimum threshold for their use-case. For example, a threshold of 6 hours would likely find trip legs that are temporally between staying at home and staying at work (see Figure 4.7 for an example). Whereas, a threshold of 1 hour might even capture trip legs between the times of staying home and staying at work, such as grocery shopping trips on the way home and a drive to a restaurant during a lunch break.

4.4 Features Extraction

In the following sections, the features used for training the transportation mode classifier, are discussed. For each window, using all records within it, all features get calculated for GPS, accelerometer and bus stop location based data. An appropriate window size is determined in Chapter 5. Section 4.5 (GPS Data Features), Section 4.6 (Accelerometer Data Features) and Section 4.7 (Bus Stop Location Features) briefly describe all the features for GPS, accelerometer and bus stop location based data, including the newly developed features. Section 4.8 (Dwell Time), Section 4.10 (Window History Queue) and Section 4.9 (Dwell Time History) describe the contributions of my thesis.

4.5 GPS Data Features

GPS data was collected at a 1Hz rate. From the GPS data we can extract the following features for the machine learning process. Those features are not new features and are some of the standard ones used throughout other studies as mentioned previously in Chapter 3.

1. *Average Speed* (km/h)

2. *Average Altitude* (m)

3. *Average Location Accuracy* (m)

4. *Maximum Speed* (km/h)

5. *GPS Signal Loss* (no-signal = 0 and signal = 1)

6. *Travel Distance* (m), Haversine Distance, which is a geodesic distance

The following features are calculated by considering the previous windows history. A window history queue, Q^{wh}, of length 15 of previous windows is kept at all times (read more about the window history queue length in Section 4.10). For each current window the following features are calculated for the machine learning process:

1. *Previous Average Speed*, which is the average speed for all previous windows in the queue.

2. *Previous Maximum Speed*, which is the maximum speed for all previous windows in the queue.

3. *Previous Maximum Average Speed*, which is the maximum average speed for all previous windows in the queue.

4. *Delta Average Speed*, which is the difference between the current window average speed and the previous windows average speed.

5. *Delta Maximum Speed*, which is the difference between the current window maximum speed and the previous windows maximum speed.

6. *Delta Maximum Average Speed*, which is the difference between the current window average speed and the previous windows maximum average speed.

More features can be calculated by using the GPS location. Those features will be discussed in Section 4.7.

In Figure 4.8, we can see a bus moving along part of its route. The locations of the bus at each data window are represented by objects B1 to B10. The distance and speed between each bus locations are decreasing when a bus decelerates to stop at a bus stop, and those values increase when the bus accelerates and moves away from the bus stop. The delta values, such as *Delta Average Speed* and *Delta Maximum Average Speed*, capture the rate of acceleration and are used in the feature window. Capturing the acceleration should help to differentiate between transportation modes because the modes tend to have different acceleration rates. For example, a bus stops slower than when a person walks and then stops. A bus also decelerates from a much greater maximum average speed to 0, than a person walking.

4.6 Accelerometer Data Features

Accelerometer data was collected at a 21Hz rate, which means that for every GPS record there will also be approximately 21 accelerometer data records. From the accelerometer data we can extract the following 13 features for the machine learning process:

1. *Average Magnitude*
2. *Magnitude Standard Deviation*
3. *Magnitude* 25^{th} *percentile*
4. *Magnitude* 75^{th} *percentile*
5. *Minimum Magnitude*
6. *Maximum Magnitude*
7. *Lag-1 Autocorrelation*
8. *Correlation between axes X & Y*

9. *Correlation between axes X & Z*

10. *Correlation between axes Y & Z*

11. *Average Roll*

12. *Average Pitch*

13. *Average Yaw*

The following features are calculated by considering the previous windows history. A window history queue, Q^{wh}, of length 15, of previous windows is kept at all times (read more about the window history queue length in Section 4.10). For each current window the following features are calculated for the machine learning process:

1. *Previous Average Magnitude*, which is the average of all Average Magnitudes in Q^{wh}

2. *Previous Magnitude Standard Deviation*, which is the average of all Magnitude Standard Deviations in Q^{wh}

3. *Previous Magnitude 25th percentile*, which is the average of all Magnitude 25$^{\text{th}}$ percentile in Q^{wh}

4. *Previous Magnitude 75th percentile*, which is the average of all Magnitude 75$^{\text{th}}$ percentile in Q^{wh}

4.7 Bus Stop Location Features

Bus stop location based features can be calculated based on a person's current GPS location and the bus stop GPS locations near that current location. For each record in a window of GPS data, all unique bus stops within 5 meters are queried from the bus stops database (read more about this threshold in Section 4.8 and Section 5.6). Afterwards, for the entire window, the following features can be calculated for the machine learning process:

1. *Stopped at a bus stop* (stopped = 1, not-stopped = 0), the person stopped at any of the nearby bus stops within the window.

2. *Number of stops at bus stops*, which is the number of stops nearby all bus stops within the window.

3. *Distance to closest bus stop stopped at*, which is the distance to the bus stop closest to the person within the entire window.

4. *Number of bus stops*, which is the number of unique bus stops within the window.

5. *Dwell Time*, which is the time a person is nearby a unique bus stop within the window and has a speed of 0 km/h (read more about this feature in Section 4.8)

6. *Total Long Dwell Time*, which is the total time of dwell times calculated from a set of consecutive windows with dwell times of at least 1 second, without a gap (read more about this feature in Section 4.9)

7. *Numbers of Long Dwell Time*, which is the total number of long dwell times encountered in all previous windows (read more about this feature in Section 4.9)

4.8 Dwell Time

"Dwell Time" is the time a person dwells near a bus stop. The idea behind this measurement is that if a person takes the bus during a commute then they usually spent some

significant amount of time at the departure and transfer and arrival bus stops. From experience, those kind of dwell times tend to be in the order of minutes. While the bus is in motion, it stops at bus stops along the bus route, either to pick up new passengers and/or let passengers off the bus. Those kind of dwell times are usually shorter but more frequent. Those short dwelling times near a bus stop could also be used in the classifier training to better identify bus mode windows. Theoretically, this feature combined with some of the parameters containing information about previous windows, such as *Previous Average Speed*, *Previous Maximum Speed*, and *Dwell Time History*, should help the the classifier to better differentiate between a car and a bus ride; More dwell time based parameters can be found in Section 4.5 and Sections 4.9 to 4.10.

This feature is measured in seconds and is calculated for every window. The nearest bus stop location is fetched based on a radius maximum threshold, D^{bus_stop}= 5m, which is determined in Section 5.6. That threshold cannot be too high but also not too low. On one hand, if D^{bus_stop}= 1m, the algorithm might not capture all dwell times because buses do not always stop exactly within 1m of a bus stop. On the other hand, if D^{bus_stop}= 50m, the algorithm might capture too many bus stops before the bus actually stops at the stop and if the person is actually in a car then false dwelling times at bus stops might occur even more often.

In Figure 4.8, we can see a bus moving along part of its route. The locations of the bus at each data window are represented by objects B1 to B10. At positions B5 to B8 the bus has stopped at the bus stop and dwells there for some time, which is represented by overlapping bus objects. When a bus dwells we calculate the dwell time at a bus stop only if the bus is close enough to a bus stop, which is determined by the radius threshold D^{bus_stop}= 5m.

Often a particular bus stop has another bus stop nearby on the other side of the street for the bus route that travels in the opposite direction. Sometimes these stops are offset across street intersections. A radius maximum threshold D^{bus_stop} helps fetch only bus stops that are on the correct side of the street.

The dwell time parameter is important for calculating the *Dwell Time History* feature, which is theoretically a good parameter for reducing the ambiguity between buses and cars data when predicting a current window's transportation mode; Read more about dwell time history in Section 4.9.

4.9 Dwell Time History

In Section 4.8 we have seen a dwell time feature that is calculated for an individual window of data. The *"Dwell Time History"* feature described in this section focuses on parameters calculated based on the window dwell times for all previous windows of data for a trip. There are two parameters that get calculated, "Long Dwell Time" and "Numbers of Long Dwell Times".

A *"Long Dwell Time"* is defined as the total time of dwell times calculated from a set of consecutive windows with dwell times of at least 1 second, without a gap. If a window is encountered where the dwell time is 0, then that is where the "Long Dwell Time" ends. The idea behind this parameter is that if this number is high enough (as determined by the training of the classifier), then it is more likely that the current window could be a bus mode window. If a person takes the bus, firstly, the person usually dwells for a relatively long time at the departure bus stop waiting for their bus to arrive. Those consecutive windows of

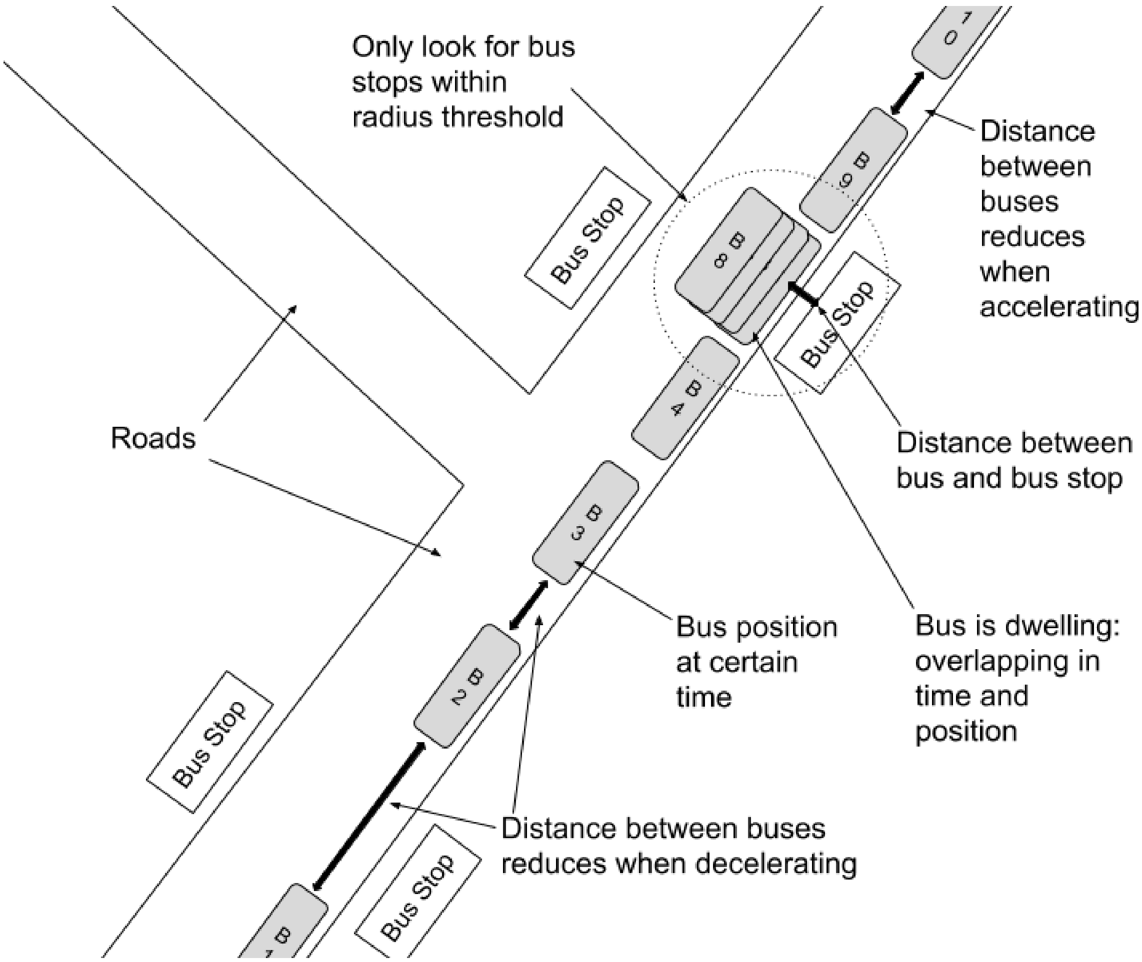


Figure 4.8: An example of positions in time of a bus moving along part of its route, dwelling at a bus stop.

dwelling are counted towards the "Long Dwell Time", which in turn are added to the "Total Long Dwell Time". Once the person gets on the bus and the bus moves along its route encountering many bus stops and also stopping at most of them, then there will be many long dwell times for the whole trip, which are considered "Long Dwell Times" and are added to the "Total Long Dwell Time", as well. Thus, one can see that the "Total Long Dwell Time" might be a good indicator for transportation mode classification and it is also not difficult to calculate and keep track of. In fact, it is easy to keep track of this feature value in real-time, which makes a good candidate for real-time transportation mode classification.

The *"Numbers of Long Dwell Times"* is simply the total number of long dwell times encountered in all previous windows. This number is essentially interpreted as the number of times a bus stops along its bus route. If this number is relatively high, then it could be likely that a person is commuting on a bus. Theoretically, a bus should stop more often at bus stops than a car following the same route. Thus, the "Numbers of Long Dwell Times" feature might be a good contributor for transportation mode classification. This value together with the "Total Long Dwell Time" seem to be good theoretical features for transportation mode classification.

In Figure 4.9 we see a bus trip example going from home to work. Along the bus route there are many bus stops on both sides of the street. D_1 to D_7 represent the bus stops where the bus stopped and dwelled for a longer time to let passenger on or off the bus. As described above, during a trip we track the number of such long dwell places and the total long dwell times. In the example in Figure 4.9, we see 7 long dwell places. Let us assume $D_1 = 300$s, $D_2 = 30$s, $D_3 = 10$s, $D_4 = 20$s, $D_5 = 50$s, $D_6 = 120$s, and $D_7 = 10$s, then the total long dwell time would be 540 seconds.

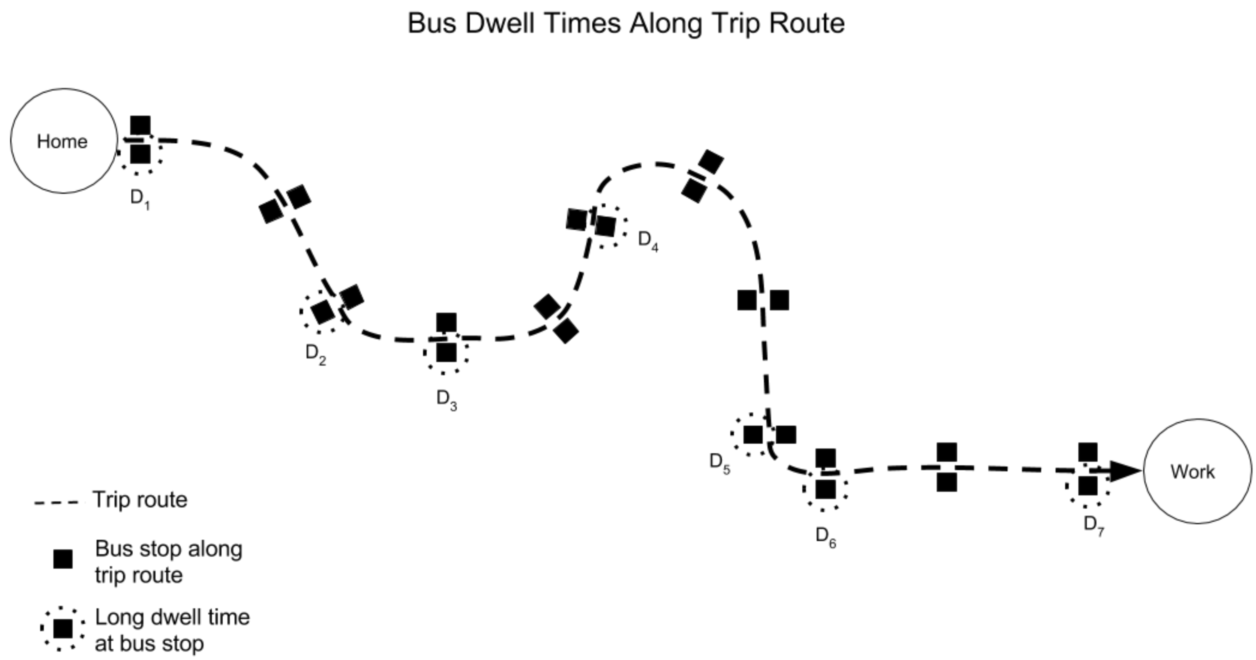


Figure 4.9: An example of a bus trip from home to work where the bus dwells at 7 bus stops during the trip.

4.10 Window History Queue

The *"Window History Queue" (WHQ)* is another feature that helps to blend the real-time and non-real-time transportation mode classification. The idea behind a WHQ is that previous windows data could help to classify the current data window. If previous data is similar to the current data, such as average speed, then the current data is more likely to have the same classification as the previous data. That also means that if the current and previous data are very different then there is a chance that transportation modes might change in the following windows. The classifier should be able to determine the subtle data differences during the training phase. For example, the random forest model would learn that a previous high average speed followed by a very low average speed would mean that the current data should represent the transportation mode "Walk". The data coming from the GPS sensor contains some errors and could cause significant differences in average data between windows for the classification model, this could potentially be compensated by taking aggregate data of multiple windows. Results of tests in Section 5.5 showed that overall accuracy increases linearly with increasing "Window History Queue" size, $|Q^{WH}|$. $|Q^{WH}| = 15$ was chosen in this study.

The following features were used in summarizing the window history queue and were then used as features for the classifier model.

GPS-based $\mathbf{Q}^{WH}$ features:

- *Previous Average Speed*, which is the average speed for all previous windows in the queue.
- *Previous Maximum Speed*, which is the maximum speed for all previous windows in the queue.
- *Previous Maximum Average Speed*, which is the maximum average speed for all previous windows in the queue.
- *Delta Average Speed*, which is the difference between the current window average speed and the previous windows average speed.
- *Delta Maximum Speed*, which is the difference between the current window maximum speed and the previous windows maximum speed.
- *Delta Maximum Average Speed*, which is the difference between the current window average speed and the previous windows maximum average speed.

Accelerometer-based Q^{WH} features:

- *Previous Average Magnitude*, which is the average of all Average Magnitudes in Q^{wh}
- *Previous Magnitude Standard Deviation*, which is the average of all Magnitude Standard Deviations in Q^{wh}
- *Previous Magnitude 25th percentile*, which is the average of all Magnitude 25$^{\text{th}}$ percentile in Q^{wh}
- *Previous Magnitude 75th percentile*, which is the average of all Magnitude 75$^{\text{th}}$ percentile in Q^{wh}

4.11 Model Construction and Data Classification

As noted in other sections, the chosen classification model is Random Forrest (Breiman, 2001). In most transportation mode classification studies it performed the best.

Before training the classification model, the data needs to be prepared. The data that the model needs are the calculated features based on the collected raw data (see sections above). The data features were stored on a per trip-basis. For each trip there is a set of feature windows. The trips for each transportation mode are shuffled and then split into two sets, T_{train} and T_{valid}. T_{train} represents a set of 70% of trips of each transportation mode and T_{train} is used for the training and testing phase (cross-validation and training). T_{valid} represents the remaining 30% of trips, which are used for validating the trained model.

After splitting the data, all windows from all trips in T_{train} are used for a stratified 10-Fold cross-validation test to confirm whether the classification model is appropriate. Once that is confirmed, the model is trained on the entire set of T_{train} to produce a classifier (a trained model) that is used for transportation mode classification. The classifier is then used for classifying all windows for all trips in T_{valid} and the results are verified with the corresponding labels set, L_{valid}, for T_{valid}. By withholding T_{valid} from the development phase and classification model training we can test the performance of the classifier on new unseen data.

4.12 Summary

In this chapter, I presented my method and details of the data collection of GPS and accelerometer trip data. Android and iOS smartphone applications were used to collect 488 trips, totalling 226 hours, and label them with transportation modes (*i.e.*, "walk", "bike", "bus", "car") during the recording phase. I also gave an overview of my data collection system, where the trip data was stored and prepared for the transportation mode classifier training process.

An overview of the transportation mode classifier training process showed how trip data gets split into data windows, and from each data window trip data features are extracted. The new trip data feature windows will then be used to train and test a Random Forest classifier.

In this chapter, I have also showed all the features that I have used for the classifier. I listed GPS and accelerometer features that are commonly used in other research works of transportation mode classification, and I presented my newly developed features that have the potential to improve transportation mode classification accuracy in a system that segments data into small windows for real-time applications.

I have developed bus stop location based features around the notion of Dwell Time nearby bus stops, which have not yet been used in other similar research work. Based on dwell time throughout a trip, I calculate *Dwell Time History* features, which are also my new features:

- *Number of Long Dwell Times at bus stops*
- *Total Long Dwell Time at bus stops*

Those features target bus ride data, capturing frequent stopping at bus stops and waiting for a bus at a bus stop. In principal, the features keep a record of how often a person stops at bus stops and how much total time they spend at bus stops. Those values are updated throughout the trip and are used when processing each new window of data. The *Dwell Time History* features seem to be good theoretical features for transportation mode classification and should help to remove some ambiguity between bus ride data features and car ride data features.

I have also developed a new *Window History Queue* technique that allows to incorporate knowledge from previously seen data windows into the currently evaluated window by adding the extra information as additional features. Basically, those features are sums and averages of previously seen data from a fix length queue of feature windows, and improve classifier training by either contrasting current data or being similar to current data, which adds more context to the current window data and helps the trained model to distinguish data windows when transportation mode changes occur and detect similar data windows, *e.g.*, where a person gets on a bus at a bus stop or gets off the bus at their destination. The *Window History Queue* is used for both, GPS and accelerometer data, and includes features, such as, *Previous Average Speed*, *Previous Maximum Speed*, *Previous Average Magnitude*, *Previous Magnitude* 75^{th} *percentile, etc.*

Chapter 5

Experiments and Results

In the previous chapters, the background and the transportation mode classification system with new classification parameters were described, which are bus stop location-based features and a parameter history queue. Those new modifications to a general transportation mode classification system have the potential to improve overall accuracy of such a system. To determine the effectiveness of the new features, this chapter describes the testing experiments and results, which will help to conclude whether the new system improves transportation mode classification accuracy.

5.1 Experiment Setup and Dataset Details

All tests and classifier training were run using a computer running Ubuntu 16.04 LTS as the main operating system. The CPU was an AMD Phenom II X6 1100T with 6 cores clocked at 3.3 GHz to 3.7 GHz. The machine also had 16 GB of RAM and a Solid State Drive.

The total amount of data obtained is 488 trips, totalling 226 hours. Trips were recorded throughout the year 2016. Hence, the dataset represents trips with different weather and road conditions from summer and winter times. The entire dataset was collected by four people living in different parts of the city of Winnipeg. If possible, the number of people collecting data should preferably be as high as possible to add as much variation as possible to the dataset. In this study, three of the people moved once to different regions of the city in the year the data was collected. As a result of that, more variation was added to the total dataset, covering more parts of the city, adding different commute routes and even changing regular transportation modes. Additionally, one person did not adhere to a regular bus schedule throughout the week, often taking different bus connections and routes between the same origins and destinations, which introduced more variation to the dataset. The following list shows how many segments and hours were collected for each transportation mode:

- "Walk": 352 segments, 65 hours
- "Bike": 32 segments, 8 hours
- "Car": 317 segments, 111 hours
- "Bus": 108 segments, 42 hours

5.2 Test Overview

The data features were stored on a per trip-basis. For each trip, there is a set of feature windows. The trips for each transportation mode are shuffled and then split into two sets, T_{train} and T_{valid}. T_{train} has 70% of the data and is used in this testing phase. Throughout all tests we use a stratified 10-fold cross-validation with a 50-50 split for training and testing for each partition, where the dataset-splitter chooses the same percentage of samples from each class to ensure that the models are trained and tested using all possible classes. After every cross-validation test, a classifier, which is called C_{valid}, is trained with data from T_{train} and is then used to classify data from T_{valid}, which has 30% of all data. The last validation test on T_{valid} simulates a classification test on new unseen data.

5.3 Random Forest Model Verification Test

To confirm whether a Random-Forest classification model is appropriate enough for this classification problem it should be enough to run a few rounds of training and classification on T_{train}. A good expected accuracy percentage range should be 80% and above. A quick stratified 10-fold cross-validation test without any new features that were mentioned above shows a about *97%* accuracy, which indeed falls in our expected range.

5.4 Feature Window Size Test

As previously described, trip data is split into windows of a specified time frame, W. To determine a good window length, $|W|$, we run the system test with different sizes for $|W|$, where $|W| = 4..60$s. While changing $|W|$ during the test, everything else is kept constant. The window history queue, Q^{WH}, will not be used and the bus stop location distance, D^{bus_stop}, stays set to 3 meters. The data we use for the window size test is GPS, accelerometer, and bus stop location data based data.

From Table 5.1 and Figure 5.1 we can see that a window size of 4 seconds is a good size for this classification system because in the test above $|W| = 4$s yields the best transportation

Table 5.1: Classification accuracy test results for varying window sizes.

Window Size (in seconds)	10-fold Cross-validation (in %)	Validation Set Test (in %)
4	**96.9647**	95.5263
5	96.8275	95.7070
6	96.7107	93.5952
7	96.6808	95.6894
8	96.5474	93.6794
9	96.5229	93.5943
10	96.4883	93.8784
15	95.9774	93.5965
20	95.8580	93.2559
30	95.5917	93.7817
40	95.2833	94.3435
50	94.9798	95.0178
60	94.4111	95.6522

mode accuracy. For now, we will use a window size of 4 seconds for all other tests while we change other parameters of the system. Once more features are used in the system that it could be possible that the window size might not be best size anymore because other new features could potentially work much better with a different window size.

5.5 Window History Queue Test

This section presents the tests for the appropriate queue length of the new *window history queue* feature, Q^{WH}. Similarly, as for the window size test, a good window history queue length can be found by keeping all variables in the system constant and vary the queue length, $|Q^{WH}| = 1..15$. With $|W|= 4$, this will give us a data range from 4 to 60 seconds. D^{bus_stop} is set to 3 meters. The data we use for the window history queue test is GPS, accelerometer, and bus stop location data based data.

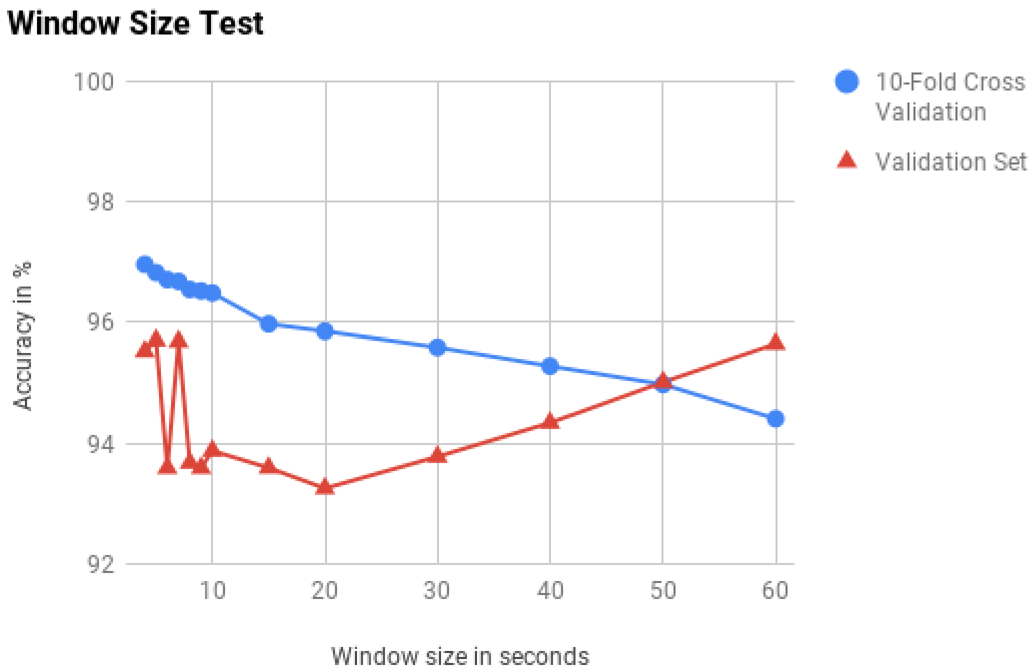


Figure 5.1: Classification accuracy test results for varying window sizes.

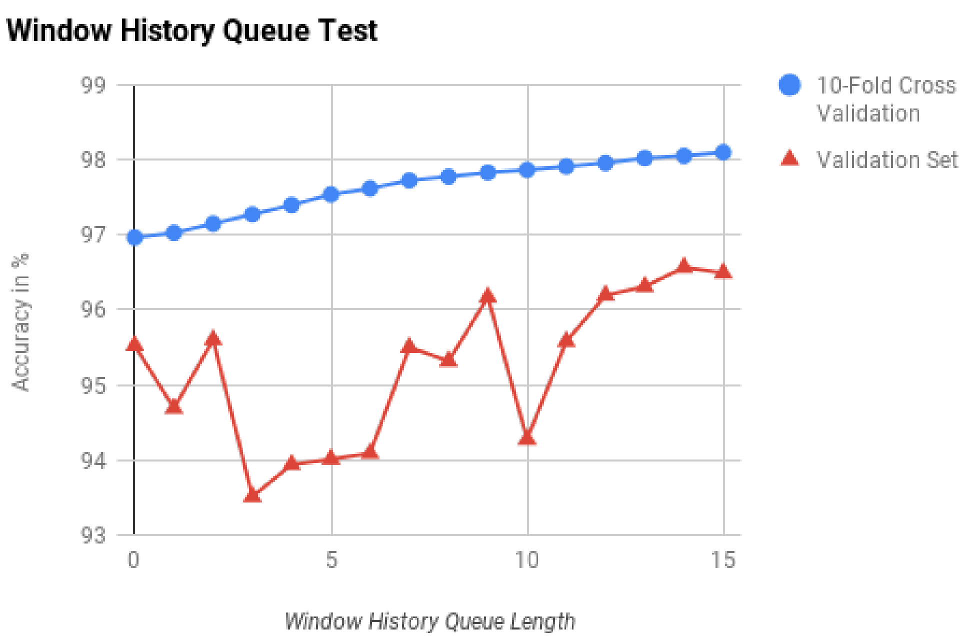


Figure 5.2: Classification accuracy test results for varying window history queue length.

Table 5.2: Classification accuracy test results for varying window history queue length.

Window History Queue Length	10-fold Cross-validation (in %)	Validation Set Test (in %)
0	96.9647	95.5263
1	97.0284	94.6902
2	97.1489	95.6035
3	97.2745	93.5132
4	97.3995	93.9392
5	97.5385	94.0111
6	97.6188	94.0910
7	97.7258	95.5024
8	97.7778	95.3186
9	97.8316	96.1707
10	97.8670	94.2774
11	97.9124	95.5822
12	97.9596	96.1947
13	98.0229	96.3119
14	98.0543	96.5675
15	**98.1000**	96.493

From Table 5.2 and Figure 5.2 we can see that the accuracy increases linearly with $|Q^{WH}|$, where $|Q^{WH}| = 15$ yields the highest accuracy. Thus, a queue length of 15 is chosen for this classification system. For now, we will use length 15 for Q^{WH} for all other tests while we change other parameters of the system. Similarly, as in the window size test, the determined queue length might not be best value anymore once other new features are added to the system, which could potentially affect the overall accuracy. In Figure 5.3, we can observe that using larger values for the window history queue length does not yield a significant classification accuracy increase anymore.

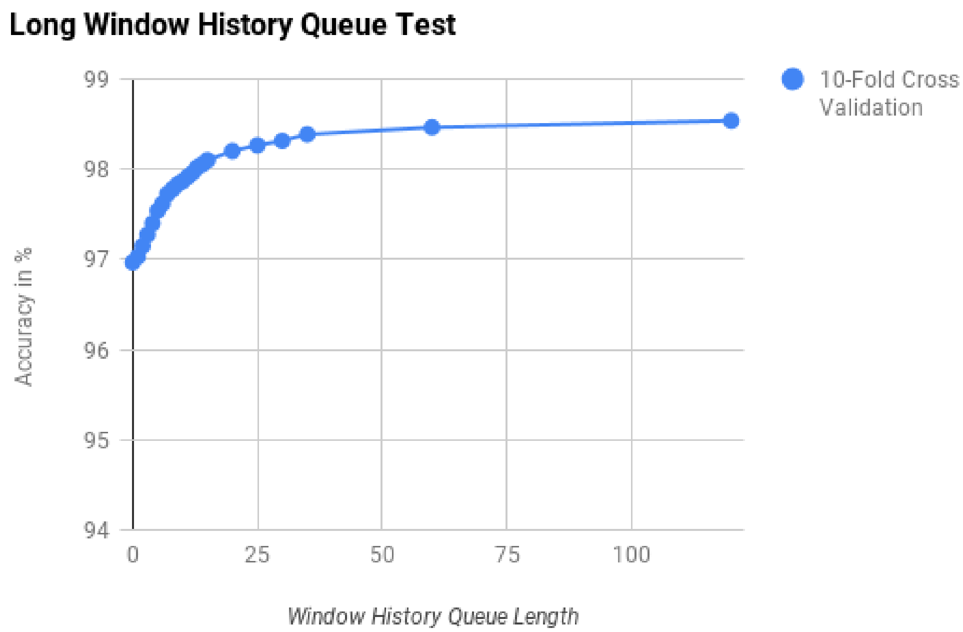


Figure 5.3: Classification accuracy test results for large window history queue lengths.

5.6 Nearby Bus Stop Distance Test

This section presents the tests for the appropriate bus stop distance, D^{bus_stop}, which is used for searching bus stop locations nearby the current GPS location. Similarly, as in the other previous tests, a good D^{bus_stop} can be found by keeping all variables in the system constant and varying the bus stop distance, $D^{bus_stop} = 1..50$m. $|W|$ is set to 4 seconds and $|Q^{WH}|$ is set to 15. The data we use for the bus stop distance test is GPS, accelerometer, and bus stop location data based data.

From Table 5.3 and Figure 5.4 we can see that the accuracy for all tested distances is very similar. If we look at the validation set test we can see that the accuracy is better for low numbers for D^{bus_stop} with D^{bus_stop}= 5m yielding the highest accuracy. Thus, a bus stop distance of 5 meters is chosen for this classification system. For now, we will use 5 meters

Table 5.3: Classification accuracy test results for varying nearby bus stop distance.

Nearby Bus Stop Distance	10-fold Cross-validation (in %)	Validation Set Test (in %)
1	98.5205	94.3010
2	98.5412	94.3137
3	98.5355	94.3773
4	98.5402	94.1229
5	**98.5402**	**94.7462**
6	98.5412	94.2374
7	98.5402	94.1229
8	98.5412	94.3900
9	98.5346	94.4409
10	98.5468	94.4155
15	98.5524	91.8967
20	98.6002	92.2020
25	98.6340	92.9271
30	98.6002	92.6854
35	98.5983	93.0543
40	98.6012	92.9398
45	98.5843	92.8762
50	98.6021	93.3342

for D^{bus_stop} for all other tests while we change other parameters of the system.

5.7 Data Combination Test

The previous sections described the tests used for finding good parameters for the transportation mode system. The following parameters were determined and are being used in the following tests:

1. Window Size, $|W|$= 4s,
2. Window History Queue Length, $|Q^{WH}| = 15$, and
3. Nearby Bus Stop Distance, D^{bus_stop}= 5m

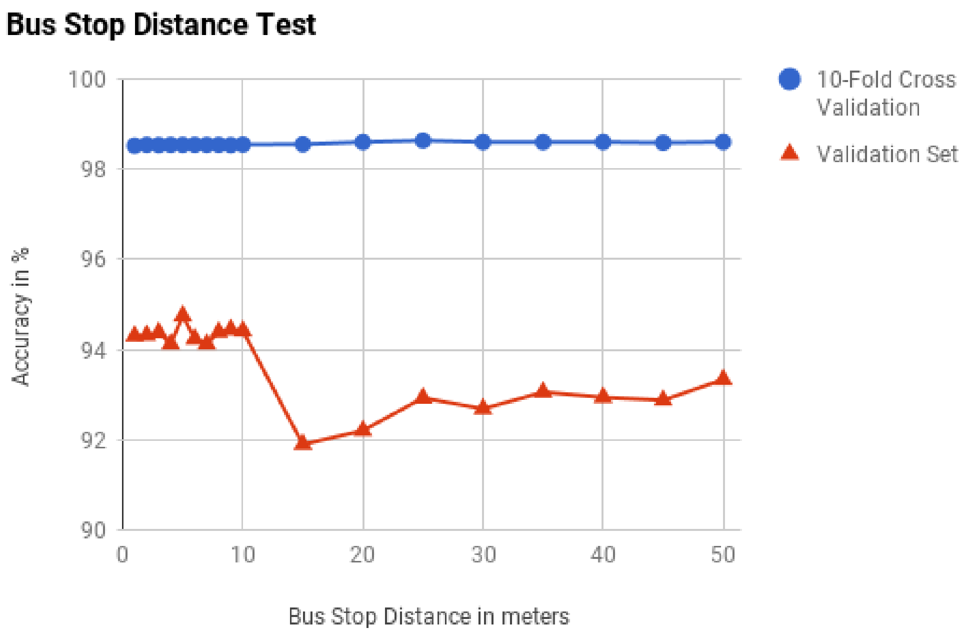


Figure 5.4: Classification accuracy test results for varying nearby bus stop distance.

The main focus of this section will be the combination of different types of datasets used for training the system classifier and measuring their performance to determine which combinations of datasets would benefit most from the new features introduced in Chapter 4. The types of datasets used are GPS based data, accelerometer based data, and bus stop location based data. For every combination test a 10-fold cross-validation with a 50-50 split is executed on T_{train} and the average classification accuracy is recorded each time. At the same time, T_{train} is used to train a classifier C_{train} that is used to classify the set T_{valid} and each time the classification accuracy is recorded for it, too. Table 5.4 shows the combinations of datasets that were tested for this study. The combinations of datasets are not all possible combinations. I only tested and listed combinations that made practical sense and are interesting. For example, it does not make sense to combine "ACC + DTH", because GPS data was not recorded, and DTH features are GPS data dependent. DTH features also dependent on bus stop location data. If GPS was recorded and bus stop location data is available then it makes sense to also calculate GPS, GIS and DTH features to get the most out of the data and maximize classification accuracy, by building a classifier with all features, "GPS + ACC + GIS + WHQ + DTH". This reasoning explains why I did not list and test certain combinations based on data dependencies. Combinations like "GPS + GIS + DTH" and "GPS + GIS + WHQ" were included to demonstrate the strength of DTH and WHQ features.

The test with all features, including the newly developed ones, is test "GPS + ACC + GIS + WHQ + DTH", which means that we turn on GPS based features, accelerometer based features, bus stop location based features, Window History Queue features with bus stop location features, and Dwell Time History features. This test case should ideally yield the best classification accuracy because it uses the most features, but theoretically it is possible that certain combinations of features might not work well together. That is why many combinations of features and datasets are being tested in this study. Maybe a certain subset of case "GPS + ACC + GIS + WHQ + DTH" could yield a performance that is a little worse but is practically good enough.

The results of the dataset combinations test can be found in Table 5.5 and Figure 5.5. Those results show that the test case with the most features indeed yields the highest accuracy of 98.5%.

Table 5.6 presents the confusion matrix for the cross-validation test performed with data from T_{train}. Table 5.7 presents the confusion matrix for the validation test on T_{valid}, which simulates a classification test on new unseen data.

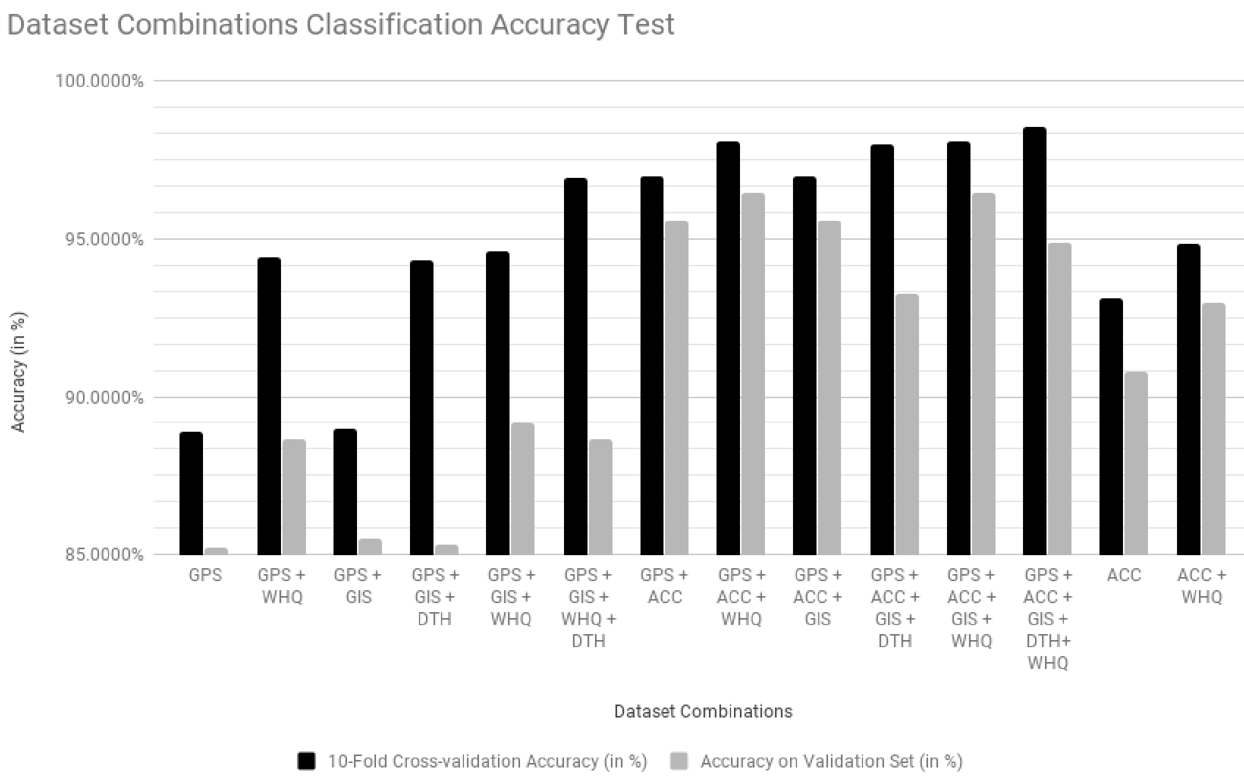


Figure 5.5: Classification accuracy results for tests of different combinations of datasets and features.

Table 5.4: Combinations of dataset types used for testing.

Dataset Combinations	Short Description
GPS	Only GPS based features
GPS + WHQ	GPS based features, and Window History Queue features based on GPS data only
GPS + GIS	GPS based features, and bus stop location based features.
GPS + GIS + DTH	GPS based features, bus stop location based features, and Dwell Time History feature
GPS + GIS + WHQ	GPS based features, bus stop location based features, and Window History Queue features with bus stop location features
GPS + GIS + WHQ + DTH	GPS based features, bus stop location based features, Window History Queue features with bus stop location features, and and Dwell Time History feature
GPS + ACC	GPS based features, and accelerometer based features
GPS + ACC + WHQ	GPS based features, accelerometer based features, and Window History Queue features
GPS + ACC + GIS	GPS based features, accelerometer based features, and bus stop location based features
GPS + ACC + GIS + DTH	GPS based features, accelerometer based features, bus stop location based features, and Dwell Time History feature
GPS + ACC + GIS + WHQ	GPS based features, accelerometer based features, bus stop location based features, and Window History Queue features with bus stop location features
GPS + ACC + GIS + WHQ + DTH	GPS based features, accelerometer based features, bus stop location based features, Window History Queue features with bus stop location features, and Dwell Time History feature
ACC	Only accelerometer based features
ACC + WHQ	Accelerometer based features, and Window History Queue features

Table 5.5: Classification accuracy results for tests of different combinations of datasets and features.

Dataset Combinations	10-fold Cross-validation Accuracy (in % with std. dev.)	Validation Set Accuracy (in %)
GPS	88.8777% (± 0.3958%)	85.2236%
GPS + WHQ	94.4165% (± 0.2494%)	88.6747%
GPS + GIS	88.9631% (± 0.3901%)	85.5005%
GPS + GIS + DTH	94.3122% (± 0.1925%)	85.3221%
GPS + GIS + WHQ	94.6253% (± 0.2322%)	89.1433%
GPS + GIS + WHQ + DTH	96.9116% (± 0.1687%)	88.6507%
GPS + ACC	96.9665% (± 0.1386%)	95.5822%
GPS + ACC + WHQ	98.0918% (± 0.1200%)	96.4610%
GPS + ACC + GIS	96.9739% (± 0.1381%)	95.5556%
GPS + ACC + GIS + DTH	97.9970% (± 0.0864%)	93.2389%
GPS + ACC + GIS + WHQ	98.0815% (± 0.1229%)	96.4504%
GPS + ACC + GIS + DTH + WHQ	**98.5475% (± 0.1370%)**	94.8712%
ACC	93.1341% (± 0.2053%)	90.7943%
ACC + WHQ	94.8517% (± 0.2853%)	92.9779%

Table 5.6: The confusion matrix for the cross-validation test.

Mode of Transportation	Precision	Recall	F1-Score
Walk	97%	98%	98%
Bike	98%	95%	97%
Car	99%	99%	99%
Bus	99%	98%	98%
Average	99%	99%	99%

Table 5.7: The confusion matrix for classification results of the validation test set.

Mode of Transportation	Precision	Recall	F1-Score
Walk	91%	93%	92%
Bike	84%	91%	87%
Car	98%	97%	97%
Bus	91%	91%	91%
Average	95%	95%	95%

5.8 Results Analysis

One goal of this study is to determine whether adding bus stop location based features to a dataset combination with GPS and accelerometer based features would improve the overall transportation mode classification accuracy. The test cases "GPS + ACC" and "GPS + ACC + GIS + DTH+ WHQ" of my results from Table 5.5 confirm that this is true. The results show that using "GPS + ACC + GIS + DTH + WHQ" yields a 1.6% better accuracy than "GPS + ACC", where "GPS + ACC + GIS + DTH + WHQ" yields 98.55% and "GPS + ACC" yields 96.97%. This confirms that adding bus stop location based features to a dataset combination with GPS and accelerometer based features improves transportation mode classification accuracy.

We can further observe the impact of my new GIS based DTH and WHQ features on subsets of "GPS + ACC + GIS + DTH + WHQ". Further examination of Table 5.5 and Figure 5.5 shows, when considering dataset and feature combinations that do not include accelerometer data, the new features have a great effect on accuracy. Let the "GPS" test case be the base case for comparison. A classifier trained with GPS features alone yields an 88.9% accuracy. The test case "GPS + GIS" shows that bus stop location based features have virtually no impact on the classifier's performance. The Dwell Time History (DTH) feature is based on previous dwell times, which is based on the bus stop location based features. If DTH is added to the classifier the classification accuracy improves greatly from 88.9% to 94.31%, which is a 5.43% improvement. We can observe a similar affect when combining Window History Queue (WHQ) feature with GPS based features. "GPS + WHQ" produces an accuracy of 94.42%, which is a 5.54% improvement. When both, WHQ and DTH are combined with GPS based features then accuracy significantly increases further to 96.91% for

“GPS + GIS + WHQ + DTH”, which is another 2.5% improvement, yielding about 8.03% improvement in total when compared to the base case of using only GPS based features. My new features, DTH and WHQ, are features that relate information from previously seen windows to the current window. From the results we can see that those features have great impact on classification accuracy and thus confirming my other research goal, which is to find out whether it is possible to calculate features based on previously seen data for a single trip and use it to improve real-time window based transportation mode classification.

From the results above we can also observe that accelerometer based features are quite strong and that developing a transportation mode classifier based only on GPS and accelerometer based features might be accurate enough for many practical applications, but when bus stop location data is available, then accuracy can even be further improved. The results in Table 5.5 also reveal more valuable information. In a project where accelerometer data is not available, it would make a great impact on performance if a database of bus stop locations could be obtained for the collected GPS data and then it would be possible to build a classifier that is on par with a “GPS + ACC”-classifier.

Accelerometer based features are strong enough that when used alone in a classifier it might deliver a classification accuracy that could be satisfying to the user. In the tests above such a classifier yielded an accuracy of 93.13%. A little boost of 1.72% can be obtained by adding accelerometer based Window History Queue features. An accelerometer based classifier could be useful for applications where GPS data is not needed. Such a classifier could easily provide a good real-time transportation mode classification on mobile devices, depending on the application. Since using GPS logging on mobile devices like smartphones is a big burden on the battery life compared to using accelerometer logging, it would be a good application for such scenarios where available energy is restricted, depending on how energy-consuming the calculations for accelerometer based features are compared to the GPS module energy consumption.

Table 5.6 shows us the confusion matrix for the cross-validation test performed with data from T_{train}. The result of it shows us that overall a trained classifier has a very balanced classification accuracy across transportation modes. We can see that the classifier during cross-validation testing achieved an average of 99% in Precision and Recall, with a F1-Score of 99%, as well.

Table 5.7 shows us the confusion matrix for the validation test on T_{valid}, which simulates a classification test on new unseen data. Here we can see that even when classifying new data the classifier performed reasonably balanced across transportation modes, except for "Biking". In average, the classifier for data of T_{valid} had a 95% Precision, Recall and F1-Score. The lower score for classification of biking data could be explained by the lower amount of data collected for that specific transportation mode. The Random Forest model for classification usually works better when all classes are represented in equal amount in the dataset. Increasing the amount of data for biking could increase the score further.

5.9 Discussion

If we compare my results to some results of similar works by other authors we can see that my classifier achieves a higher accuracy in multiple categories than existing classifiers.

Stenneth *et al.*'s (2011) used GPS data, bus stop locations and real-time bus locations. With that data combination they achieved a classification accuracy of 93.5%. In their test they found that the real-time bus location based features had a much greater contribution to classification accuracy than bus stop location based features. I did not have access to real-time bus locations in my study, hence I could only work with bus stop locations. In one of my tests I trained a classifier with GPS data and bus stop location data. My classifier achieved an accuracy of 96.9%, which is 3.4% higher than Stenneth *et al.*'s (2011) classifier. Presumably, if I could add features based on real-time bus locations, I could produce even better results. If we compare another study result, by Liang *et al.* (2017), which used GPS data and bus stop locations only, to train their classifier, we can see that they had an even lower classification accuracy of 86.5%.

Ellis *et al.* (2014) build a transportation mode classifier based on GPS and accelerometer data and achieved a classification accuracy of 91.9%. As the results above show, we can see that my classifier, trained based on only GPS and accelerometer data, achieved an accuracy of 98.1%, which is 6.2% higher than what Ellis *et al.*'s (2014) classifier was able to achieve.

Hemminki *et al.*'s (2013) work focused on building a classifier based only on accelerometer data. They were able to achieve a classification accuracy of 80.1%. When we compare that to my classifier that was trained only on accelerometer data then we can see that my classifier was able to achieve an accuracy of 94.9%, which is a 14.8% improvement.

The main focus of this study was not about power consumption, but there are some conclusions about it that can be drawn from my experiments. Some rough run-times for calculating features were also recorded for the different dataset combinations. Figure 5.6

presents some approximate times for those tests. We find a striking time difference between cases where accelerometer based features were calculated and cases where accelerometer data was not used. It appears that when using accelerometer based features, run-times increase by about 8 times. This observation might be useful when considering which dataset combination should be used in a project to build a transportation mode classifier. Assuming that the CPU is 100% utilized during features calculations, which it was in my case, then a longer run-time could translate into a higher power consumption. Previously, we have observed that "GPS + ACC"- and "GPS + GIS + DTH+ WHQ"-classifiers have similar practical accuracy performance. If power consumption is of importance in an application, then Figure 5.6 might add some valuable information to that decision. Longer run-times could imply that computational load and power consumption would also increase. This could mean that when choosing to add a classifier that uses accelerometer based features it could also increase power consumption significantly. Since in this study, no power consumption values were tested and recorded, further research of this area would need to be conducted to get accurate number, but at the least we know that accelerometer based features have a significant impact on run-time and presumably power consumption, as well, which could be taken into consideration when developing mobile applications.

5.10 Summary

In this chapter, I have tested my transportation mode classification system. I have used GPS, accelerometer data and bus stop locations that I have collected with smartphone applications, in Winnipeg. I used the data to train a Random Forest machine learning model classifier that is able to classify four different transportation modes, namely, "walk",

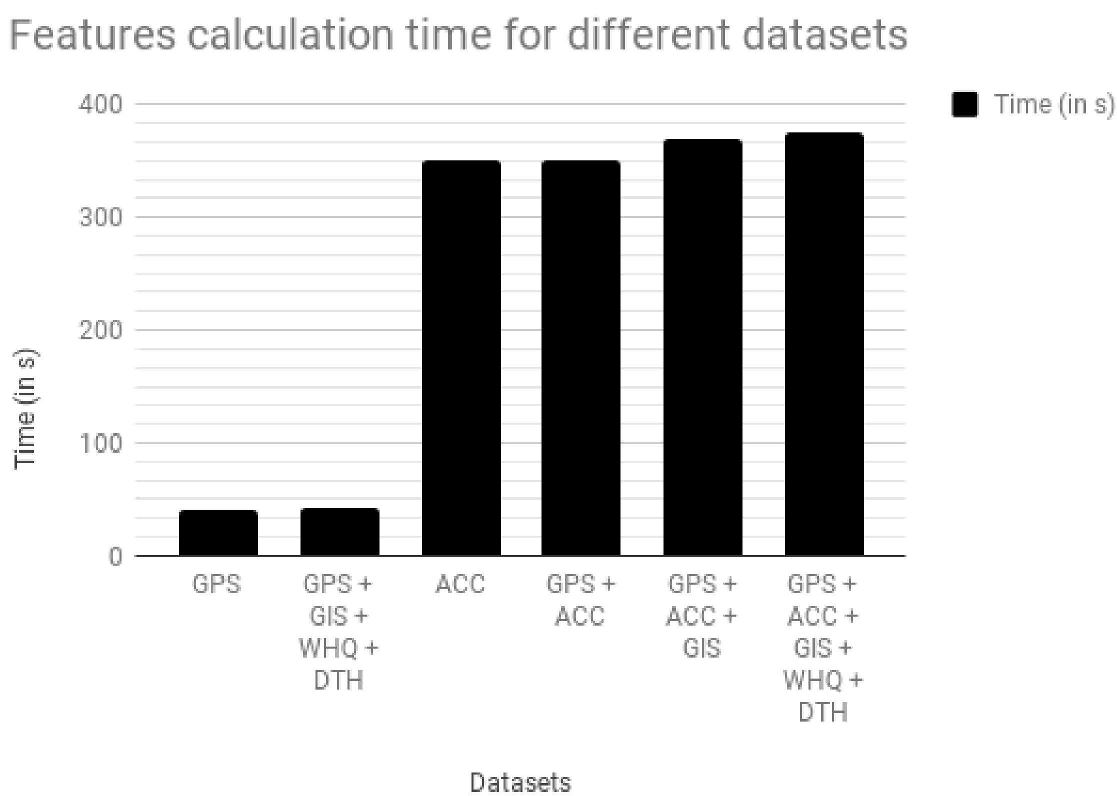


Figure 5.6: These are run times for calculating features for different dataset combinations.

"bike", "car", and "bus". In the experiments, I have figured out a good window segmentation size (4 seconds), a good bus stop distance (5 meters), and a good Window History Queue length (15 windows), based on my dataset. I found the small window size of 4 seconds to be best for my dataset. Such a small window size has the advantage that it lets us perform transportation mode classification at a real-time pace, which could make it practical for applications that need classification capabilities "on-the-go". Using the aforementioned parameter values, I have tested my new classification features used in the classifier training, and tested the features' classification performance. My newly developed features are the "Window History Queue", and the "Dwell Time History" features. Stratified 10-fold cross-validation tests yielded a maximum classification accuracy of 98.55% for the classifier that uses all my different data types and all features, GPS based features, accelerometer based features, bus stop location based features, Window History Queue features with bus stop location features, and Dwell Time History feature.

The classification accuracy is on par with classifiers from existing works. It is difficult to compare raw performance directly with other works because that would require having access to the same datasets and replicating existing studies first before one could add more improvements. Unfortunately, in the field of transportation mode classification, there is no open benchmark dataset that researchers use and share. I did my best to use commonly used features and techniques to make comparison easier. To determine the accuracy performance impact of my new features I trained multiple classifiers with different types of data and features to compare them. This way, I was able to compile a very useful table of different classifier performance results that can be used to make better decisions about future development of transportation mode classifiers. In this chapter, I compared the accuracy of the my different classifiers to comparable existing works and determined that my classifiers' accuracies were significantly higher. The results of the dataset combinations test can be found in Table 5.5 and Figure 5.5. Those results also show that the test case with the most features indeed yields the highest accuracy of 98.55%. From my experiments, we can also observe that my new features, "Window History Queue" and "Dwell Time History", which are features that relate information from previously seen windows to the current window, have a great impact on classification accuracy.

Another outcome of my experiments was that I could make some conclusions about power consumption by looking at the run times of my feature calculations. By looking at Figure 5.6, which shows run times for calculating features for different dataset combinations, we find a striking time difference between cases where accelerometer based features were calculated and cases where accelerometer data was not used. Longer run-times could imply that computational load and power consumption would also increase. Such knowledge could be useful when developing a classifier, where power consumption needs to be taken into consideration.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The goal of my thesis was to find out whether bus stop location data, in addition to GPS and accelerometer data, could be used to improve transportation mode classification. In my system, I used a windowed segmentation approach that allows for both real-time and non-real-time classification. Existing works of transportation mode classification that take a similar real-time classification approach use many kinds of sensor data, like GPS, accelerometer, *etc.*, but as far as I could determine, there are not many studies that used bus stop locations for calculating machine learning features for their classification systems. To the best of my knowledge we are the first to uniquely use GPS, accelerometer data and bus stop locations in one transportation mode classification system that uses windowed segmentation of trip data. I have developed new features based on bus stop locations, described in Chapter 4, that could be used in a windowed segmentation approach of transportation mode classification. The outcome of my tests in Chapter 5 show that the newly developed features based on bus stop locations, i.e., GIS based DTH and WHQ features, further improved existing work by providing a high classification accuracy of 98.5%. A transportation mode classifier that uses only GPS and accelerometer based features achieved 97.0%. Existing work (Su et al., 2015) that is similar to mine achieved an accuracy of 96.4%, with a classifier that used many smartphone sensors. The impact of my new features are probably significant enough for future exploration. With further tweaking of parameters and improvement of the new features I suspect that a classifier with better accuracy performance could be trained.

I believe my study has great value for future research and application development. The results of Table 5.5 show classification accuracies of different transportation mode classifiers that have been trained with different combination of datasets and features. This can be used to make better decisions when picking data for training a classifier. For example, when compared to the base case of using only GPS based features, which has a 88.9% accuracy, then “GPS + GIS + WHQ + DTH”, which has a 96.9%, yields a 8% improvement in total.

The results of my study suggest that accelerometer based features alone are quite strong and that developing a transportation mode classifier based only on GPS and accelerometer based features might be accurate enough for practical applications. Accelerometer based features are strong enough that when used alone in a classifier it might deliver a classification accuracy that could be satisfying to the user. Figure 5.5 shows such a classifier yielded an accuracy of 93.13%. 94.85% can be obtained by adding accelerometer based Window History Queue features.

The confusion matrices in Table 5.6 and Table 5.7 indicate a lower classification accuracy score for classification of biking data. This could be explained by the lower amount of data collected for that specific transportation mode. The Random Forest model for classification usually works better when all classes are represented in equal amount in the dataset. Increasing the amount of data for biking could increase the score further.

An accelerometer data based classifier for offline on-device classification might consume less power and could be more accurate than using only GPS data for such a classifier. From the run-times chart in Figure 5.6 we know that accelerometer based features have a significant impact on run-time. Run-time increases by at least 8 times and presumably power consumption, as well, which could be taken into consideration when developing mobile applications for smartphones or other portable devices. Without further experiments we cannot definitely conclude whether transportation mode classification systems based on accelerometer data consume significantly more power than classification system based on other data, but my run-time chart in Figure 5.6 gives us at least some clues that it is likely that an accelerometer data based system could consume more power. If a GPS module would be used in a system and only gets used with a very low sampling frequency then a GPS data based classification system might be more power efficient. More work would need to be done to figure out the exact sampling rates that would allow a power efficient data collection that yields a transportation classification accuracy that is satisfying for the application and use case.

When I compare my classifier that uses only GPS and GIS data, it performs better than a classifiers that processes data after the entire trip data collection. My classifier showed an accuracy of 96.9%, compared to Biljecki *et al.'s* (2013) classifier that worked with similar data as mine and showed a 91.6% accuracy. As mentioned before, Biljecki *et al.'s* (2013) solution and mine are different. They used a post-analysis approach, whereas I used a real-time strategy. My approach seems to be simpler to implement and did not require much data cleanup and preparation. My classifier could also be used to classify trips after data collection is finished.

In my thesis, I have focused mainly on the feature and classifier development without focusing on data cleanup, such as, GPS trajectory smoothing and accelerometer data reorientation, and only performed minimal data preparation. Even with such raw data, I was able to develop a transportation mode classifier that was very accurate. I even mixed different accelerometer datasets that were sampled at 10 Hz and 22 Hz. Despite that, my system performed well and demonstrated robustness in the presence of noisy data.

Initially, in Section 1.1, I stated the goal of my thesis with two questions, **Q1** & **Q2**. I believe I was able to successfully answer the research questions:

Q1: Does adding bus stop location data based features to a feature set based on accelerometer and GPS data improve real-time window based transportation mode classification?

A1: I successfully solved the first problem by developing *Dwell Time* based features around bus stop locations and used a *Dwell Time History* (Section 4.9) per trip to reduce ambiguity between public and private transportation mode.

Q2: Is it possible to calculate features based on previously seen data for a single trip and use it to improve real-time window based transportation mode classification?

A2: I solved the second problem by developing a *Window History Queue* (Section 4.10) feature that allows to incorporate knowledge from previously seen data windows into the currently evaluated window.

Both solutions together further improved existing transportation mode classification work by providing a high classification accuracy of 98.5%. I further looked into the impact of my new features for GPS only classifiers and concluded that, by adding them to such a classifier, they improve classification by about 8%, from 88.9% to 96.9%. I was also able to show more interesting conclusions of my experiments that have value for further research and classifier development, which I described above in this section.

6.2 Future Work

Although I successfully developed a transportation mode classification system with a good classification accuracy for real-time applications, there are limitations that could be addressed in future studies. Parameters, such as, nearby bus stop distance threshold, feature window size, and window history queue size might have a different optimal value for different cities and countries or even different regions within a city. This could be further explored if more data would be available. In the future, I need to collect more varied data and tweak the parameters to develop a transportation mode classifier that could work robustly in many major cities. Access to GIS data, such as, bus stop locations could also vary for different cities and especially small towns. In order to ensure that a classifier could work in different regions it would be good to develop a classifier that could dynamically account for missing data and still reliably classify data. This could be achieved by training different models that use different combinations of datasets, such as, "GPS" + "ACC", "GPS" + "GIS", "GPS" + "ACC" + "GIS", *etc.*

Some machine learning models could be better at classifying particular transportation modes than others. Xiao *et al.* (2017) implemented an ensemble classifier based on tree-based classifiers. Although, their results show no significant improvement it could be worthwhile to experiment with different combinations of different categories of classifiers to find a significant improvement. I would like to research impacts of ensemble classifiers for transportation mode classification, as well.

Since travel log data is of sequential nature it should be possible to further explore how relationships of data points within a trip can be processed to improve classification accuracy. *Probabilistic Graphical Models (PGM)* are often used for training classifiers for sequential data, and could be worth exploring. I think PGMs could be used as a secondary classification step on trip leg classification. I would like to explore how trip windows could be clustered into appropriately sized trip legs in order to be better suited for PGM training and classification.

In the future, I would like to extend my system to detect exact bus routes that are taken when traveling by bus. This would be very useful for automated public transit surveys. To achieve that I could use bus route shapefiles provided by Winnipeg Transit (Winnipeg Transit, 2018), origin and destination bus stops (Winnipeg Transit, 2018), road maps to match routes (OpenStreetMap, 2018a,b) (note: most major cities provide these kind of public transit information in General Transit Feed Specification (GTFS) format (Google Inc., 2018b) via Google Maps Transit (Google Inc., 2018c)). Stenneth *et al.* (2011) did public transit route detection but they did not reveal how exactly they achieved it in a real-time system with data window segmentation.

I would also like to implement a pre-processing stage and a post-processing stage to explore potential classification accuracy gains. The pre-processing stage would consist of data preparation, like clean up outliers and smoothing GPS trajectories. Accelerometer data could also be augmented with gravity vector data that could be calculated by averaging vectors for a given frame, which would make it possible to calculate more accurate movement patters along the vertical axis. The data pre-processing would essentially reduce the noise in the data and hopefully improve classification accuracy. The post-processing stage would consist of refining classified widows of data from the machine learning classifier. For example, if we have 10 short windows of data where 8 of 10 windows were classified as "bus" mode and the other 2 windows are classified with different modes, like "car" or "bike", then it is likely that those 2 windows are misclassifications and should be labeled as "bus", because it is unlikely that a person would switch such transportation modes in such a short time.

My study was limited to data collected only by four people in Winnipeg. More people had the application I developed on their phones but unfortunately I did not receive any data for more than four people. I suspect that it was too inconvenient to use the application and people tend to forget to start and end trips at the correct times, which was important to build a well labeled dataset. For future studies I would like to make the mobile application easier and more convenient to use. The current dataset allowed me to build a classifier that I could use to improve data collection. This could be achieved by asking users of the application to label the trips after recording at a time that is more convenient for them. A classifier in the smartphone application could classify the trips and present the user with the results and they could confirm or correct mislabeled sections of their trips on a map before uploading the data to the server. The application could also run continuously in the background and collect data without user interaction. At the end of the day the user would be reminded to review daily trips. The data for a day could be automatically segmented into trips by finding very long dwell-times as a cut-off. Furthermore, the application could be optimized to detect movement with a set minimum threshold and only collect data when a person is commuting, effectively reducing power consumption and burden on the user. With such an improved application it could be much more likely that people would use the application, and more people could contribute to the total dataset to increase city coverage and variety of data.

For future studies I would also like to include more transportation modes beyond the four modes (walk, bike, bus, car) that I used in my thesis. In order to collect more data from other cities with more public transit options available (*e.g.*, metro, train, streetcar, *etc.*), it could be worthwhile to extend the data collection application with features beneficial for users, such as, speed limit and school zone warnings. The application could be offered on the Google Play Store and the Apple App Store for free, which would allow for easy distribution and the potential of receiving that from many different regions.

As mentioned earlier in my thesis, exploring power consumption of different transportation mode classifiers would also be an interesting research extension. I would like to figure out, which combination of datasets are sufficient for building a classifier that is both very accurate at classifying transportation modes and power efficient during run-time. For that, I could also look at different data features, sampling rates, hardware parameters and offline- vs online-classification systems to determine minimal feasible power consumption. For such a study, I could also go beyond the use of smartphones and explore a hybrid solution with smartphones and dedicated logging devices with at least GPS and accelerometer modules, to explore advantages and disadvantages for transportation mode classification and maybe even other applications, such as, traffic analysis, navigation, alert systems, *etc.*

Bibliography

Andrienko, G., Gunopulos, D., Ioannidis, Y., Kalogeraki, V., Katakis, I., Morik, K., & Verscheure, O. (2016). Mining urban data (part b). *Information Systems*, *57*, 75–76.

Andrienko, G., Gunopulos, D., Ioannidis, Y., Kalogeraki, V., Katakis, I., Morik, K., & Verscheure, O. (2017). Mining urban data (part c). *Information Systems*, *64*, 219–220.

Arentze, T., Dijst, M., Dugundji, E., Joh, C.-H., Kapoen, L., Krygsman, S., Maat, K., & Timmermans, H. (2001). New activity diary format: Design and limited empirical evidence. *Transportation Research Record: Journal of the Transportation Research Board*, *1768*, 79–88.

Behnisch, M. & Ultsch, A. (2008). Urban data mining using emergent SOM. In *Data analysis, machine learning and applications* (pp. 311–318). Springer.

Biljecki, F., Ledoux, H., & Van Oosterom, P. (2013). Transportation mode-based segmentation and classification of movement trajectories. *International Journal of Geographical Information Science*, *27*(2), 385–407.

Blair, D. C. (1979). *Information Retrieval, CJ Van Rijsbergen. London: Butterworths; 1979: 208 pp.* Wiley Online Library.

Bohte, W. & Maat, K. (2009). Deriving and validating trip purposes and travel modes for multi-day GPS-based travel surveys: A large-scale application in the netherlands. *Transportation Research Part C: Emerging Technologies*, *17*(3), 285–297.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Cajori, F. (2007). *A History of Mathematical Notations: Vol. II*. A History of Mathematical Notations. Cosimo, Incorporated.

Chen, C., Gong, H., Lawson, C., & Bialostozky, E. (2010). Evaluating the feasibility of a passive travel survey collection in a complex urban environment: Lessons learned from the new york city case study. *Transportation Research Part A: Policy and Practice*, *44*(10), 830–840.

Chou, Y.-l. (1969). Statistical analysis, holt, rinehart and winston. *Inc., New York*, *794*.

Chung, E.-H. & Shalaby, A. (2005). A trip reconstruction tool for GPS-based personal travel surveys. *Transportation Planning and Technology*, *28*(5), 381–401.

Cvetković, B., Janko, V., & Luštrek, M. (2015). Activity recognition and human energy expenditure estimation with a smartphone. *Proceedings of IEEE Pervasive Computing and Communication (PerCom)*, 193–195.

Draijer, G., Kalfs, N., & Perdok, J. (2000). Global positioning system as data collection method for travel research. *Transportation Research Record: Journal of the Transportation Research Board*, (1719), 147–153.

Ellis, K., Godbole, S., Marshall, S., Lanckriet, G., Staudenmayer, J., & Kerr, J. (2014). Identifying active travel behaviors in challenging environments using GPS, accelerometers, and machine learning algorithms. *Frontiers in Public Health*, *2*, 36.

Ettema, D., Timmermans, H., & van Veghel, L. (1996). Effects of data collection methods in travel and activity research.

Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996). A definition of knowledge discovery in databases. In U. M. Fayyad, G. Piatetsky-Shapiro, P. SmytDedh, & R. Uthurusamy (Eds.), *Advances in knowledge discovery and data mining* chapter 1.2, (pp. 6–9). Palo Alto: American Association for Artificial Intelligence (AAAI) Press.

Forrest, T. & Pearson, D. (2005). Comparison of trip determination methods in household travel surveys enhanced by a global positioning system. *Transportation Research Record: Journal of the Transportation Research Board*, *1917*, 63–71.

Geisser, S. (2017). *Predictive Inference*. CRC Press.

Gleave, S. D. & GeoStats, L. (2003). The use of gps to improve travel data. study report.

Google Inc. (2018a). `https://www.google.com/maps/timeline`.

Google Inc. (2018b). `https://developers.google.com/transit/gtfs/`.

Google Inc. (2018c). `http://maps.google.com/landing/transit/index.html`.

Greenfeld, J. (2002). Matching GPS observations to locations on a digital map.

Hemminki, S., Nurmi, P., & Tarkoma, S. (2013). Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, (pp. 13:1–13:14)., New York, NY, USA. ACM.

Liang, J., Zhu, Q., Zhu, M., Li, M., Li, X., Wang, J., You, S., & Zhang, Y. (2017). *An Enhanced Transportation Mode Detection Method Based on GPS Data*, (pp. 605–620). Singapore: Springer Singapore.

Luštrek, M., Cvetković, B., & Janko, V. (2014). Monitoring patients lifestyle with a smartphone and other devices placed freely on the body. In *Ambient Intelligence* (pp. 96–99). Springer.

Maat, K., Timmermans, H. J., & Molin, E. (2004). A model of spatial structure, activity participation and travel behavior. In *10th World Conference on Transport ResearchWorld Conference on Transport Research SocietyIstanbul Technical University*.

McGowen, P. & McNally, M. (2007). Evaluating the potential to predict activity types from GPS and GIS data. In *Transportation Research Board 86th Annual Meeting, Washington*. Citeseer.

Mizell, D. (2003). Using gravity to estimate accelerometer orientation. In *Proc. 7th IEEE Int. Symposium on Wearable Computers (ISWC 2003)*, volume 252. Citeseer.

Murakami, E., Wagner, D. P., & Neumeister, D. M. (2004). Using global positioning systems and personal digital assistants for personal travel surveys in the united states. In *International Conference on Transport Survey Quality and Innovation*.

Ohmori, N., Nakazato, M., & Harata, N. (2005). Gps mobile phone-based activity diary survey. In *Proceedings of the Eastern Asia Society for Transportation Studies*, volume 5, (pp. 1104–1115).

OpenStreetMap (2018a). `http://wiki.openstreetmap.org/wiki/API`.

OpenStreetMap (2018b). `https://wiki.openstreetmap.org/wiki/Key/:highway`.

PERSENTECH (Persen Technologies Inc.) (2018). `http://persentech.com`.

Powers, D. (2007). Evaluation: from precision, recall and f-factor to roc, informedness, markedness & correlation (technical report). *Adelaide, Australia*.

Schlich, R. & Axhausen, K. W. (2003). Habitual travel behaviour: evidence from a six-week travel diary. *Transportation*, *30*(1), 13–36.

Schönfelder, S., Axhausen, K. W., Antille, N., & Bierlaire, M. (2002). Exploring the potentials of automatically collected gps data for travel behaviour analysis. *Arbeitsberichte Verkehrs-und Raumplanung*, *124*.

Shafique, M. A. & Hato, E. (2015). Use of acceleration data for transportation mode prediction. *Transportation*, *42*(1), 163–188.

Shafique, M. A. & Hato, E. (2016). Travel mode detection with varying smartphone data collection frequencies. *Sensors*, *16*(5).

Sokmenoglu, A., Cagdas, G., & Sarıyıldız, S. (2010). Exploring the patterns and relationships of urban attributes by data mining. In *Proceedings of the 28th Education and Research in Computer Aided Architectural Design in Europe (eCAADe) Conference, Zurich, Switzerland*, (pp. 873–881).

Stenneth, L., Wolfson, O., Yu, P. S., & Xu, B. (2011). Transportation mode detection using mobile phones and GIS information. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, (pp. 54–63). ACM.

Stopher, P., Clifford, E., Zhang, J., & FitzGerald, C. (2008). Deducing mode and purpose from GPS data. *Institute of Transport and Logistics Studies*, 1–13.

Stopher, P. & Wilmot, C. (2000). Some new approaches to designing household travel surveys–time-use diaries and gps. In *79th Annual Meeting of the Transportation Research Board*.

Stopher, P. R. (1992). Use of an activity-based diary to collect household travel data. *Transportation*, *19*(2), 159–176.

Su, H. X., Caceres, H., & He, Q. (2015). Travel mode identification with smartphones. *Sensors*, *15*, 16.

Su, X., Caceres, H., Tong, H., & He, Q. (2016). Online travel mode identification using smartphones with battery saving considerations. *IEEE Transactions on Intelligent Transportation Systems*, *17*(10), 2921–2934.

Tundo, M. D., Lemaire, E., & Baddour, N. (2013). Correcting smartphone orientation for accelerometer-based analysis. In *2013 IEEE International Symposium on Medical Measurements and Applications Proceedings (MeMeA)*, (pp. 58–62). IEEE.

Winnipeg Transit (2018). Winnipeg transit open data web service. `http://api.winnipegtransit.com`.

Wolf, J., Guensler, R., & Bachman, W. (2001). Elimination of the travel diary: Experiment to derive trip purpose from global positioning system travel data. *Transportation Research Record: Journal of the Transportation Research Board*, *1768*, 125–134.

Xiao, Z., Wang, Y., Fu, K., & Wu, F. (2017). Identifying different transportation modes from trajectory data using tree-based ensemble classifiers. *ISPRS International Journal of Geo-Information*, *6*(2).

Zheng, Y., Chen, Y., Li, Q., Xie, X., & Ma, W.-Y. (2010). Understanding transportation modes based on GPS data for web applications. *ACM Trans. Web*, *4*(1), 1:1–1:36.

Zheng, Y., Li, Q., Chen, Y., Xie, X., & Ma, W.-Y. (2008). Understanding mobility based on GPS data. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, (pp. 312–321)., New York, NY, USA. ACM.