University of Zurich[UZH]

Department of Geography

Transport mode detection using Cellular Signaling Data (Case study of Graz and Vienna, Austria)

GEO 620 Master's Thesis

Author
Kimberley Chin Jiaqi
16-721-185

Supervised by
Dr Haosheng Huang, University of Zurich
Christopher Horn, Invenium Data Insights
Dr. Ivan Kasanicky, Parkbob

Faculty representative
Prof. Dr.Robert Weibel

29.06.2018
Department of Geography, University of Zurich

ACKNOWLEDGEMENTS

As I embark on my final leg of my studies at the University of Zurich, I would like to express my heartfelt thanks to everyone who has taught, helped and supported me throughout my studies and master's thesis. This journey would have been a lot harder without:

- My supervisor Dr Haosheng Huang. Thank you for all the support, ideas and time you have given me during this past year. Without your critical input and constant guidance this would not have been possible.
- Co-supervisors Christopher Horn of Invenium Data Insights and Dr. Ivan Kasanicky of Parkbob. Many thanks for giving me the opportunity to work on a topic in such an exciting field, and all your valuable insights and suggestions. Special thanks goes to to Christopher for providing the data on their own as well as from Alexandra Lechner of TU Graz.
- My dear friends and family and especially Nick, for proofreading and more importantly, for the much needed emotional support and encouragement you have given in more ways than one.

Contact

Author
Kimberley Chin
Rousseaustrasse 61,
8037 Zürich, Switzerland
kimberleychin@hotmail.com

Supervisor
Dr Haosheng Huang
Geographic Information Science (GIS)
Department of Geography
University of Zurich
Winterthurerstrasse 190
8057 Zürich, Switzerland
haosheng.huang@geo.uzh.ch

Co-Supervisor
Christopher Horn
Science Tower
Waagner-Biro-Strasse 100/11
8020 Graz, Austria
christopher.horn@invenium.io

Co-Supervisor
Dr Ivan Kanasicky
Parkbob Gmbh
Treustrasse 22-24
1200 Vienna, Austria
ivan.kanasicky@parkbob.com

ABSTRACT

The rise of new Big Data sources such as cellular network data has allowed us to observe and comprehend human behavior and the interactions between them and the environment on a much deeper level. This leads to both new research opportunities as well as challenges. Transport mode detection plays a key role directly or indirectly in many fields such as urban planning, epidemiology, transportation science and many more. Improving travel demand surveys is an important driving factor and motivation in this research. Aspects like scalability of these alternatives are critical considerations in their development in terms of data collection and processing. Researchers have looked to Global Positioning Systems (GPS) in the form of loggers or GPS-enabled mobile phones, as well as Call Detail Records (CDR) as alternatives. While these methods have shown promising results, they are not without flaws.

The aim of this research is thus to design a methodology that can detect modes of transportation from another more unknown type of data, cellular signaling data. Cellular signaling data does not require overhead as it can be described as data crumbs leftover by everyday usage of one's cellphone. Based on the results, we can present a deeper understanding into the data characteristics and its potential in understanding human mobility flows in cities. This research will present a set of supervised and unsupervised methods that are applied to data that is collected in Vienna and Graz (Austria) in two separate data collection campaigns by a group of 2 and 9 participants. The results from the proposed methods show promise and are comparable to existing GPS studies of the same aim. The best performing method, a hybrid method of rule-based heuristics and supervised random forest managed to correctly distinguish between U-Bahns, S-Bahns, cars, bikes and walk modes 73% of the time. Rule-based methods perform especially well on rail modes (U-Bahn/S-Bahn). For the more similar modes (cars, bikes for example), random forest does the best at distinguishing between these modes. While unsupervised methods are not able to achieve the same accuracies, the results are still comparable with a 68% accuracy achieved with the partitioning-around-medoid technique.

Keywords: Transport mode detection, cellular signaling data, fuzzy logic, rule-based heuristic, random forest, unsupervised clustering, principal component analysis

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
Introduction

1.1 Context and Motivation

> "Research in human movement in time and space has been around for at least over five decades" - (Weiner, 1986).

Research in human movement has been given a huge helping hand with the rise of new Big Data sources such as mobile phone call detail records or social media records with location tags. We now have the ability to observe and comprehend human behavior and how they interact with their environment on an unprecedented level of detail. This leads to both new research opportunities as well as challenges. Such location-based data can give us valuable insights to human movement in both time and space once new techniques are developed to harness this potential (Zook et al., 2015). The global spread of mobile technologies for communication has brought the world closer together whilst also resulting in the existence of an unparalleled data source capable of describing diverse dealings in the world of human and social behavior. One example of this is the Call Detail Records, the byproduct of billing services for calls, which include timestamps and location coordinates of these transmissions. These widespread datasets can reveal compelling information on patterns on an individual as well as collective scale (Blondel et al., 2015).This is a passive data type which is usually by-products of existing structures that were generated for purposes that were not for but could potentially be used for research (Chen et al., 2016). Other passive data types include social media data that have been posted voluntarily by online users (Gonzalez et al., 2008) and transit card data used in public transport systems (Hasan et al., 2013; Liu et al., 2009).

One important group of benefactors of this data is those in the field of transport science. Urban planners, policy makers and transport management are interested in how people travel, how infrastructure and the environment affect movement, and of course, in obtaining a realistic picture of travel demand. In doing so, many other fields can benefit from it. In this vein, these mobile phone traces have been used to further several aims such as estimating human mobility patterns and population distribution (Calabrese et al., 2015; Gonzalez et al., 2008, 2010; Sevtsuk and Ratti, 2010;Reades et al., 2007), analysis urban activities (Jiang et al., 2013; Ricciato et al., 2017; Widhalm et al., 2015), generating Origin-destination flows (Calabrese et al., 2011; Horn et al., 2017; Kalatian and Shafahi, 2016; Tettamanti et al., 2012; Wang et al., 2010), and many more. All these pursuits are of great interest to the field of transportation science and planners are looking into how best to yield this information for cities' transportation systems. Singapore, a rapidly growing, densely populated metropoliton city is using decade long demand forecasts on their public transportion, among other planning sectors such as land use and urban redevelopemnt planning. Their SmartMobility2030[1] initiative is leading the way in incorporating such location data to plan the nations transit needs and is continueing to grow in this area.

When we have a better understanding of human and traffic flows, we gain greater insights and management capabilities for traffic congestion, health monitoring, elderly care and even epidemiology. One way this can be done is by understanding the transport mode choices of people and this is achieved through collecting information through travel demand surveys or travel diaries. Improving travel demand surveys is an important driving factor and motivation in this research. Aspects like scalability of these alternatives are critical considerations in their development in terms of data collection and processing. For example, traditional surveys may take the form of manual collection and labeling, like telephone interviews and questionnaires. Inaccuracies are introduced as a result. Researchers have looked to Global Positioning Systems (GPS) in the form of loggers or GPS-enabled mobile phones, as well as CDRs as alternatives. Another motivation is context-aware location-based services. Transportation modes such as walking, cycling or train denotes certain characteristics of a user. One use of this knowledge is targeted and customized advertisements that may be deployed to the relevant markets. As people have began to see the large potential of these datasets, and while our telecommunications infrastructure has improved leaps and bounds, so have the quality of the cellular network data that comes with it. A step up from the usual CDR data is Cellular Signaling Data (CSD). CSD consists of not some normal CDRs, but other cellular network related data including both network and event-driven data (section 3.4.2). With an additional map-matching step, CSD ultimately lends itself an increased spatial and temporal resolution. Similar to ad targeting strategies of some social media platforms, there are potential avenues to generate

[1] https://www.lta.gov.sg/content/dam/ltaweb/corp/RoadsMotoring/files/SmartMobility2030.pdf

more revenue with more targeted and customized services. Telecommunication providers have taken notice and begun to invest in developing techniques to mine information and provide access to this data. As such, this thesis will attempt to achieve the aims of transport mode detection with CSD.

This research is in collaboration with Parkbob, a rapidly growing company that delivers context-aware parking information to drivers as well as using predictive models with real-time, crowd sensed data to provide parking availability information. While this is in the realm of LBS, the motivation for this is largely driven by the desire to understand the transport demand that drives the need for this LBS, and is hence lies more in the vein of transportation science.

1.2 Problem statement and research aims

With the aims of transport mode detection in mind, CSD offer a much more opportunities in terms of types of modes and performance due to its higher quality. However, despite this passive data type having the advantages of large sample size and long observation periods, they also have obvious weaknesses: cell phone traces can be sparsely sampled in time during idle periods, they might provide only a low spatial resolution and include noise stemming from pure signal movement. Therefore the data has to be carefully processed to extract trip origins and destinations. Access is also hindered due to varying privacy and business-sensitivity considerations. Many of the previous studies involving cellular network data for mobility analysis have been limited to CDRs and have come up with methods to alleviate the impact of these challenges (Qu et al., 2015; Wang et al., 2010). It was only recently that companies have allowed access to this new cellular signaling data whose greater detail means much more information can be mined as compared to CDR. The main challenge here however, is handling the still much lower spatial and temporal resolutions associated with this passive data type without having to actively solicit other supplementary data in a time and resource intensive manner. Privacy concerns also mean that existing studies do not have ground truth data to evaluate their results. Because of this, the number of modes that have been distinguished using passive mobile phone data has been rather limited, usually to between motorized and non-motorized, or private and public transportation modes.

As such, the aim of this paper is to overcome the restrictions of active data types (GPS) and passive data types (CDR-only) for mode detection and propose novel methods for this relative newcomer, CSD, while also accounting for its low and irregular spatial and temporal resolution. The study area consists of two major cities in Austria, Vienna and Graz. The modes of transport of interest will be both private and public transportation modes: car, bike, walk, tram, S-Bahn (commuter trains) and U-Bahn (metro). This is also limited by the amount of actual data made available to this study. Methods that are taken from existing CDR and GPS studies, whether in part or in whole, will be adjusted so that they are more suitable to deal with the unique data characteristics of CSD. This thesis will consist of two main parts. First, several methods will be developed using combined approaches of several popular mode detection methods proposed by several existing studies. This will include both scenarios whereby labels are available and the more likely ones whereby they are not. The second part will evaluate the performances of these proposed approaches, and compare them based on various performance metrics.

> ***Research Question 1:*** *Development and Implementation: How can various modes of transportation (walk, bus, tram, car) be detected from cellular signaling data (CSD) considering its lower and more irregular spatial and temporal resolution?*
>
> ***Hypothesis 1:*** *Due to the unique characteristics of this dataset, tailoring existing methods to be applied here can detect various modes of transportation. This is possible by distinguishing between their spatiotemporal characteristics as well as complementary common sense information such as locations of transport networks. Supervised mode detection methods developed here would follow a combination of rule-based heuristics and fuzzy logic systems or machine learning. Unsupervised mode detection methods would follow a clustering approach combined with unsupervised random forests. These methods will use variables selected through various variable selection measures.*

Following the implementation of this developed methods the next question addresses their performance and quality and determines the best method that should be adopted for mode detection in urban areas for these particular modes of interest. Using various performance metrics, the results of these proposed methods will be compared against each other as well as against those in existing studies to give an idea of how well the chosen method performs.

Research Question 2: *Evaluation and comparison: How do these proposed methods (RQ1) perform and compare against each other? Which is the best method of mode detection for detecting these modes of transportation? What are the most useful features for transport mode detection using CSD?*

Hypothesis 2: *The results of these algorithms will be compared against ground truth provided by the data collectors' annotations. Due to the noisier and dirtier characteristics of CSD as compared to GPS data, it is likely that the inclusion of contextual data such as GIS data of the transportation network to supplement the CSD will lead to better performances of the methods.*

1.3 Main expected outcomes

The main contribution of this thesis is thus a methodology to detect modes of transportation from CSD data based on their spatial and temporal features. A set of methods that are permutations of various existing approaches will be proposed with the goal of finding the one best suited to CSD after evaluating their results. This will be a novel contribution to the field as transport mode detection using CSD is still very much in its infancy.

1.4 Thesis structure

A summary of related works will be presented in the next chapter, chapter 2. It will also highlight the pros and cons of existing mode detection methods and how lessons learned from them are used in the development and creation of our proposed methods. Following this, chapter 3 will give an overview of the data and chapter 4, the methods proposed for mode detection in this study. The evaluation results will be presented in chapter 5, along with sensitivity analyses of the chosen parameters. Upon discussion of these various methods in Chapter 6, the method (/s) deemed as best and most suited for CSD will be recommended. Both research questions as well as limitations of the study will also be discussed in this chapter. Lastly, Chapter 7 concludes the study with a summary as well as considerations for future work.

CHAPTER 2
BACKGROUND AND RELATED WORK

2.1 The world of transportation

Recent decades of massive population growth combined with the huge influx of urban migration has called for the need to manage our urban resources in a more effective way to streamline already limited resources. Transportation is one of these key issues that growing populations have to grapple with, as it is an unavoidable aspect of everyday life. Motivated by the need to better serve society, cities need to be able to forecast future travel demand so as to channel the right investments in the right volumes to the right places, such as to large-scale transportation projects. Much effort has gone into seeking to develop models that predict where and when people travel to, how they do it, and what affects these choices. Information like how transport networks perform in terms of congestion and flow, or how route and mode choices respond to road pricing schemes are extremely important in aiding dense cities keep up with the increasing pressures and demands of a growing population. By evaluating that information, decision makers are offered valuable insights into urban activities and movement flows, enabling themselves into making the best and most prudent decisions for the good of the people they serve (Rasmussen et al., 2015). For example, urban planners can make a city more livable by mitigating congestion and planning for developments to cater to high volumes of people. Transport planners can understand the mobility patterns in greater depth, knowing times and locations of traffic hotspots on the roads, as well as on the transit networks. Even companies who want to streamline their products and services can do so to those who need it most. In the grander scheme of things, models that can predict how people travel in time and space can help in the fight to reduce our global carbon footprint. Our reliance on motorized forms of transportation is one of the key driving forces of climate change, further motivating transport researchers and providers to strive towards more sustainable transport networks, one that promotes the use of non-motorized or public transportation for example.

2.2 Transport mode detection

A key way to achieve a greater understanding of human travel patterns is an understanding of the modes of transportation they take, as well as its corresponding temporal distributions. This problem has been tackled differently based on the what objective of the researchers is and can be summarised into three main branches (Prelipcean et al., 2017). First is Location-Based Services (LBS) whereby the goal is to detect the mode as close to real time as possible so that important and relevant information can be given to the commuters or interested parties at the suitable times and places, such as with Parkbob's smart car parking application[2]. This "on-demand" kind of mode detection is in line with many cities aspirations toward a fully functional smart city, where resources can be allocated on the fly to places or people who require them. Another huge and long-standing branch is transportation science, which aims to generate reliable and usable statistical data on usage of the transport system. This data is ued as the foundation to answer many city planning questions and further many of the applications mentioned previously. With such valuable uses of this data, transportation scientists have continually tried to gather this information mainly in the form of actively solicited travel diaries, or paper, internet and phone surveys (Rojas et al., 2016; Shen and Stopher, 2014a). These traditional approaches have proven to be inaccurate, time-consuming and resource intensive as it relies on people manually self-reporting their daily activities, travels, and corresponding schedules. Often, these get under-reported due to forgetfulness or the amount of effort it requires (Bohte and Maat, 2009). Transportation scientists are motivated to overcome these problems and automate part of this data collection, as the positive impact of good quality data on transportation mode usage is compelling. Thirdly, transport mode detection is of great interest to the field of human geography, whose objectives are largely to enrich these datasets with domain-specific semantics such with associated Points-Of-Interests. This field of research has methods of mode detection similar to that of transportation science, but has outputs that cover a huge scope, using these human mobility trajectories to answer a myriad of questions such as linguistic evolution or human interaction patterns(Prelipcean et al., 2017). This research will have aims more line with that of transportation science, in developing methods to collect information on people's mode choices. The applications however, are motivated to support Parkbob, whose services lie more in the LBS realm.

[2] http://www.parkbob.com/

2.3 Transport mode detection using Global Positioning Systems data

In May 2000, the US government decided to remove selective availability of Global positioning systems (GPS), which was a military effort towards security reasons to intentionally degrade GPS signals. Now GPS devices can determine locations with accuracies of less than 10m (Bohte and Maat, 2009) for various purposes in the civilian world. Some examples include in agriculture to accurately monitor yield data or to enable work in poor visibility or weather, aviation for the continuous provision of reliable and accurate information on flights as well as for more efficient air traffic management (Kaplan and Hegarty, 2005). Disaster management is another area that benefits from this technology. When little information is available, GPS makes mapping of disaster zones possible. Flood and earthquake prediction capabilities are also improved with this technology (Kaplan and Hegarty, 2005).. Transportation is a great benefactor of GPS, with more accurate positioning leading to better schedule adherence and transport demands, for example.

As compared to more conventional means of collecting such data through surveys and travel diaries, collection via GPS devices alleviates many of the formers' shortcomings, on top of providing greater opportunities of quality and quantity of data collected. Providing more comprehensive information on origins, destinations and the routes taken between them, trip start and end times as well trip lengths can be more realistically achieved by the respondent as they do not rely on memory or need to make the effort to pen down their schedule. The data tends to be more accurate and independent on the respondents perception of durations, distances, and departure/arrival times (Rojas et al., 2016; Shen and Stopher, 2014a). Underreporting is also avoided as the GPS logger captures all movements of participants (Stopher et al., 2008). This also means that data collection can be done over a prolonged period of time. Furthermore, GPS can also be used in conjunction with traditional methods as a means of verification. These advantages have led to the rise of incorporating these technologies as a supplementary tool or to completely replace travel surveys.

Now, GPS units are commonplace and are accurate and lightweight enough to make this a feasible alternative to data collection, facilitating more complete analyses (Bolbol et al., 2012; Gong et al., 2012). More complex travel patterns can be mined from information on mode such as what combinations are taken, route choices in multi-modal trips and how they vary on different days or at different times.

2.3.1 Pre-processing

The raw GPS datasets collected are extremely large, sometimes containing logs in the order of millions. This also includes irrelevant data such as when the person is not travelling. Combined with issues like signal loss and cold starts (when the device is turned back on hand takes time to recalculate information), a set of pre-processing techniques must be applied to turn this dataset into a comprehendible information source. This usually entails cleaning the data of noise and then segmenting them into individual trajectories, or trips with start and end points (also known as Segment Identification or SI). A commonly used method for this step is through the use of rule-based algorithms, often by identifying stop points and assign them as start/end points of the trajectory (Shen and Stopher, 2014b). Many studies use a threshold of 120 seconds as the minimum time a person must be in the same place for it to be considered a start or end point which could be activities or mode changing points (Chung and Shalaby, 2005; Gong et al., 2012; Stopher et al., 2008). Traffic light change times or bus stop tend to be lower than that, making it a reasonable criterion. To date, this rule is still being used, but also supplemented with other rules. For example, Schüssler & Axhausen (2009) combine this threshold and point density as their criteria for activity detection. Activity locations are detected when observations meet two criteria: (1) low speeds (<0.1m/s) for more than 120s; (2) the points are located very close together (diameter of <30m). This is seen in Stopher et al.'s paper as well, where location and speed conditions (distance travelled, speed and heading change) have to hold for more than 120s on top of a point density consideration for it to be identified as a mode change point (Stopher et al., 2008). Examples of these spatio-temporal rules in different permutations can be found in many other studies as well (Bohte and Maat, 2009; Gong et al., 2012; Rasmussen et al., 2015; Shah et al., 2014). Gong et al. cluster points within 50 m of each other for a minimum of 200s. Trip ends are hence the first point of the cluster and the next trip's start point is the last point of the cluster. As the consistently higher spatial and temporal granularity of GPS mean that these thresholds can afford to be that low without having the concern of excluding relevant points. However despite the good results of the studies that use this threshold, some argue that the dwell time of 120s or more is excessive and can lead to the underestimation of many trips that are shorter, and concluded that 60 seconds would be a better parameter input (Shen and Stopher, 2014a). Regardless, the main theory behind the method is still widely accepted as a sound approach for SI. Other approaches to this include using density-based clustering and machine learning methods (Gong et al., 2015).

2.3.2 Mode detection

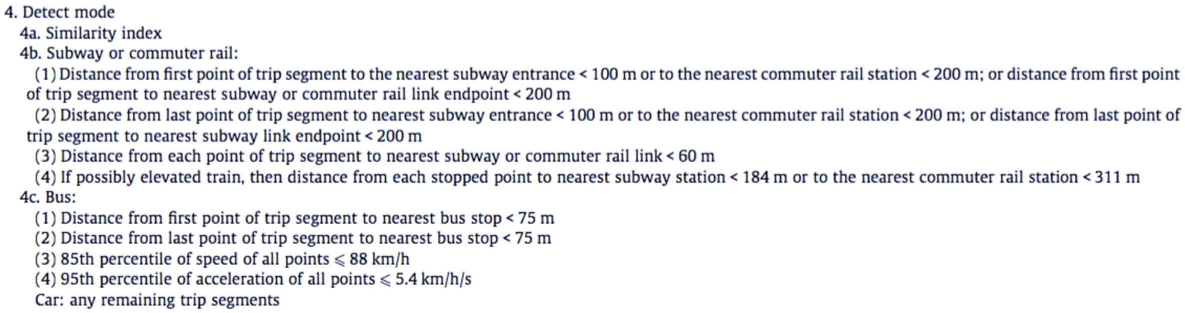
Earlier GPS studies differentiated between walking, driving and motorized modes (Bohte and Maat, 2009; Chung and Shalaby, 2005), but more recent studies have begun to detect public transportation modes as well (Axhausen and Schüssler, 2009), and many go a step further as to classify these motorized modes into the various modes of public transportation like buses or trains (Gong et al., 2012; Rasmussen et al., 2015; Stenneth et al., 2011).

Input variables that determine modes such as average, maximum and standard deviation of speed, acceleration measures, average dwell time and average heading change are frequently used in this stage (Gonzalez et al., 2010; Stenneth et al., 2011; Xiao et al., 2015). Due to the growing popularity of geospatial data, many mode detection studies also incorporate data from external sources such as the transportation network or real time public transport information (Asgari et al., 2016; Gong et al., 2012; Stenneth et al., 2011; Tsui and Shalaby, 2006). Especially in times of heavy traffic when movement is slow, it is difficult to infer mode solely from velocity. The areas where a bus or tram can be are generally fixed. This type of data fusion with GIS data has proven to make mode detection more robust and produce much better results than when compared to the baseline method without contextual information.

When it comes to research more in the line of transportation science, there seems to be a preference for inferring modes using Rule-based methods (Bohte and Maat, 2009; Chung and Shalaby, 2005; Gong et al., 2012) and fuzzy logic systems (Axhausen and Schüssler, 2009; Rasmussen et al., 2015). Some also use supervised learning methods such as Random Forests. These three types of methods will be described in greater depth in the following section.

Rule-based heuristics

Gong et al. (2012) use a rule-based GIS algorithm that automatically processes GPS data to detect 5 modes. The algorithm also recognizes whether mode transfers within a trip are feasible. By combining GIS data like street centerlines, bus routes and stops, subway lines, stations and station entrances, this method is able to achieve a promising 82.6% accuracy. First, trajectories are split into segments by identifying stop and mode change points. Through a set of hierarchical rules, walk modes are inferred first based on speeds. Next, by comparison to the public transportation network, rail followed by bus modes are inferred. The rest are considered as car modes. Thresholds used are based on the specifications of the city. For example, in the study area of NYC, the maximum length of trains was 184m, so the threshold for maximum distance to a station to be considered rail mode was set at 200m to account for the fact that the user could be at the end of the train while it stopped at the station. Other rules derived from context aware information include the third rule, where the maximum speed and acceleration of an express bus in New York City is 88km/h, or 1.5m/s^2. The full set of rules can be seen in Figure 1.The study showed promising results, however noted that the relatively lower accuracy of bus and car mode identification was due to the dense street networks and the consequences of the urban canyon effect which sometimes cause a parallel shift of GPS observations (Gong et al., 2012). This can lead to misclassification of bus modes as car or walk modes. Map-matching techniques derived from Chung and Shalaby's (2005) paper were also applied to match walk segments to street segments. Furthermore, that paper developed a trip reconstruction tool using GPS data with a rule-based algorithm as well, which achieved an accuracy of 92% of all four modes of interest. Bohte and Maat (2009) also use straightforward rules (Figure 2) on measures of maximum and average trip speeds to infer modes, starting from the slower modes, walk, then bicycle and car, followed by public train modes, due to its characteristic location that is constrained by the rail network. Stopher et al. (2008) manage to achieve an impressive 95% accuracy with another hierarchical set of rules together with external transport network data. Furthermore, the studies found that the distinction between bus and car modes was very sensitive to their specific rules, and due to the similar speed profiles of both modes, there is usually a high trade off between success rates for one mode and the other (Bohte and Maat, 2009; Gong et al., 2012).



4. Detect mode

4a. Similarity index

4b. Subway or commuter rail:

(1) Distance from first point of trip segment to the nearest subway entrance < 100 m or to the nearest commuter rail station < 200 m; or distance from first point of trip segment to nearest subway or commuter rail link endpoint < 200 m

(2) Distance from last point of trip segment to nearest subway entrance < 100 m or to the nearest commuter rail station < 200 m; or distance from last point of trip segment to nearest subway link endpoint < 200 m

(3) Distance from each point of trip segment to nearest subway or commuter rail link < 60 m

(4) If possibly elevated train, then distance from each stopped point to nearest subway station < 184 m or to the nearest commuter rail station < 311 m

4c. Bus:

(1) Distance from first point of trip segment to nearest bus stop < 75 m

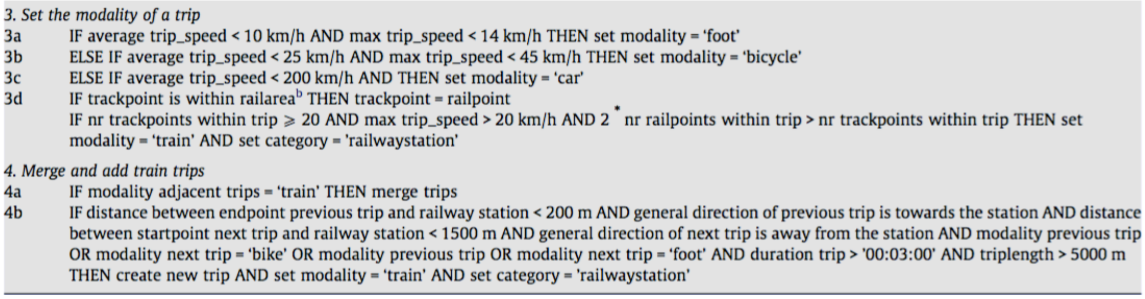
(2) Distance from last point of trip segment to nearest bus stop < 75 m

(3) 85th percentile of speed of all points ⩽ 88 km/h

(4) 95th percentile of acceleration of all points ⩽ 5.4 km/h/s

Car: any remaining trip segments

Figure 1 Rules used in Gong et al.'s paper in the mode detection process (Gong et al., 2012)



3. Set the modality of a trip

3a	IF average trip_speed < 10 km/h AND max trip_speed < 14 km/h THEN set modality = 'foot'
3b	ELSE IF average trip_speed < 25 km/h AND max trip_speed < 45 km/h THEN set modality = 'bicycle'
3c	ELSE IF average trip_speed < 200 km/h AND THEN set modality = 'car'
3d	IF trackpoint is within railarea[b] THEN trackpoint = railpoint IF nr trackpoints within trip ⩾ 20 AND max trip_speed > 20 km/h AND 2 * nr railpoints within trip > nr trackpoints within trip THEN set modality = 'train' AND set category = 'railwaystation'

4. Merge and add train trips

4a	IF modality adjacent trips = 'train' THEN merge trips
4b	IF distance between endpoint previous trip and railway station < 200 m AND general direction of previous trip is towards the station AND distance between startpoint next trip and railway station < 1500 m AND general direction of next trip is away from the station AND modality previous trip OR modality next trip = 'bike' OR modality previous trip OR modality next trip = 'foot' AND duration trip > '00:03:00' AND triplength > 5000 m THEN create new trip AND set modality = 'train' AND set category = 'railwaystation'

Figure 2 Rules used in Bohte & Maat's paper for mode detection (Bohte and Maat, 2009)

Another study by Kasahara applied a rule-based mode detection method, but detected high-speed modes first, and assigned modes to the individual observations instead of trips (Kasahara et al., 2017). Observations of the same mode are subsequently merged into trips, provided the time period is less than a certain threshold. However, despite the high performance, some opine this method struggles with low generalizability as rules obtained from a one city may not be so applicable to another city due to various reasons like the built environment affecting GPS signals (or density of cell towers, affecting overall coverage and signal strength). However, the simplicity and comprehensibility of these methods mean that it is feasible to derive parameters for each city (lengths of trains or average distance between stops from the cities transport provider, for example).

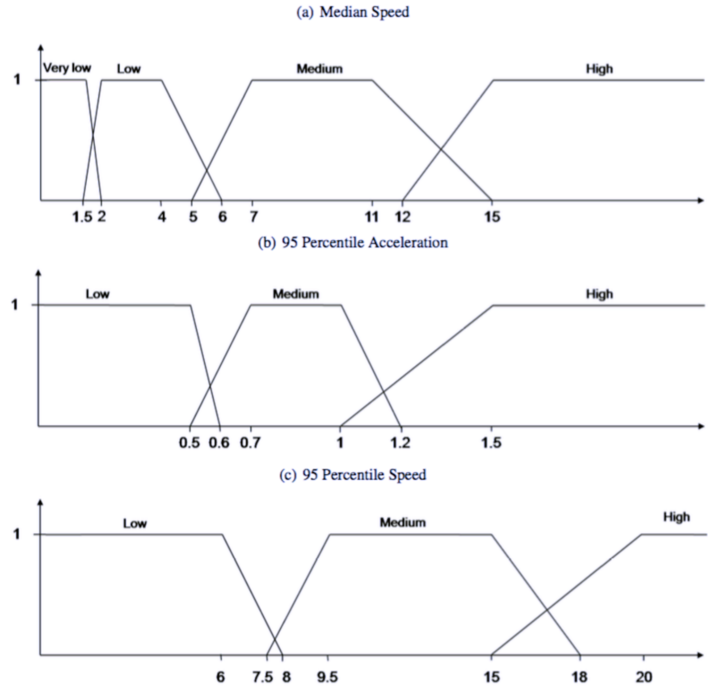
Fuzzy Logic Systems

Fuzzy logic (FL) systems are powerful predictive models as they can handle uncertainty and vagueness in a way that is understandable by humans. However, the success of a fuzzy expert system lies in proper selection of its functions and parameters, which are usually done manually (Das and Winter, 2016a). Unlike crisp sets with hard border values, fuzzy set theory assigns membership values to an element, introducing the concept of partial membership of that element in a set, or a number of sets.

The way FL is used in these studies is mostly subjective, as the empirical approach to generating these rules is dependent on human judgment to define them. Schussler & Axhausen (2009) use an open source FL platform to generate trapezoidal membership functions of their fuzzy variables. The variables were median of speed, 95th percentile speed and acceleration, and were explicitly chosen over average values to make the algorithms more robust against outliers. Figure 3 shows the membership functions of each variable. A minimum of one rule is defined for each mode based on these membership functions, as seen by the examples in Table 1.

Ambiguity and fuzziness is intentionally introduced through the rules as well as from the overlapping membership functions. This can be especially useful for modes that have variable speed profiles, such as buses which start and stop frequently, and speed changes depending on whether they are in the city center and the stops are close together, and in residential areas where there are longer stretches between stops (Tsui and Shalaby, 2006). Modes are finally inferred based on the membership values from the aggregated membership functions.

Ramussen et al. (2015) applied a similar technique to their study area in Copenhagen using the same variables, but with values derived from their own expert knowledge and analysis. They also combined the FL system with a Rule-Based method first sieve out rail trips, due to the assumption that rail lines are characteristically different to road networks and thus a trajectory aligned with rail lines has a high possibility of being a rail tip. The study found that the FL rules were still insufficient to effectively distinguish between bus and car modes. In response to that, they measure alignment of the identified GPS stops with that of the bus routes. Both studies applied feedback mechanisms for weird combinations such as car to bicycle or car-bus-car were applied to correct these to more realistic modes. For example, if the sequence of modes were car-bus-car, the algorithm would flag it and reclassify the trajectories as a car trip. High spatial and temporal granularity allows for shorter and more detailed trips that may constitute one single journey to be identified, allowing for this method to act as a suitable feedback algorithm.

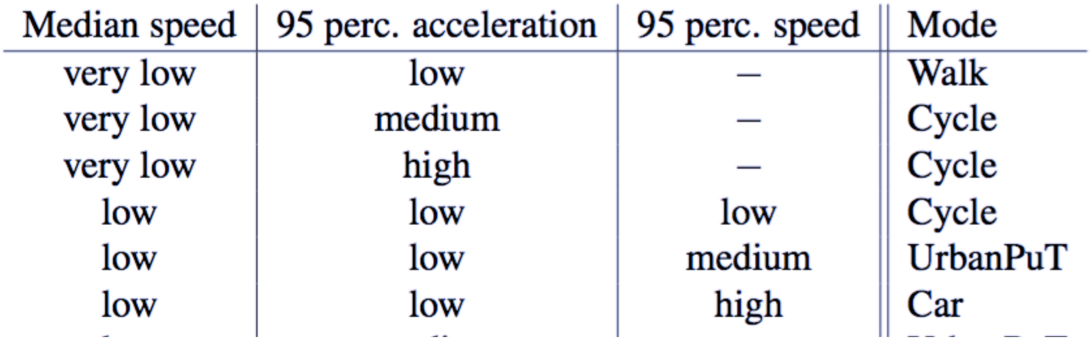

(a) Median Speed

(b) 95 Percentile Acceleration

(c) 95 Percentile Speed

Figure 3 Fuzzy Logic membership functions generated by human expertise (Axhausen and Schüssler, 2009)

Table 1 Examples of fuzzy rules for mode detection (Axhausen and Schüssler, 2009)



Median speed	95 perc. acceleration	95 perc. speed	Mode
very low	low	–	Walk
very low	medium	–	Cycle
very low	high	–	Cycle
low	low	low	Cycle
low	low	medium	UrbanPuT
low	low	high	Car

The paper did not report any accuracy measures of this probabilistic method, but compared the results with the official census data on travel behavior that was released a few years prior to the stud, and concluded that this form of mode detection yields realistic and reasonable results. A study released after that designed a more complex FL system so as to classify more modes. A few more fuzzy variables were added to the list, including proximity to a network like the railway or bus network. This method was able to detect walking, bicycling, car, ferry boat, sail boat, train, subway, bus, tram and flight modes using GPS data and these fuzzy variables with an accuracy of 91.6% (Biljecki, 2010). The fuzzy system had certainty factors applied to each result to measure the confidence of the inference.

One drawback of FL systems is that these rules tend not to take into account inter-variable correlation, and as the rules are generated using experts' understanding of the field, any class (mode) additions to the model would be extremely costly (Elkan et al., 1994). This may prove to be a problem when trying to transfer this method that was designed for GPS data to CSD. However, it is still possible to construct a FL model without expert given a set of input and output pairs. The task then is fundamentally akin to determining a system that provides the best fit to these pairs (Mendel, 1997), and will be explored further in the next chapter.

Machine Learning

Due to certain limitations of setting rules and algorithms in programs, some studies have turned to machine learning methods instead. These methods commonly include neural networks and tree-based models, among a few others.

Gonzales et al. applied a reduced sub sampled GPS dataset to neural networks to infer modes. The subset consisted only of critical points, which were characterized by heading change and a minimum speed so as to remove redundant data and minimize processing. The inputs chosen for the neural network algorithm were the common variables such as acceleration, speed, distances between stop locations, dwell time, as well as GPS specific variables like estimated horizontal accuracy uncertainty (Gonzalez et al., 2010). The neural network was able to learn to distinguish between walk, car and bus trips. However, the paper cited that the critical points used in the proposed algorithm were insufficient to achieve a good result. Tsui & Shalaby (2006) proposed a hybrid method that combines this neural network with a fuzzy logic system for mode detection. The fuzzy variables chosen were similar to the other FL studies described above, with the inclusion of data quality. However, the parameters of these variables and their membership functions were set by a neural network algorithm (NEFCLASS-J). Their work managed to identify modes (walk, bus, bicycle, car, rail) with an overall accuracy of 91%, though the performance of bus modes was relatively poor due to the considerable overlap of characteristics with other travel modes. These, combined with the large variability of movements in buses were cited as reasons for this poorer performance. The paper however, did not report on the mode share of the actual data.

Several works also include temporal measures in their learning methods such as time of day to give context to a probability model to estimate mode choice (Liao et al., 2007). Stenneth et al. (2011) incorporate live bus and train times when inferring between stationary, walking, cycling, bus, driving, and train modes. They extracted variables such as average speed, heading change, acceleration, as well as context-aware information like average bus line closeness, rail line closeness as well as bus stop closeness. The authors ran these variables through several learning algorithms (Random Forest, Decision Tree, Naïve Bayes, Bayesian Network and Multilayer Perceptron) and found that Random Forest had the best performance. A few important strengths of RF that are relevant to this study are that it is one of the highest performing machine learning algorithms in terms of accuracy and can run efficiently on large datasets. Estimates of what variables are important are also included in the output, which can be useful for purposes like dimension reduction (Degenhardt et al., 2017). It is also able to generate pairwise proximities between data points that can be used as input in other classification methods like unsupervised k-medoid clustering.

2.4 Transport mode detection using cellular network data

Pervasive technologies such as mobile phones create datasets that give an inside look of how people use the city's infrastructure. Urban planning is one of the greater benefactors of the analysis of this collective personal location data. Mobile phone traces contribute to a massive pool of passive data that can provide knowledge on the whereabouts and movements of individual users. According to the Global System for Mobile Communications Association (GSMA), an international trade body that represents the interests of the world's mobile network operators, there are about 7.7 billion mobile connections by 5 billion unique subscribers in 2017. 465 million of these subscribers reside in Europe alone[3]. This large potential has not gone unnoticed as many of these operators have begun to experiment with new business models that would generate revenue from both their mobile subscribers as well as other customers such as traffic analysis, advertising and marketing, and social networking companies. As such, it is no surprise that the sharing of such mobile data with research communities has started to pick up speed (Calabrese et al., 2015).

The main hurdle is the lower spatial resolution, inconsistent and sometimes sparse samples of data. As such, they require a specialized set of techniques for extracting valuable and usable information from them. There are various types of cellular network data such as call detail records and cellular signaling data. The latter, which is the one that will be used in this research is known by many names, including floating cellular/phone data, sightings data, and so on. Furthermore, this data type can have varying properties depending on whether the phone is connected to a 2G, 3G or 4G network. This gives an indication of how recent this data type has been incorporated into such research fields.

For example, Sevtsuk and Ratti (2010) address how coarse-grained call volume data in Rome can be used to tease out properties of user mobility, where they found regularity and patterns in urban mobility at different times of the days, as well as which day of the week it was. Other indicators like demographic, economic and infrastructural indicators were used to supplement and account for these patterns. Travel routes can also be estimated using cellular network based voronois and map matching (Tettamanti et al., 2012). Other studies also use the fluctuations of

[3] https://www.gsma.com/newsroom/press-release/number-mobile-subscribers-worldwide-hits-5-billion/

signal strength in GSM cellular data to estimate more precise geographic coordinates for these purposes (Thiagarajan et al., 2011).

Transport mode detection is another area of research using cellular network data that is still in its early stages. This is largely due to data's lower spatial and temporal granularity, the main challenges to the computation of specific measurements on speed. However, there have been a few studies attempting to estimate coarse speeds according to the rate of change of the connected cells, as well as the distances between them (Gonzalez et al., 2008; Reddy et al., 2010a; Sohn et al., 2006). Others have made sense of coarse CDR data by clustering travel times of trips into the corresponding transportation mode clusters (Kalatian and Shafahi, 2016; Wang et al., 2010). The next sections will delve deeper into how cellular network data is generated, processed, and used for transport mode detection.

2.4.1 Mobile phone network structure

The basic network structure is composed of a Core Network (CN) and Radio Access Network (RAN). The CN is divided into either Circuit-Switched (CS) for activities like voice calls or Packet-Switched (PS) domains for packet data transfers. Radio communication occurs between the mobile phones (terminals) and the base station serving that cell. As such, cells are the smallest spatial entities in the cellular network, with a geographic coverage that varies from magnitudes of meters (microcells), up to several kilometers (macrocells). Several cells together make up a Location Area (LA) (Janecek et al., 2012; Miao et al., 2016). This structure is illustrated in Figure 4.

2.4.2 Data Generation

Mobile phone positioning occurs whenever a terminal communicates with the network, essentially when a user uses his phone (Chen et al., 2016). Calabrese et al. (2015) categorize mobile phone data into two types, event driven and network driven. Event driven data is generated during billed activities such when calls or texts are made, or when data is being transferred while browsing the Internet for example. These include Call Detail Records (CDRs) and Internet Protocol detail records (IPDRs) respectively. At this stage, the terminal is said to be in active state, whereby the voice call or data connection is open. At any given time, the majority of mobile terminals are not in the active state, but in the idle state. Even terminals that have their data connection permanently switched on remain in the idle state, switching to the active state only during packet bursts, like data downloads (Janecek et al., 2012). The information in each event that is ultimately recorded in the data depends largely on the mobile phone provider that operates the network. For example, CDRs could include the IDs of the callers, receivers, cell towers and start and end time stamps. Similarly, IPDRs will consist of information on Internet usage and other cellular data related activities.

Network driven data is also known as floating cellular/phone data or signaling data and is generated whenever a phone is localized, i.e., during different types of location updates (Figure 4). Periodic updates occur on a periodic basis as determined by the telecommunications provider to generate periodic information on which cell tower the terminal is currently connected to. Handovers are generated when an active terminal moves between two cells and lastly, mobility location updates are generated when a terminal moves between two location areas. As such, depending on the state (active vs. idle) of the terminal, the spatial granularity of the data recorded can be at the cell level or at the location area level.

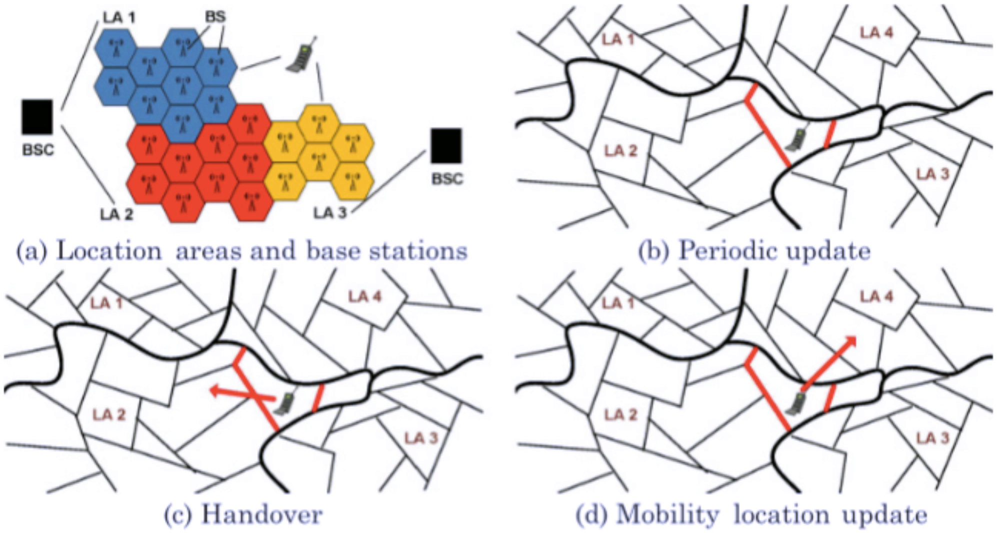

LA 1
BS
BSC
BSC
LA 2
LA 3
LA 4
(a) Location areas and base stations
(b) Periodic update
(c) Handover
(d) Mobility location update

Figure 4 a) Location Area and Base Stations b) Periodic updates c) Handovers d) Mobility location updates (Calabrese et al., 2015)

2.4.3 Spatial and Temporal Granularity

The frequency of the data depends on the type of mobile data generated and is largely user dependent. Chen et al. (2016) found that the frequency of both event and network driven data types display high heterogeneity in the number of times the phone was localized or a call was made for example, whereby the majority of users have a small number of records and only a few have more a large numbers. Each voice call generates one CDR. However, that same call might generate multiple network driven data points if multiple cells are traversed during the duration of the call. Periodic updates are typically in the order of a few hours (Wildham et al., 2015). For purposes of clarity, from this point on the term cellular signaling data (CSD) will refer to both event driven and network driven data.

A study that used CDRs found that the average time interval between each event was about 8 hours (Gonzalez et al., 2008). These intervals accurately represent the time intervals between each call and are much longer than datasets that include network driven data as well. In the latter, as a single call might trigger multiple network driven events, these events might tend to be more clustered together in terms of the times they were recorded(Chen et al., 2016). In Calabrese et al.'s (2011) paper, the average inter-event time intervals that included network driven data was found to be 260 minutes, with the average of the quartiles' medians to be less than 1.5 hours. As such, the data was fine enough for the researchers to identify stops of lesser than that time interval. However, it is also important to note that stops of less than 1.5 hours will be missed. This might add to inaccuracies of the processed data especially since some household travel surveys define a stop exceeding 5 minutes to be an activity that should be recorded (Chen et al., 2016). As for location area updates, there may be cases whereby no updates are sent from the mobile phone despite large amounts of movement if the location area covers a large area, some several hundreds of kilometers.

In terms of spatial granularity, these data types differ in from GPS data in the sense that the location information has to be estimated using various methods. As this means that the cell phone events only contain approximated locations, this is significantly less accurate than that of GPS data (Horn et al., 2014). Triangulation is often used and results in coordinates that do not correspond to the cell tower location but are an estimate of the terminals position. Measures such as received signal strength, transmission time and angles of multiple towers are used in the estimation if there are multiple base stations in available range (Chen et al., 2016). The algorithms here used are usually undisclosed by the mobile provider and use both event driven and network data. Furthermore, most mobile providers do not disclose the structure and organization of their cellular networks, or the spatial extents of each cell or LAs, which vary depending on the density of towers and the level of urbanization (Widhalm et al., 2015). Experiments show that the spatial resolution was found to be from the order of a few meters (Chen et al., 2016) to about 300m (Calabrese et al., 2011; Jiang et al., 2013) or 500m (Horn et al., 2014) in urban areas where the density of cell towers is much higher, to that of several kilometers (Horn et al., 2014; Widhalm et al., 2015) in rural, less heavily populated areas.

2.4.4 Pre-processing

Several pre-processing techniques must first be applied before valuable information on human patterns can be extracted. These vary depending on the research aims, but generally, noise reduction techniques in the form of filters or through clusters are applied to filter out inaccuracies in the data. Next, these observations are then segmented into individual segments, whereas some GPS studies assign modes to the observations and then group similar consecutive observations into a mode trajectory (Kasahara et al., 2017; Reddy et al., 2010b).

Noise reduction

Pre-processing needs to be done to reduce noise. This is done to lower the influence of outliers that arise from various phenomena on the final analysis. One such phenomenon, known as the ping-pong effect, occurs when the terminal bounces back and forth between multiple base stations while the user is not moving (Fiadino et al., 2012; Miao et al., 2016). This occurs due to fluctuations in the received signal strength and hence leads to the oscillation between different cell towers despite being stationary. These fluctuations in signal strength are also likely to have an impact on the estimated triangulated position, leading to what appears to be drifts and shifts in the location of the data points. Also, in the event there are several cell towers whose signals reach a terminal, the connection of this device may hop between these towers. This means that outliers can suddenly occur kilometers away within an unrealistically short period of time. While some (like the above) can be the result of localization errors, others can be intentionally triggered for privacy protection reasons. For instance, arbitrary events can be inserted into the mobile traces to prevent the creation of movement profiles. These events are part of efforts towards privacy protection and are called temporary mobile subscriber identities or TMSI (3GPP, 2010)[4].

One approach is through pattern-based recognition and this requires information on which cell tower the phone is connected to (Iovan et al., 2013; Schlaich et al., 2010). Users with high oscillations between cell towers are identified with a proposed "jumpiness rule" using the number of updates and area codes. Another approach does this through speed-based corrections. A threshold is chosen to distinguish between what is a reasonable speed and what is not. Instances that produce values that exceed this threshold are flagged. This can also be done through a number of ways as explored by Horn et al. (2014). They tested a series of filtering techniques including a recursive naïve filter, recursive look-ahead filter and Kalman filter. Outliers are identified as data points where the speeds calculated are exceptionally fast, and the threshold was set to 250km/h. The recursive naïve filter simply removes any outliers from a sequential stream of events whereas the recursive look-ahead filter accounts for the possibility that the event before the outlying event is the outlier instead. The Kalman filter is more complex in that it takes a probabilistic approach and is a popular choice in data prediction tasks including traffic modeling using GPS and other sensor data (Faragher, 2012). It produces estimates of unknown variables in nosy time series through approximating joint probability distributions over the variables in their time frames (Kalman, 1960). Results show that the recursive filters outperformed the Kalman filter, and one of the reasons proposed was due to the temporal sparseness of the cellular signaling data. As such, the former is more suitable as a noise reduction method for more irregular data like CSD.

Trip extraction

In many of the studies using cellular network data to mine human patterns, individual trips are first extracted before they are analyzed for mode detection or aggregated for more large-scale analysis such as in urban activity analysis. To achieve this, key places must be identified. On top of assigning these users to these locations, there must be a distinction on whether these places are stops or the user is merely passing through it. The former can be places of activities, like work, home or leisure, origins or destinations when trying to generate Origin-Destination (OD) matrices, or more exact locations as start and end points of trips. Assigning these start and end

[4] 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Numbering, addressing and identification

points to specific locations can be done using a few methods. A more straightforward and commonly used approach is by using centroids of cell areas if the cell tower locations are known (Gonzalez et al., 2008; Tettamanti et al., 2012).

Many other studies use stop detection to filter out significant places, with the rationale being that if a person stops there for a reasonable time period these places are important in human patterns. Due to the noisy and raw nature of cellular network data, the same event can sometimes be registered as many consecutive events that could be related to various locations in its surroundings. These locations are filtered out either by a set of rules pertaining to space, space and time or speed. A commonly used method here is through spatial and temporal clustering. Wang et al. (2010) worked with CDRs, and use an incremental clustering algorithm to extract these stop locations. A radius corresponding to the estimated positioning error (set at 1 km) and minimum dwell time were defined as the thresholds to form clusters. The medoids of these clusters were found and the remaining points in the clusters were deleted. Consequently, these medoids were set as the start and end points of each trip. Jiang et al. (2013) apply similar thresholds to cellular signaling data, but with finer thresholds of 300m and 10 minutes to detect stay locations, as illustrated in Figure 5.

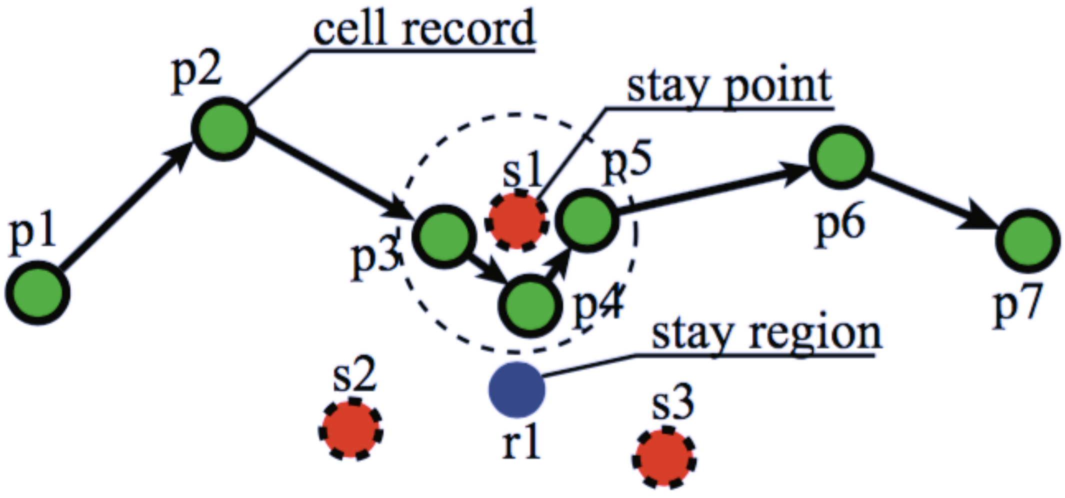


Figure 5 Illustration of data pre-processing to extract stay locations (Jiang et al., 2013) with distance and time constraints of 300m and 10 minutes.

Another study by Widhalm et al. (2015) uses a similar clustering algorithm to both CDRs and cellular signaling data but incorporates the geometry of the trajectory to filter out passing by locations. Figure 6 shows an example of how B in I) is not detected as a stop and that in II) is. As the study was mining urban activity patterns, the reasoning behind this was that significant extra distances travelled are often motivated by an activity.

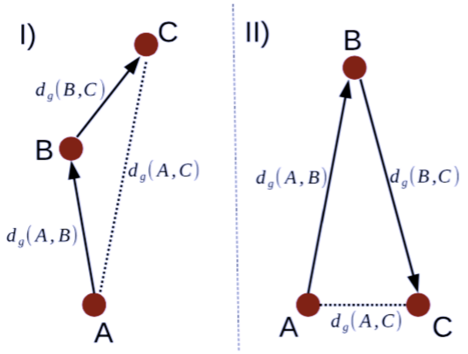


Figure 6 Detection of stays using geometry (Widhalm et al., 2015)

2.4.5 Mode detection

Once these trips are extracted they are now ready to be analyzed to identify the trip modes. In order to do so, information on these trips are extracted as trip features. Unlike in GPS studies that assign modes to individual observations, most cellular network studies only do so after grouping these observations into segments, treating the segments as the smallest unit to infer modes instead of on each individual observation. Due to the infancy of mode detection using mobile phone data, the studies are usually limited to CDR-only data and the existing methods used here can be classified into unsupervised k-means clustering, rule-based heuristics and machine learning.

Clustering

K-means

A well-known method of mode detection using cellular network data is with k-means clustering of travel times. Wang et al. (2010) worked with anonymized CDRs in their study. Start and end points of these trips were assigned to cells in a grid. Trips with the same ODs were grouped together. Trips over 63 minutes were removed as they were deemed to be too long a travelling time within the study area. K-means unsupervised clustering was then performed on the travel times of each of these OD groups, with distinctions between weekdays and weekends. K-means is a centroid-based clustering algorithm that uses the mean value of each cluster (centroid) to represent the cluster (

Figure 7). The goal of K-means is thus to reduce the sum of squared error between the individual objects in the cluster and their centroids (Hastie et al., 2009a). The clustering partitioned the records into two separate clusters corresponding to the modes of interest, namely driving and public transport (Wang et al., 2010), where the clusters with the shorter travel time is assigned to driving and vice versa. There is an assumption of single modal trips here, similar to many mode detection studies of this nature. The error of the inference is then calculated as the average of differences between the travel times and obtained from the clustering and that of reported by Google Maps. Silhouette values to measure how well associated the cluster members are to the representative of the cluster was also measured, and this indicated a good performance of the model. Due to lack of official census data of the city with regards to transportation mode, the model could not be validated against such official records. Kalatian and Shafahi's (2016) paper also detected walking, and used a similar approach. Their study worked on anonymized signaling data and grouped the trips into traffic zones instead of grid cells, and separated by time of day to account for traffic. Grouping them by the hour meant that there were insufficient records to perform clustering well. As such they were grouped into trips occurring at similar hours of the day, such as when people commute home from 4PM to 9PM. However, while the paper stated that validation was done against surveyed data collected by the city, the results of that validation were not included in the paper. These clustering methods only use one feature of the trips; the travel time and assigning modes to these clusters may not be so straightforward. In a city with a well-integrated public transport system such as many major cities in Europe, travel times when private or public motorized modes can be extremely similar, if not shorter.

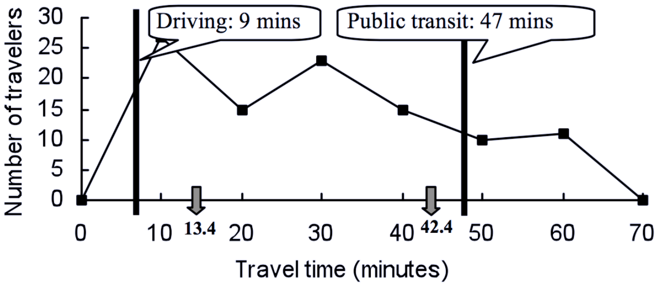


Figure 7 Trip data clustered to two subgroups, driving and public transit. The arrows show the average travel times of the subgroups. Black lines are the travel times as reported by Google Maps. (Wang et al., 2010)

Rule-based mode split

Qu et al. (2015) worked with CDR data to detect transportation mode using a rule-based mode split algorithm that combined speed, trip distance and a logit model. The paper focused on estimating transportation mode shares at the traffic zone level of the city, and only looked at commutes between work and home. These home-work trips were extracted through a longer observational period of 3 weeks and was possible as the dataset was not subject to anonymization every 24 hours. By approximating the home and work areas as places where the user is mostly found between 8pm – 7am and 9am- 5pm respectively, the travel times are subsequently estimated as the time difference between the latest time one is found at home and the earliest time one is found at work. This is to account for the fact that it is unlikely that a user makes a call just before leaving or upon arrival. As a result, they were able to estimate the travel distances and times between home and work, and subsequently from these two values, the speed. Here, the distinction is also between driving, public transportation and walking, where each trip only constitutes one of these modes.

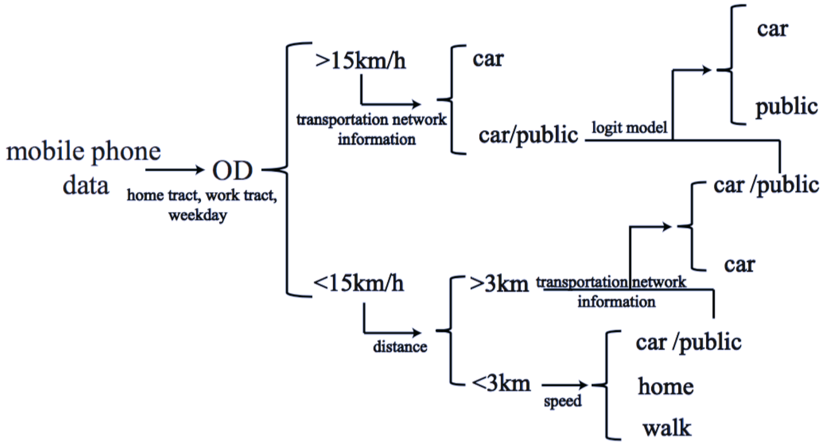


Figure 8 Framework proposed by Qu et al. (2015) for mode detection with CDR data through a speed split, augmenting the dataset with transportation network information and utility approximations.

Based on the assumption that 15km/h is the maximum speed of a non-motorized mode, Figure 8 shows the speed rule used to split the trips into high and low speed trips. High-speed trips whose average distance to the underlying public transportation network counted as a car trip. The rest are fed through the logit model. For the low speed trips, the distinction is made using trip lengths based on the rationale that people do not walk for more than 3km. Those more than 3km are also fed through the logit model, which is a discrete choice model that predicts an individuals choice base on utility or attractiveness. For example, in the study area of Boston, it is regarded as more attractive to use public transportation in the central Boston region and cars for the surrounding suburb region. This differs from many GPS studies using rule-based algorithms that usually detect slowest modes first. This can be attributed to the better resolution of data, enabling more representative measurements of slower speeds in modes like walking, with lesser chances of data inaccuracies resulting in higher speeds. Linear relations between census data and the predictions for each census tract are used to evaluate the performance of the model and the model does well for identifying car modes, but not for the other two. Also, while some areas observe high prediction accuracies, others have larger deviations from the survey data. One reason cited was the confounding effects of other factors such as income and land use that may have caused larger errors especially in their logit model.

Machine learning

Machine learning is sometimes used in mode detection studies using mobile phone data, more specifically, the GPS and accelerometer data collected from phone applications. However, a study by Sohn et al., (2006) applied some of these machine learning methods on cellular network data, or more specifically GSM data. A special mobile application was created for this study to capture this data. The dataset they generated were labeled with these modes and included signal strength values, cell IDs, as well as the channel numbers of at most 7 of the nearest cell towers. The method used here assumed that a user is stationary when the observations have a consistent set of towers and signal strengths, and moving when there are changes in these sets. They also found that the Euclidean distances between consecutive observations were proportional with the speed of movement.

In essence, a theoretical fingerprint of the signal strength and constituent cell towers were created for each observation and from this, seven features were chosen to train the model. This included the Euclidean distance, correlation of signal strengths from common cell towers and number of common cell towers between two measurements. The remaining variables were various descriptive statistics of Euclidean distances in the various windows of measurements. The classifiers were trained with a boosted logistic regression technique with a single-node decision tree. Overall the model performed well with an accuracy of 85% though the identified modes were only stationary, walk and drive. The signal strength information however is not available to this study as they were collected by an app developed by the researchers. There is still value in their work in terms of important variables that can be used in our study.

In a more recent study, Asgari et al. (2016) developed an unsupervised algorithm that enables the mapping of coarse mobile phone traces over a multimodal transportation network, where mobile trajectories are the observations and hidden states to be predicted are nodes of the multilayer graph. This unsupervised HMM completes the originally sparse trajectory and enriches it with the used modes by leveraging on the transportation layer type and their topological properties (i.e. route complexity). Transition probability predicts how likely an individual moves from one hidden state to another using factors like edge type, speed, and length. The model performs well and proves that using the transport network improves performance. Other studies have also attempted to match the observations to the underlying network.

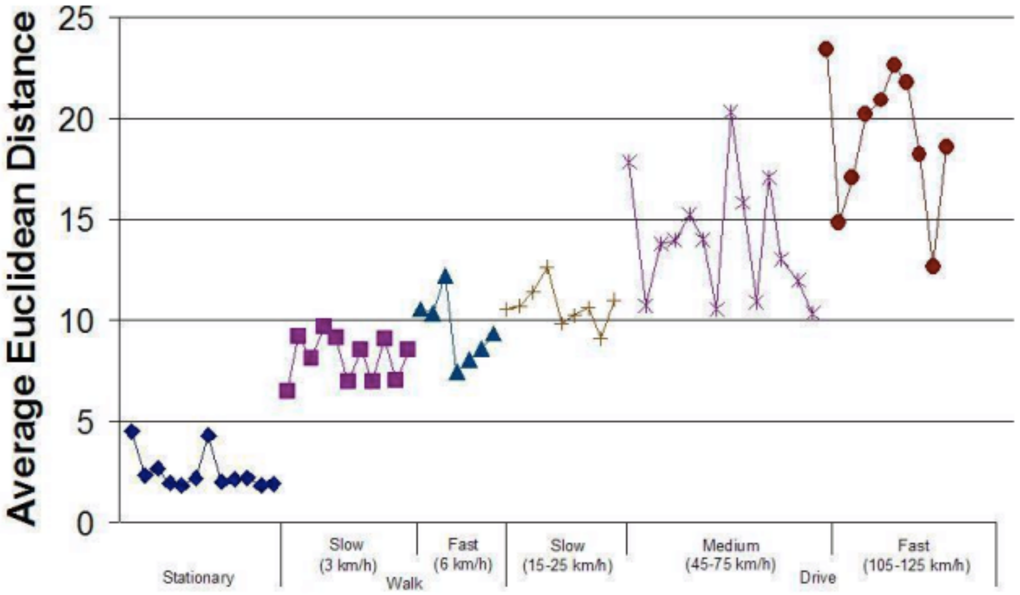


Figure 9 Average Euclidean distance between subsequent observations during stationary, walking and driving periods. (Sohn et al., 2006)

2.5 Variable selection

Methods like machine learning, FL systems and unsupervised clustering have proven to be powerful tools for classification tasks in both GPS and mobile phone data studies. Especially for data with high dimensionality, often selecting a reduced set of relevant variables would be ideal if the objective was to build a classification model for the purposes of identification. This will reduce the processing time and storage space needed. Furthermore, selected variables may also provide a suggested framework for future studies using CSD. To the best of our knowledge of existing work using CSD, there seem to be no documented cases of variable selection processes, or descriptions of such methods. As such, this paper has explored a few techniques that could be relevant to our research aims. Two options are explored here: Random Forest (RF), a tree-based model whereby labels are required and Principal Component Analysis (PCA) where they are not.

Random Forest

Random Forest is a tree-based learning model that has been shown to perform well in mode detection studies using GPS data. Labels are used in the generation of the random forest (RF). These algorithms have the power to handle high dimensional data and a key strength is that it outputs the most significant variables, which is especially relevant if the objective is variable selection. Another benefit of using RF is that it is able to account for and balance errors in imbalanced datasets, where one class may be disproportionately more represented than another.

Tree-based models use labels to build their trees, by splitting the population into two or more homogenous sets based on the most important variable. This is decided by using the Gini index or entropy to evaluate the quality of a particular split, and is usually used in classification problems rather than regression ones (James et al., 2013). The Gini index is defined as:

$$Gini\,(n) = \sum_{c=1}^{C} \widehat{p}_c^n \left(1 \;-\; \widehat{p}_c^n\right)$$

where $\widehat{p_c^n} = \frac{n_c}{n}$ is the proportion of individuals that have class *c* at node *n.* Gini is lowest when all observations in the nodes belong to the same class, and increases as the observations of the same node have a more even class distribution. The mean decrease of Gini or the information gain for splitting at node *n* on variable x_i, is defined as the difference between impurities of the node and the weighted averages of their child nodes:

$$Gain(x_i,n) = Gini(x_i,n) - w_L Gini(x_i,n^L) – w_R Gini(x_i,n^R)$$

Where n^L *and* n^R are the left and right child nodes of parent node *n*, and the weights assigned to the left and right nodes are w_L and w_R respectively. Based on this calculation, variable x_i with the lowest impurity is selected to be the basis of the split at node *n.* An alternative to the Gini coefficient is another measure of mean decrease in accuracy. Each tree that uses this particular attribute will compute value separately and then the average of all loss of accuracy is calculated (Degenhardt et al., 2017).

For example, in a sample of 100 children with variables gender, height and age, half of them ate meat and the other did not. The most important variable of determining their meat consumption status would be the one that produces the most homogenous or pure sets after a split based on that variable, where one resulting set has a high percentage of non-meat eaters and the other has a high percentage of meat eaters. RF is an extension of these decision trees in that it grows multiple trees. The definition of an RF algorithm is *"RF is a classifier consisting of a collection of tree-structured classifiers {h(x, k), k = 1,...} where the {k } are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x."* (Breiman, 2001, p. 2). Each tree will have one vote that will count towards the final classification. While RF is generally considered a supervised machine learning method, it can also be adapted in for unsupervised learning to derive a proximity matrix from unlabeled data (Shi and Horvath, 2006). This will be elaborated further in Section 4.8.

As for its role in variable selection, there are various approaches proposed to identify the most important variables based on this ranking. Degenhardt et al. (2017) did a comparative study of these methods and concluded that the Boruta method was the most powerful approach, and will be described further in Section 4.9. The overarching concept is to add randomness to the system and by collecting results from this system, the deceptive impacts of random fluctuations and correlations can be lessened, providing a better picture of which attributes are really important (Kursa et al., 2010).

Principal Component Analysis (PCA)

PCA is an unsupervised process of transforming data by plotting it on different axes so as to derive a set of smaller representative variables, or principal components (Abdi and Williams, 2010). The aim of PCA is to try and explain as much variation as possible in the data. Since labeled data is not required, PCA is especially useful to determine inputs to methods such as unsupervised clustering. These principal components are axes whereby the data is most spread out when projected to it. In order to find these lines, one derives eigenvectors and values, which come in pairs. The eigenvector is a direction and the eigenvalue is a number representing how much variance there is in the data in that direction, or how spread out the data is in that direction. The first principal component is thus the eigenvector with the highest eigenvalue, and can also be seen as the line that is closest to the original data (Abdi and Williams, 2010).

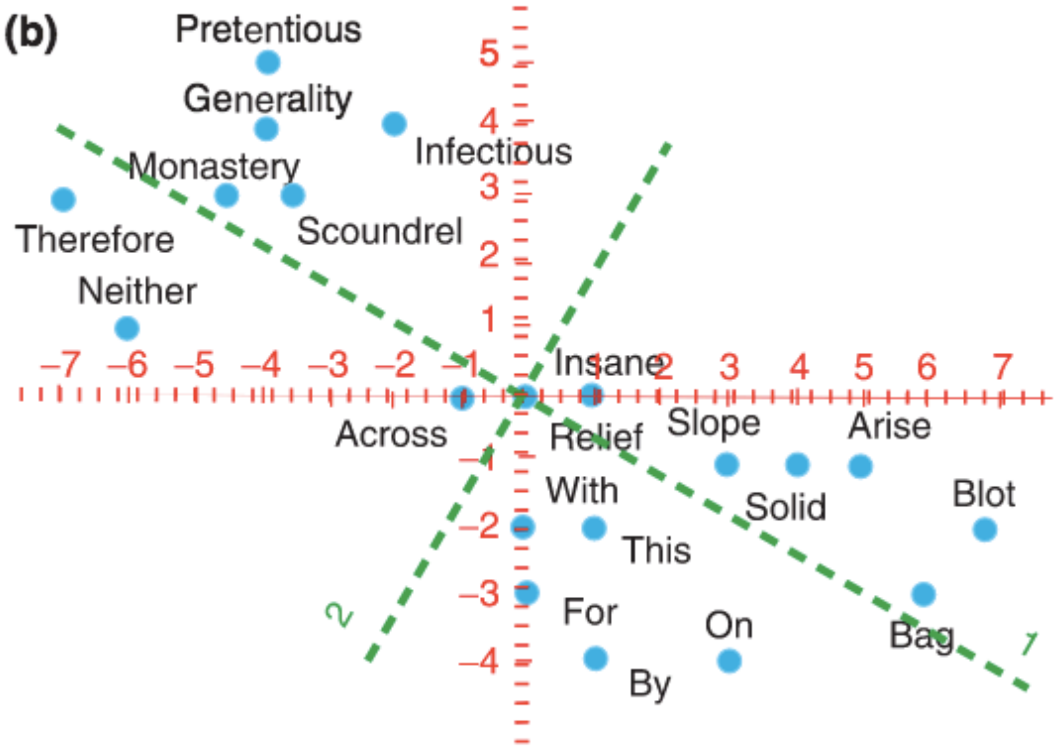


Figure 10 Example of first two components based on two variables, word length and number of lines in dictionary definition. Green lines represent the vectors whose main direction leads to the maximum sum of squared distances from the points to the vectors (Abdi and Williams, 2010).

PCA is commonly used as a dimension reduction technique where large datasets with redundant variables can be discarded without the loss of variation. Each of the resulting principal components will have differing contributions from different variables. The first principal component can be dominated by a few variables, and this can be the basis of how the analysis is interpreted. These variables can give an indication of what key combinations of variables account for a high proportion of total variance in the data. But because the data is orthogonally transformed onto a new coordinate system and because the values are scaled, the variables (principal components) do not explicitly represent the system-produced variables, hence applying PCA to the data set might cause it to lose interpretability. Nevertheless, PCA can be used to select those variables that contain the most information. King and Jackson (1999) did a comparative research study on the best method of variable selection using PCA, so that the reduced subset you are left with is as representative of the original dataset as possible. The results of the study were conclusive and they recommended that the B4 method worked best when complemented with the Broken-stick model criterion of number of variables to select (Jackson, 1993; King and Jackson, 1999). This will be described in Section 4.9.

2.6 Ethical issues

Despite overcoming many of the shortcomings of GPS, the use of CSD in any form has many ethical implications that need to be carefully considered before charging forward with the use of this type of data. Location data from mobile phones can reveal a lot about a person, not just where one works and lives, but also visits. Activities like participation in protests, or alcohol consumption based on frequency of being located in bars and pubs can be inferred and assumed of a user. This includes their schedules as well (Carter et al., 2015). Unsurprisingly, this is a major concern, especially when such information is made available to applications that serve third parties such as commercial companies (Calabrese et al., 2015). To combat this, regulations like the General Data Protection Regulation[5] have been put in place in May 2018, dictating that the data telecommunication companies release must be treated such that it was impossible to associate the location data with a cell phone number. Exceptions include cases whereby consent of the users who are tracked is explicitly conveyed. As such, researchers develop methods that reflect compliance and that abide by these regulations. Some of these attempts include location obsfucation, where locations are slightly altered but within the realms of being useful for services (Krumm, 2009). Another key change in the GDPR is the strengthening to privacy by design principles by making a legal requirement. Companies are now required to include data protection from the onset of designing a new system, rather than an additional feature at the end.

[5] The General Data Protection Regulation (GDPR) aims to protect all EU citizens from privacy and data breaches in an increasingly data-driven world. It was first established in 1995

However, despite the efforts made to deliberately encumber any form of matching of these trajectories to individual users, it can be argued that it is insufficient to truly protect users' privacies. While it is the predictability and repetitiveness of human behavior that make mobile phone data valuable in terms of mobility research, it is the very same set of traits that makes it difficult to completely anonymise this data. A recent study found that just 4 spatiotemporal observations are required to uniquely identify 95% of the individuals in their tests (de Montjoye et al., 2013). As such, moving forward, more complex techniques should be designed and implemented to protect individual privacy. Furthermore, there is also the issue of "group privacy" where people can be targeted on the basis of the social group they belong to. For example, certain groups may be represented more in mobile phone datasets depending on their age, gender, ethnicity etc. as indirect reasons for their level of mobile phone activity at particular times or particular places (Calabrese et al., 2015). Implications of being recognized as a result of identifying with a particular social group start to become a concern (Letouzé et al., 2015) .

We acknowledge that these constraints to individual privacy are important issues to consider. In the years to come, it is expected that the scrutiny on data mining and its associated privacy concerns will continue to increase. Users of this data must be sensitive to their methodologies and how legal privacy issues might impact them (Calabrese et al., 2015). With rising consumer concern, there might be legal challenges that this field runs into if these concerns are not adequately addressed with the implementation of more effective design frameworks devoted to privacy protection.

2.7 Summary and Research Gaps

A major motivation of pursuing this research is that CSD is a passive data type. When carrying out important urban movement analyses whose results will have an impact on decisions made by city and transport planners, this data must be as representative of the population in question as possible. Since cellular network data already exists and much of the population already carry personal mobile devices, there is no need to actively solicit respondents to complete surveys, or to track their movements on GPS loggers or GPS enabled devices. Even with GPS enabled mobile phones, the modules tend to be highly battery intensive due to high resolution of the data collected, which is extremely undesirable for respondents and proves to be a double edged sword. When the only available cellular network data was event-driven, i.e CDRs, the studies ran the risks of characterizing only highly active users (Calabrese et al., 2011). This became less of a problem when the penetration of smartphones increased and internet billing records were included. It was not until recently when companies started to release datasets that contained both event and network driven data that these cellular network datasets can be touted as being more representative of the population. With the inclusion of network driven data, i.e signaling data with periodic and location updates the sampling would be less dependent on a persons usage (Calabrese et al., 2015).

The main challenge of using CSD is handling the lower spatial and temporal resolutions, as well as the inconsistent and sometimes sparse samples of data. The previous section has shown how more and more researchers are trying to incorporate this data type into transportation science and LBS driven research, but those that use it primarily for mode detection methods are few and far between. Most of them use event driven data, and some use more current and less temporally sparse network driven data, though these constitute the minority. In addition, many of these studies do not have ground truth data to validate their studies. Due to privacy and ethical considerations, companies have made it difficult to track a mobile phone trace to a specific user. As this data type is relatively new in the mode detection realm, there has yet to be a data collection and labeling campaign to produce such benchmark ground truth data for cellular network data. Another gap in the body of work is that a lot of the studies only differentiate between stationary, walking and motorized modes, or at the most distinguishing motorized modes into general groups, car or public transport. Even when GPS data is used, it can be difficult to differentiate between cars and buses, especially in slow and congested traffic where speed and acceleration profiles can overlap a great deal. To alleviate this problem, contextual GIS information of public transport routes have been used to supplement the main cellular network data. While this concept has started to arise in studies using mobile phone data, they still lump public transportation modes into a single class (Qu et al., 2015).

To the best of our knowledge there has yet to be a method using cellular network data where distinctions are made for bicycles, trams, trains or buses. Supervised machine-learning methods using CSD data has also not yet been explored. As for methods that have been employed in CDR studies, the differences in CSD’s data characteristics mean that important values may not be transferrable, such as speed and distance thresholds in RBH methods. This also means that it is currently unclear as to which are the most important and useful features that can be extracted from CSD data as model inputs to distinguish between various modes of transportation. This thesis aims to address these gaps, to incorporate methods used in GPS studies and apply them to cellular signaling data provided by A1 (one of the main mobile network operators in Austria) for this thesis. For example, FL systems have a potential to be applied here as their fuzzy boundaries mean that they can handle more uncertainties in data. The method of building a FL system described above is especially applicable to transportation modes as the consequent part can be categorical.

While cellular signaling data is not as accurate as GPS data, it is a step up from the usual event driven data that has been used in mode detection studies with network data (CDR/GSM) in terms of spatial and temporal granularity. As such, the goal of this research is to propose and test new mode detection methods that are more well-matched and appropriate for CSD. Ultimately, this thesis looks at ways to infer a greater number of modes with higher accuracy than the current state of the art.

CHAPTER 3
DATA AND ITS CHARACTERISTICS

3.1 Cellular Signaling Data

The CSD used in this thesis is provided by A1 through two data collection campaigns. A1[6] is Austria's leading communications provider and has almost 6 million mobile users across the country. This works out to be 44.8% and 59.9% of the broadband and telephony market share in in Austria respectively[7]. The first data collection was done by Invenium Data Insights, a spin off company from the Technical University of Graz (TU Graz) that focuses on big data and mobility[8]. The second set of data came as a result of another ongoing research project in TU Graz investigating the distribution of urban activity using CSD. The study will only focus on the urban areas of Graz and Vienna. As such, trips to and from these cities will not be considered. This is to reduce complications that could be introduced by differences of data quality and granularity in urban and rural areas. Each CSD record consists of an anonymized ID, a timestamp and a triangulated spatial position. Special requests and procedures had to be followed in order to get clearance to link the raw mobile data to their phone numbers. The data providers do not reveal the full details of how localization is done through triangulation. In the first dataset, 2 volunteers from Invenium provided the data of their own mobile phones where their CSD was collected from mid Spetember to mid Novemeber. During this time they were mostly in the Austrian city of Graz, followed by Vienna. The second collection campaign was run by 9 volunteers from the 21st of March to 18th of April. The study area was mostly in Graz and wider Styria and the participants were encouraged to travel over different parts of this study area and to vary the modes of transportation taken. This time, GPS observations are also recorded with a mobile application. Each participant was given a smartphone provided by the researcher, and this came

[6] https://www.a1.net/
[7] https://cdn1.a1.group/final/en/media/pdf/pr-results-qu4-2017.pdf
[8] http://www.invenium.io/en/

with an application "Modalyzer" installed. This application enables the collection of GPS points each second, if there are enough GPS satellites, and automatically generates a trip diary (including time of day, travel time, travel distance and transport mode). Each recorded day is checked and corrected by the participants themselves, so that the verified data reflect ground truth as close as possible. Additionally they were asked to add the trip purpose in a "comments" field. The data annotation can be done in the app itself or on a corresponding website. The participants were also asked to use the smartphone actively in order to provide more raw CSD observations. All the data is characterized by the following definitions:

> ***Observation (o)****: Any event where the cell phone is communicating with a cell tower, and a data point is recorded. Each observation is a tuple (id, x, y, t), with a user id id, longitude x, latitude y, and timestamp t.*
>
> ***Trajectory (T)****: A sequence of observations of a single user, in chronological order. T = (o_1, o_2, o_3, ….o_n).*

In the first collection campaign, the observations were segmented into trajectories by Invenium and the two volunteers provided labels for 99 trajectories, 56 in Vienna and 43 in Graz by manually annotating the extracted trajectories from 3383 final observations. In the second collection campaign, the 9 volunteers generated 48 days worth of raw data in the cities of Graz and Vienna. This translates into 14802 raw observations over 920 hours.

3.1.1 Spatial resolution

Before analyzing the spatial and temporal resolution of the data, redundant points are first removed. If a series of points have the exact same geographic co-ordinates, the first and last points are kept while all points in between are removed. Also, duplicated observations (observations with the same geographic coordinates and timestamps) are removed. These redundant points do not offer any extra information and will give us a misleading indication of spatial and temporal resolution that is actually meaningful as these are based on mean and median values. Upon initial exploration of the data, the spatial resolution seems to vary considerably.

Figure 11 shows examples of two U-Bahn (metro) trajectories (trajectories where the user was taking the U-Bahn) that appear to have varying spatial resolutions. The pink trajectory seems to have observations that veer up to about 350m away from the nearest U-Bahn link (red tracks), while the blue trajectory seems to have a consistent proximity of less than 150m away from the nearest U-Bahn link. This pales in comparison to GPS data, which usually has a more consistent accuracy to about within 10m (Bohte and Maat, 2009), though this can also vary. To further explore the spatial accuracy of the CSD data, each CSD observation is matched with the corresponding GPS point by time to the nearest second (if available), and the distance between them is calculated. This will be used as a proxy to estimate the spatial resolution of the CSD data. Figure 14 shows the distribution of the spatial resolution of all the raw CSD points. For purposes of illustration, the range on the x-axis of the histogram below has been capped at 1.5km. The data has an extremely large range, with the maximum distance being over 67km away from the GPS point. This usually happens when the participant is commuting for long distances between cities, where a single LA may cover a large area, considerably larger than that of those in urban areas (Figure 15). It is observed that the distances calculated are extremely left skewed, with the median distance (273m) and the even the 3rd quartile (858m) being lower than the mean (1174m) (Table 2). This gives an indication of the general spatial accuracy of the data. In comparison, the GPS data collected in tandem with CSD data had a mean spatial accuracy of 25 m and a median of 10 m.

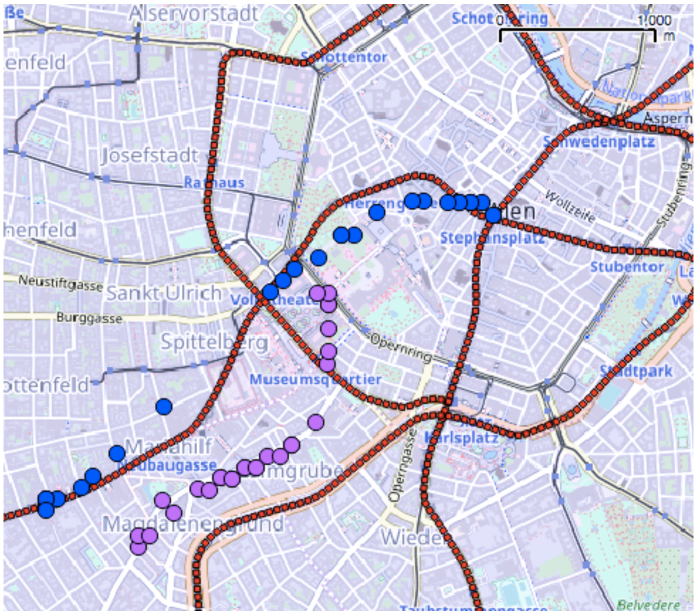


Figure 11: Examples of differing spatial resolution of observations, each color representing a different U-Bahn trajectory (Source: OpenStreetMap)

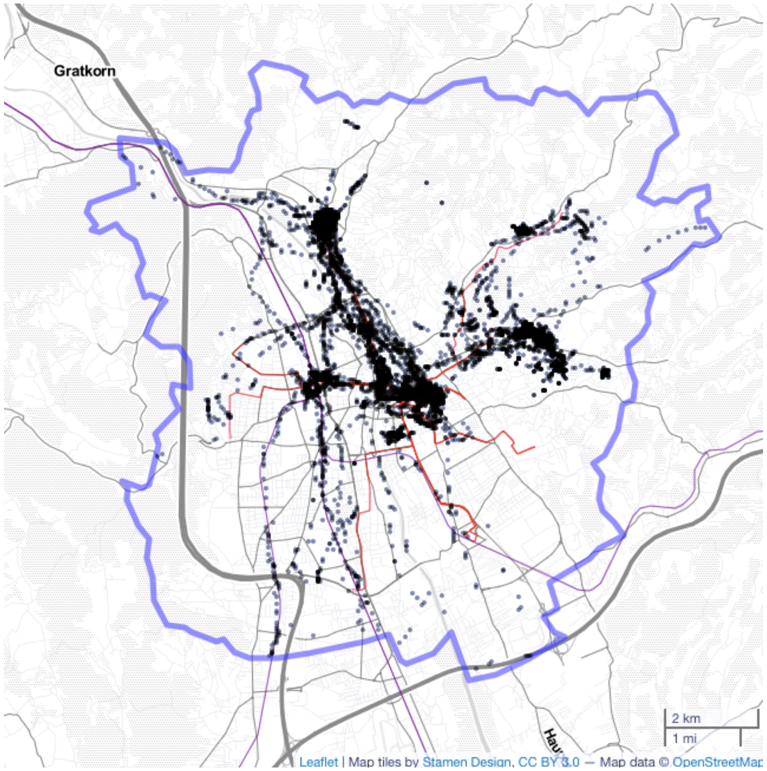

Gratkorn
Hau
2 km
1 mi
Leaflet | Map tiles by Stamen Design, CC BY 3.0 — Map data © OpenStreetMap

Figure 12 Visualization of CSD points in Graz (Source: OpenStreetMap)

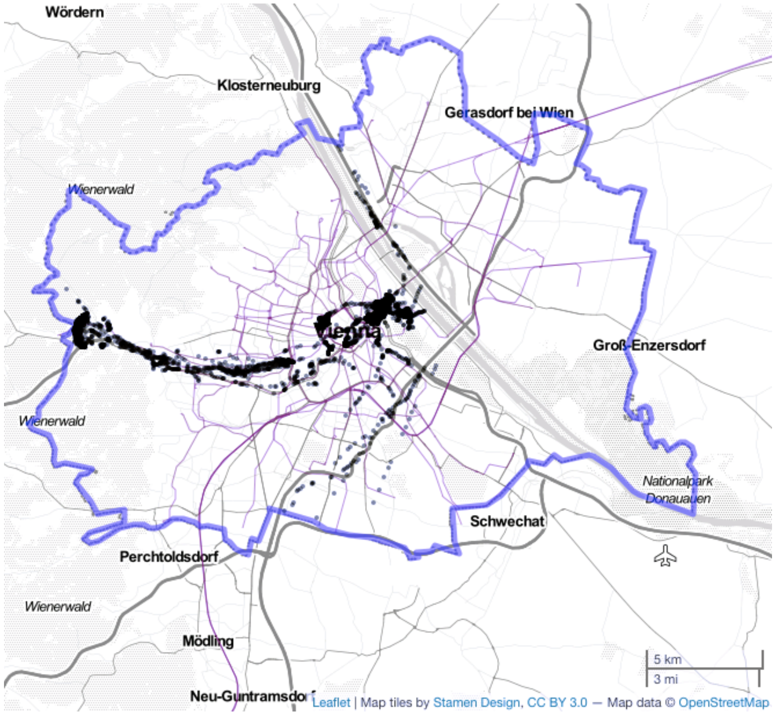


Figure 13 Visualization of CSD points in Vienna (Source: OpenStreetMap)

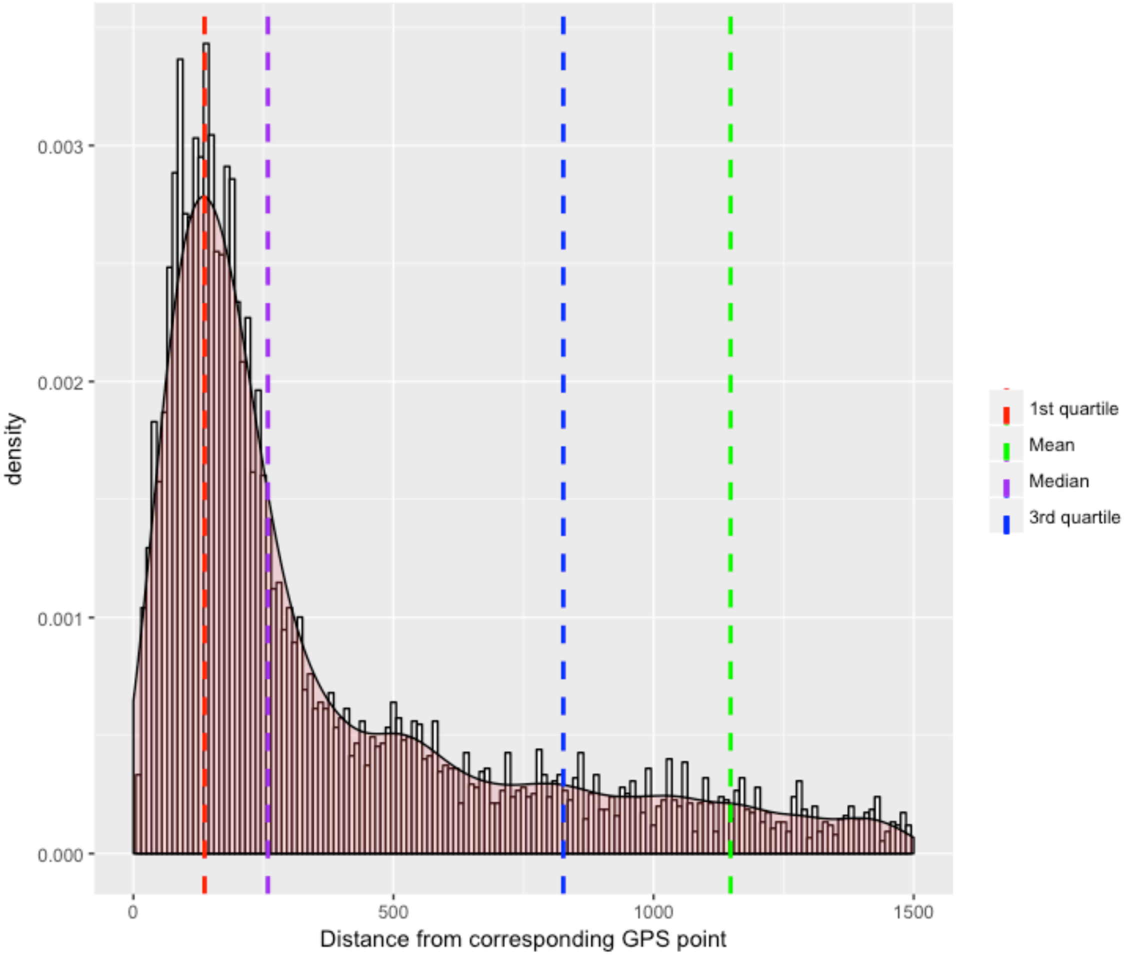


Figure 14 Distribution of raw CSD observations to corresponding GPS observations

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Raw (All points)	13.73	149.9	273.2	1174	858.6	67200
Raw (In cities)	13.73	120.5	188.8	332.1	324.5	64970

Table 2 Distance of raw CSD to corresponding GPS points in all points and in the urban study areas (in meters)

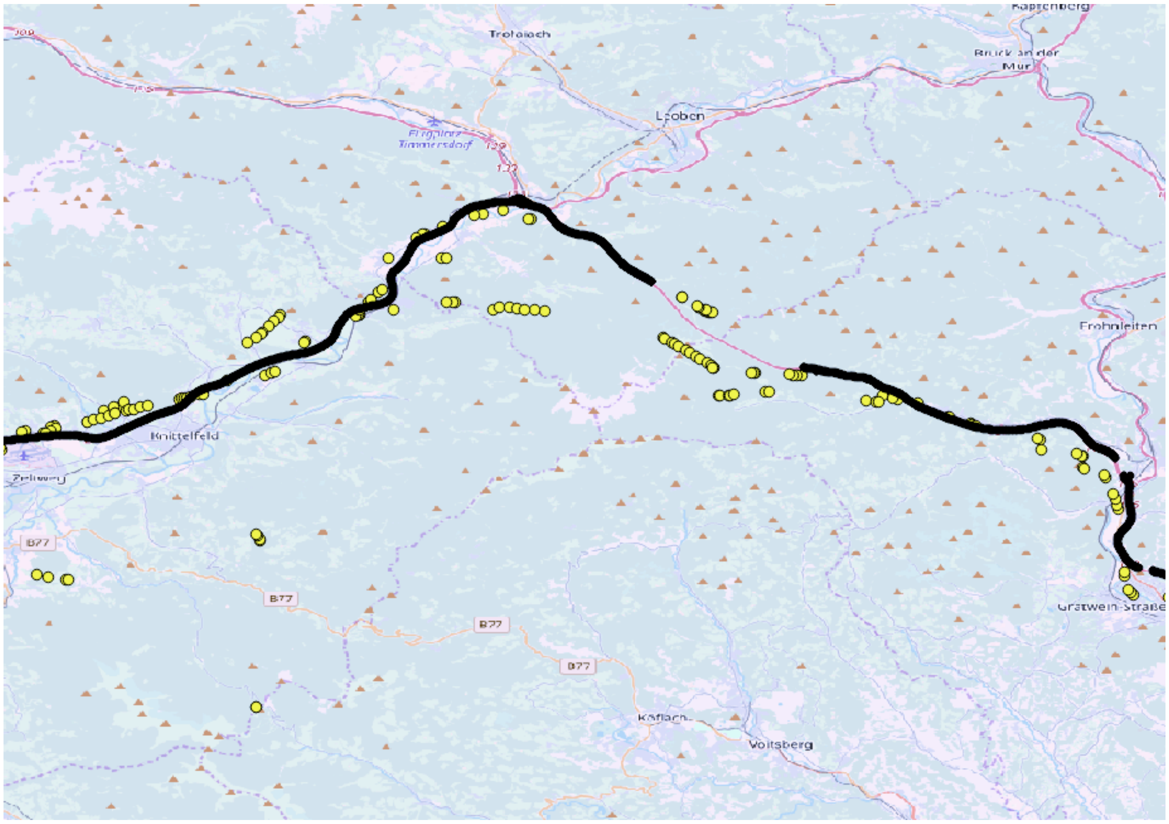


Figure 15 CSD (yellow) and corresponding GPS trajectory (black) when commuting between cities where cellular coverage is not as strong

Figure 16 shows the distribution of distances between CSD observations and the corresponding GPS points for CSD found in the cities of Graz and Vienna. While the maximum distance is still the same at almost 65km (not pictured), the mean (332.1 m) and median (188.8m) is now a lot lower (Table 2). However, due to the effects of outliers, the mean is still lower than the third quartile. In this particular case of the CSD found 65km away from the corresponding GPS point, the user was in an area where the user was travelling towards Graz from Vienna. Due to the ping-pong phenomenon or errors in the triangulation calculation, the recorded CSDs reflected a jump while on the Semmering expressway S6 a freeway to Graz for 2 observations, and then back again to the original location.

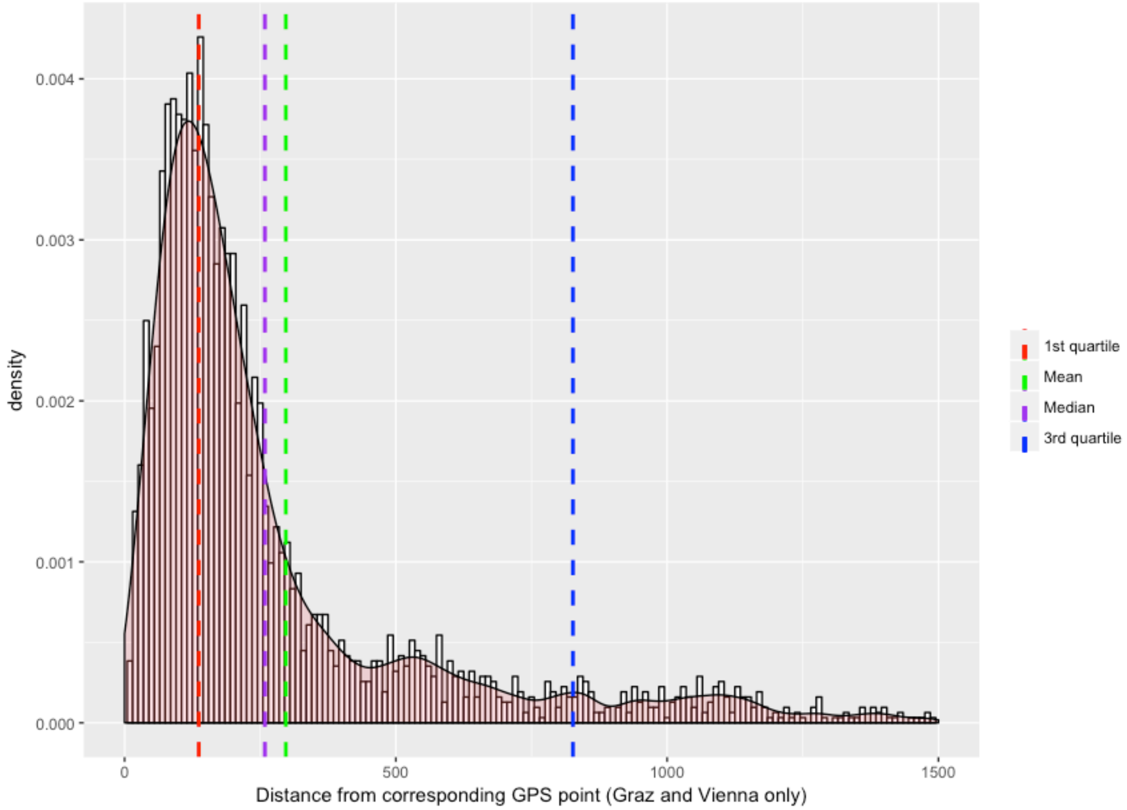


Figure 16 Distribution of raw CSD observations to corresponding GPS observations in Graz and Vienna.

3.1.2 Temporal resolution

The temporal resolution as expected, is highly irregular, with the shortest and longest time interval between consecutive observations to be less than 1 second and 5160 seconds respectively. The distribution of these time intervals is quite skewed towards the low end, with a median of 26 seconds and mean of 175 seconds. In a CDR study, Calabrese et al. (2011) used the mean of the medians as an indicator of the temporal resolution and the minimum duration of a stop that can be detected. He found that the arithmetic average of medians was 84 minutes, meaning that they could detect stops as low as about 1.5 hours.

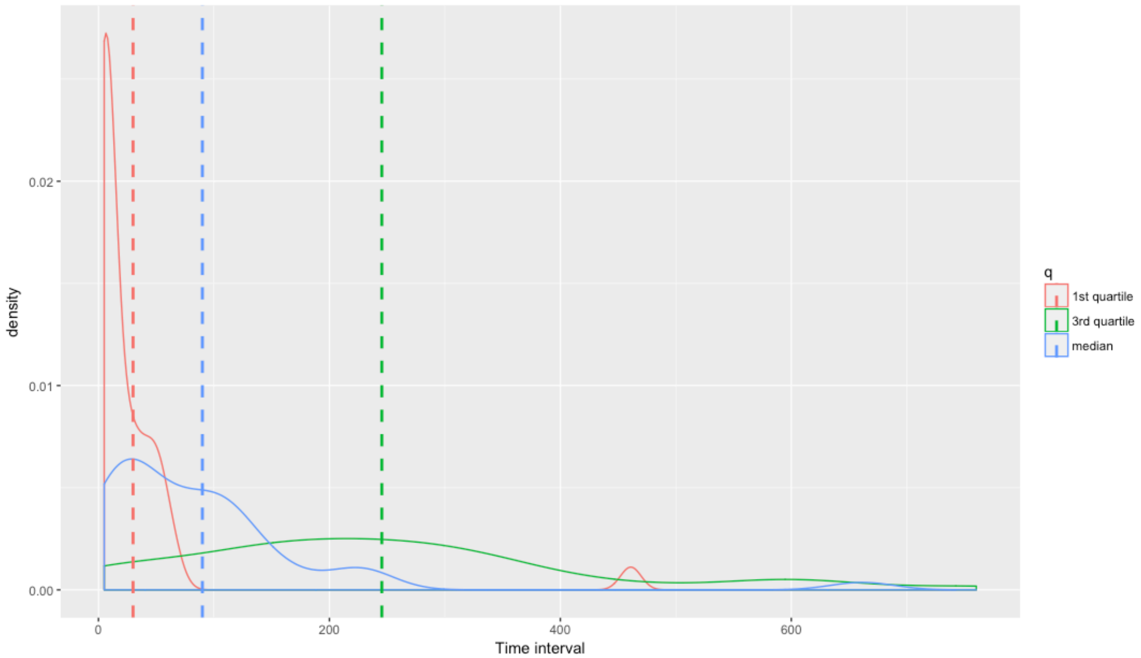


Figure 17 Distribution of the first quartiles, medians and third quartiles of the time intervals between observations of each user

The raw dataset has an arithmetic average of medians of 90 seconds, about 1.5 minutes. Like the spatial accuracy, the distribution of the time intervals is extremely left skewed, with 75% of the values being less than 110.50 seconds (3rd quartile) (Figure 17). However, as public transport modes do not generally stop at bus or tram stops for over a minute and going by Calabrese et al.'s (2011) interpretation of these values, it is unlikely that we will be able confidently to segment observations into trajectories in between stops. In contrast, the sampling rate of GPS data depends on the device that is collecting it, and this can be set differently for different purposes. For example, the GPS data collected in this study had a mean sampling rate of 1.97 seconds.

3.2 Other data

External GIS transportation network data for contextual information was also used. For Graz, these static geospatial data of the bus, tram and S-Bahn networks were obtained from

OpenStreetMap and for Vienna, U-Bahn, S-Bahn, bus and tram network shapefiles were taken from Vienna's Open Government Data website[9].

There are several important things to note of the collected data. Firstly, as the process of identifying trace of cellular signaling observations to a user is extremely difficult due to privacy obligations, we were only able to get two people from Invenium to be able to collect their cellular signaling data in the first collection run. Secondly, as these two volunteers had full-time day jobs, they went about their normal daily routines, meaning the observations followed the same route and had the same modes almost everyday as the volunteers travelled to and from work and home. Only when they were in Vienna, the modes and visited places were slightly more varied. This might also have implications on phone usage whilst travelling. As their home city was Graz, they would not need to use navigation apps for example, to reach their destinations. This might be different in Vienna, a city they might not be as familiar with. As such, the temporal resolutions might be generally higher in Vienna as the data is made of event driven data as well. This is in contrast to the second collection campaign where the data was actively sought over different areas and for different modes, where the participants were actively encouraged to use their phones more whilst travelling to increase the amount of CSDs generated.

[9] https://open.wien.gv.at/site/open-data/

CHAPTER 4
METHODOLOGY

4.1 Methodological procedure

Certain considerations were made prior to designing the mode detection algorithm:

1. S-Bahn and U-Bahn tracks are unlikely to run completely parallel to roads, especially in the city. As such, a trajectory with a consistently close proximity to these networks is highly likely to be of that particular mode (Figure 18). A trajectory's proximity to these networks can be measured in a few ways; as a percentage of points within a distance threshold to the network (Bohte & Maat, 2008), or the average distance of all points to the network (Qu et al, 2015).
2. Fastest modes can produce both low and high speeds, whereas non-motorized modes like walking can only produce low speeds, not taking outliers or data anomalies into account.
3. Slow modes like walk generate poorer quality data. Bicycles may tend to have a lower number of event driven data points (data generated when the phone is being used) as compared to a commuter on a tram for example, and network driven data points (data generated during location area changes) if trip distance is low (Figure 19).
4. As start and end points accuracy unclear, it might problematic to use the first and last points as the true start and end point.
5. Percentile values of speed or acceleration may be more meaningful measures to use than merely average values.
6. Differentiating between very slow cars, bikes and walking might be problematic as they can have almost identical motion profiles. This can be true for cars and buses in traffic jams as well, for example. As such, simple rules based on speed may not be sufficient. While a possibility is using real time bus locations, which is in line with the "smart city" goal that many places are aiming for, it still remains unavailable for many places, including Graz. Using the scheduled time is also

problematic due to the not uncommon delays faced by any public transport system, due to unforeseen circumstances like accidents, construction, or just heavy traffic in general.

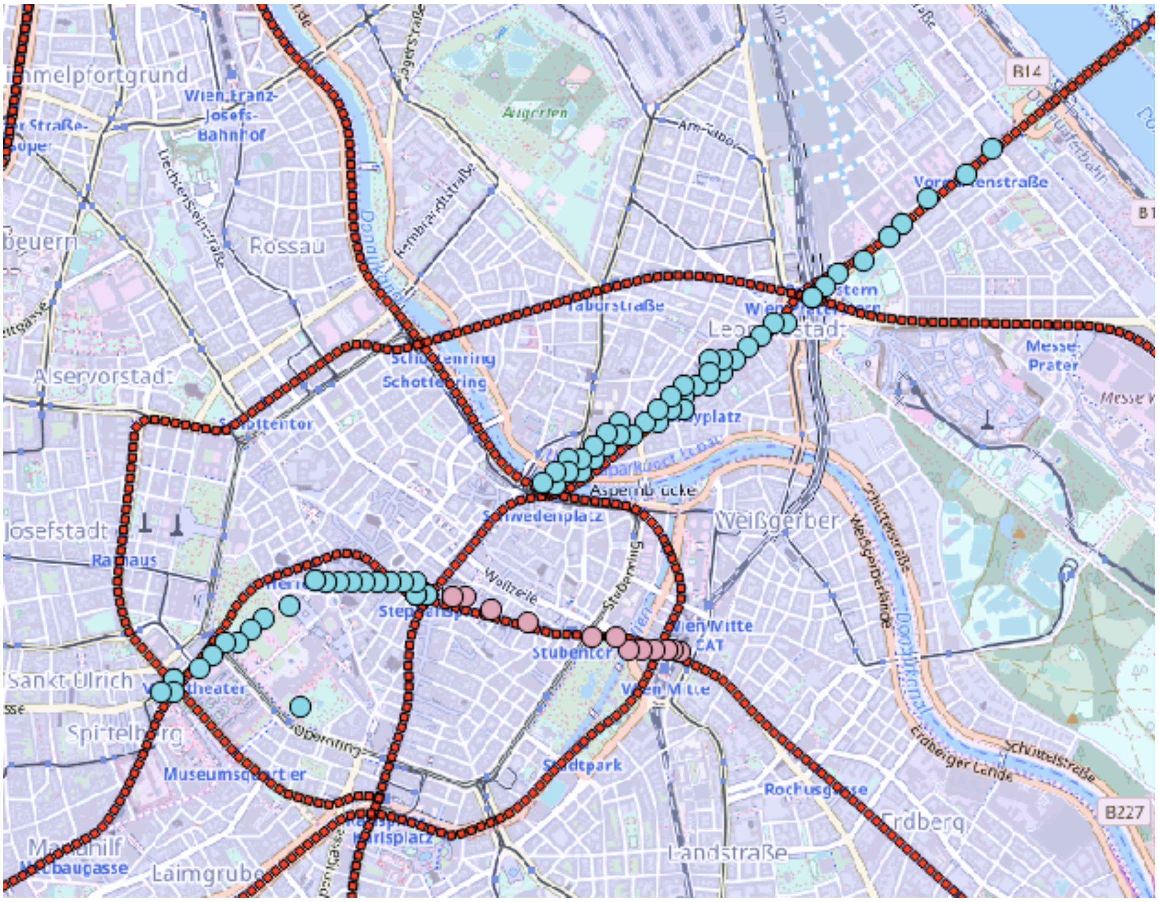


Figure 18 Example of two U-Bahn trajectories. The points are individual observations of the user taking the U-Bahn. Blue circles represent a trip in the morning and pink circles represent trips in the afternoon. Red tracks symbolize the U-Bahn network. (Source: OpenStreetMap)

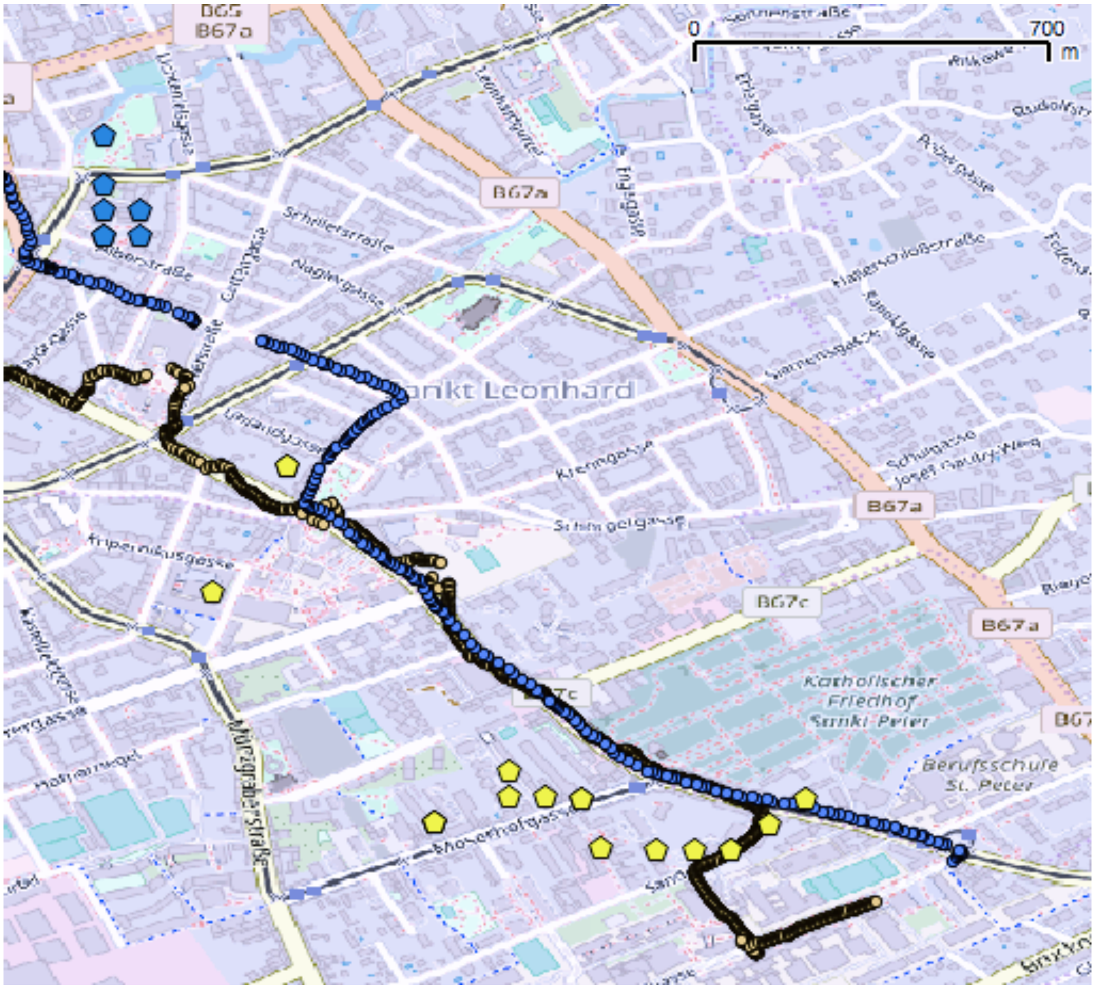


Figure 19: Example of CSD (pentagons) generated for walk (yellow) and bike (blue) compared to their corresponding GPS tracks (circles). (Source: OpenStreetMap)

With these considerations in mind, there are several approaches that will be described in this section. For both supervised and unsupervised approaches, they can be split into two main groups Table 3.

For group A, trajectories are subjected to a set of rules that allow us to filter out some more easily identified modes such as U-Bahn and S-Bahn. As such, the output is a set of determined and undetermined trajectories. The secondary steps in this combined mode detection method are either with a Fuzzy Logic (FL) System or with Random Forest (RF). Variable selection will also be carried out for the secondary mode detection step. This is done three ways:

1. Existing Literature in GPS-based mode detection (Schussler et al., 2009; Ramussen et al., 2015) *95th percentile acceleration, 95th percentile speed, median* speed

2. Machine Learning: Random Forest (RF). This method of supervised machine learning was picked, as it has been known to perform well in similar studies as the main mode detection step. However, in this case, RF is used as a dimension reduction technique so as to obtain an informative but small set of variables to be fed into the FL system. As the algorithm outputs variables that are of high importance, it can be an indication of best variables to use in the FL system.

Supervised A: Combined method of Rule-Based Heuristics (RBH) + secondary mode detection step (Fuzzy Logic / Machine Learning)	
RB_FLEL	RBH for U/S-Bahn and car + Fuzzy Logic with variables from existing literature in GPS
RB_FLRF	RBH for U/S-Bahn and car + Fuzzy Logic with Random Forest for variable selection
RB_RF	RBH for U/S-Bahn and car + Random Forest for classification
Supervised B: Fuzzy Logic or machine learning	
FLEL	Fuzzy Logic with existing literature in GPS
FLRF	Fuzzy Logic with Random Forest as variable selection
RF	Random Forest for classification
Unsupervised: Combined method of Rule-Based Heuristics (RBH) + secondary mode detection step (K-means or Paritioning around Medoids)	
RB_KMEANS	for U/S-Bahn and car + unsupervised K-means clustering
RB_PAM	RBH for U/S-Bahn and car + Partitioning around Medoids with Random Forest proximity matrix
PCA	Principal Component Analysis, the unsupervised variable selection method

Table 3 List of mode detection methods proposed and their abbreviations

If labels are not available, unsupervised methods will be used, and the two methods will be Partitioning around Medoids using the RF algorithm to generate the dissimilarity matrix, and K-means clustering. Principal Component Analysis (PCA) will be used to select the input variables. As mentioned earlier, the characteristics of cellular signaling data are still markedly different from that of GPS data, and as such there may be other variables that are more suitable as input fuzzy variables. Each PC will have a set of variables that contribute the most significantly to that PC, thus giving us an idea of which variables are more informative. With this in mind, we use the high-contributing variables of the leading PCs as inputs for the clustering. PCA was chosen as it is a popular dimension reduction technique, and as it is especially useful here in that it does not require labels.

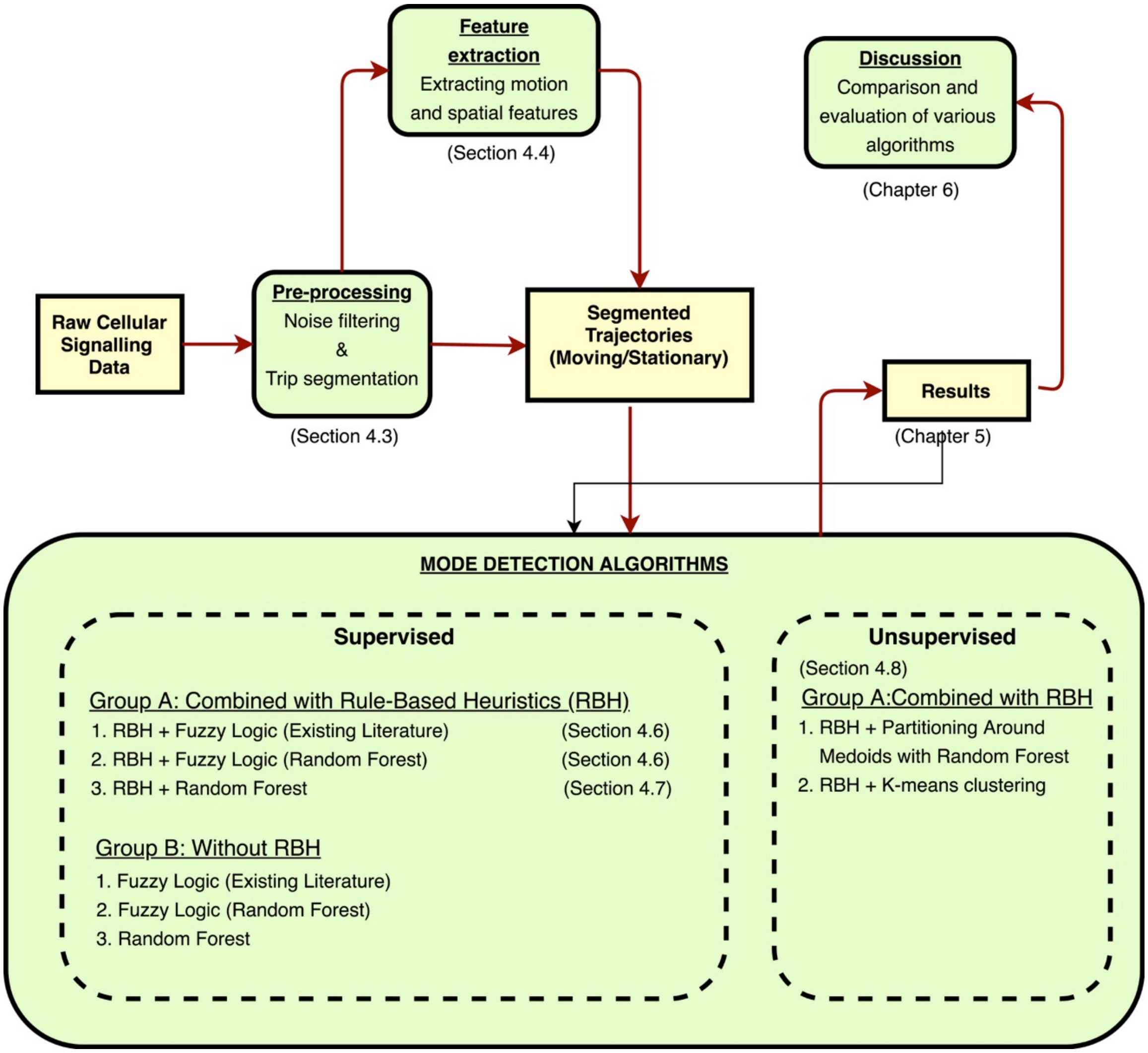

Feature extraction
Extracting motion and spatial features
(Section 4.4)
Discussion
Comparison and evaluation of various algorithms
(Chapter 6)
Raw Cellular Signalling Data
Pre-processing
Noise filtering
&
Trip segmentation
(Section 4.3)
Segmented Trajectories (Moving/Stationary)
Results
(Chapter 5)
MODE DETECTION ALGORITHMS
Supervised
Group A: Combined with Rule-Based Heuristics (RBH)
1. RBH + Fuzzy Logic (Existing Literature) (Section 4.6)
2. RBH + Fuzzy Logic (Random Forest) (Section 4.6)
3. RBH + Random Forest (Section 4.7)
Group B: Without RBH
1. Fuzzy Logic (Existing Literature)
2. Fuzzy Logic (Random Forest)
3. Random Forest
Unsupervised
(Section 4.8)
Group A:Combined with RBH
1. RBH + Partitioning Around Medoids with Random Forest
2. RBH + K-means clustering

Figure 20: Methodological Framework

The methodological framework of this thesis is summarized in Figure 20. First, in section 4.2, we describe the computing environment. Section 4.3 summarizes how it was cleaned and pre-processed for the purposes of this study. Section 4.4 describes an initial set of features that are extracted from the pre-processed data; these features will be then used in the mode detection algorithms. The next Section 4.5 outlines the initial and final designs of the RBH that is the first step of all methods in Group A. Here, rationales behind the inclusion or exclusion of certain rules and their threshold values are discussed. Following that is the description of the methods that precedes the RBH, specifically, FL systems in Section 4.6 and classification with Random Forest in Section 4.7. Section 4.8 explores an unsupervised method for situations when only unlabeled data is available. As these various methods require some form of variable selection step to reduce the number of features used, the next Section 4.9 outlines the two variable selection methods chosen here, with random forest that requires labels and Principal Component Analysis that does not.

4.2 Computing environment

The data was visualised in QGIS and R was used on RStudio as the analytical environment where the algorithms were coded and run. The table below shows an overview of the most important R-packages used in this thesis.

4.3 Pre-processing

The data provided by Invenium was already pre-processed and segmented into alternating stationary and moving trajectories. First, the outliers were detected and removed using Horn et al.'s recursive look-ahead filter (Horn et al., 2014). The recursive filter first detects outliers, and then looks-ahead to the subsequent observations to decide which of these observations are the outlier and which has the correct measurements. First, the speed between two successive events o_{i-1} and o_i is calculated and if this speed exceeds $v_{supersonic}$, o_i is flagged as a potential outlier. The look-ahead portion of the filter calculates the distance between the o_i and $o_{i+1,}$ $D_{i,i+1}$, and the distance between o_{i+1} and $o_{i-1}, D_{i+1,i-1}$. If $D_{i,i+1}$ is larger than $D_{i+1,i-1}, o_i$, is considered the outlier and removed. If not, o_{i-1} is removed. The filter can be described in the following algorithm (Horn et al., 2014), where the input for the filter is a sequential list of observations. $V_{supersonic}$ is set at 260km/h, about twice the maximum speed limit of an Austrian highway. For a sequence of observations, *S:*

```
Sort $S$ by time of events
    For $i$=0 to $S$• size:
    If $i$ > 0 And $i$ < $S$.size − 1:
        $v$ = distance($O_{i-1}$ • position, $O_i$ • position)/($O_i$ • time – $O_{i-1}$• time)
        If $v$ > $V$supersonic:
            $d$1 = distance($O_{i+1}$ • position, $O_i$ • position)
            $d$2 = distance($O_{i+1}$ • position, $O_{i-1}$ • position)
            If $d$1 > $d$2:
                Remove $O_i$
            Else:
                Remove $O_{i-1}$ End if
            End if
        End if
    End for
Return $S$
```

Package	Purpose	Reference
tmaptools	Reading in GPX data	(Tennekes, 2018)
dplyr	Data manimulation	(Wickham et al., 2017)
rgdal	Spatial manipulation and projections	(Bivand et al., 2018)
rgeos	Spatial calculations	(Bivand et al., 2017)
sp	Handling spatial objects	(Pebesma et al., 2018)
flexclust	k-means clustering	(Leisch and Dimitriadou, 2018)
cluster	PAM clustering	(Maechler et al., 2018)
frbs	Fuzzy Logic methods	(Riza et al., 2015)
randomForest	Random Forest methods	(Cutler and Wiener, 2018)
Boruta	Variable selection with RF	(Kursa and Rudnicki, 2018)
rpart	Tree-based classification	(Therneau et al.,2018)
rpart.plot	Visualisation of tree-based classification results	(Milborrow, 2017)
ggplot2	Visualisation of data	(Wickham et al., 2016)

Table 4 Overview of main R-packages used

Table 6 shows the list of features that were extracted from each observation. Segmentation will be done using the values of these features. The method is similar to distance and speed-based clustering, as explained in section 2.3.1 and 2.4.4, and can be described as a split and merge approach. After the observations have been pre-processed and the outliers removed, the sequences are then split into individual trip segments, alternating between moving and stationary trajectories. A low speed threshold of 3km/h is used to differentiate between stationary and moving observations. If the *inst.vel* (speed between two consecutive observations o_i and o_{i+1}) is below 3km/h, o_i is initially labeled with a state of “Stationary”, otherwise the state will be recorded as “Moving”. Consecutive stationary observations within a certain distance, *maxD*, are merged into a stationary segment. *maxD* is set at 500m. The algorithm consecutively examines the observations chronologically and incrementally creates and appends observations to clusters with small distances. The distance between a new stationary point and an existing stationary cluster is the average distance of that point to all points in already the cluster. Once the next stationary observation, O_{s1} exceeds *maxD*, a moving segment will be injected into the sequence at the position preceding O_{s1}. The moving segment will be made of $O_{s1\text{-}1}$ and O_{s1} as the start and end points of the segment respectively. In other words, clusters of stationary observations are formed sequentially so long as the distance between the points do not exceed *maxD*. Once *maxD* is exceeded, a new cluster is formed and a moving segment is inserted between the two clusters. Now the sequence of observations consists of groups with stationary and moving states, representing alternating stationary and moving segments. Next, in order to merge stationary segments within a certain range, centroids of the stationary segments are calculated. If centroids of stationary segments S_1 and S_2 are less than *maxD*, the moving segment between them, M_1, is treated as a stationary segment, and S_1, M_1 and S_2 are merged into a single stationary segment. The start and end points of all moving segments are the last and first points of the previous and next stationary segments respectively.

Prior to the pre-processing and segmentation, each CSD observation is assigned a label if its time stamp falls within the start and end time of a trip reported in the ground truth data. After the segmentation is done, only moving segments that have a single label covering over 80% of the duration of the segment will be considered. The rest will be considered non-trips. After removing the outliers, the subsequent distribution of spatial resolution measured by distances to GPS points of the remaining points (just within the study areas of Vienna and Graz) is shown in Table 5.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Raw (In cities)	13.73	120.5	190.6	332.1	324.5	64970
Outliers removed (In cities)	14.65	119.2	188.8	314.6	332	4016

Table 5: Distance of GPS points to of CSD points in metres

4.4 Feature list

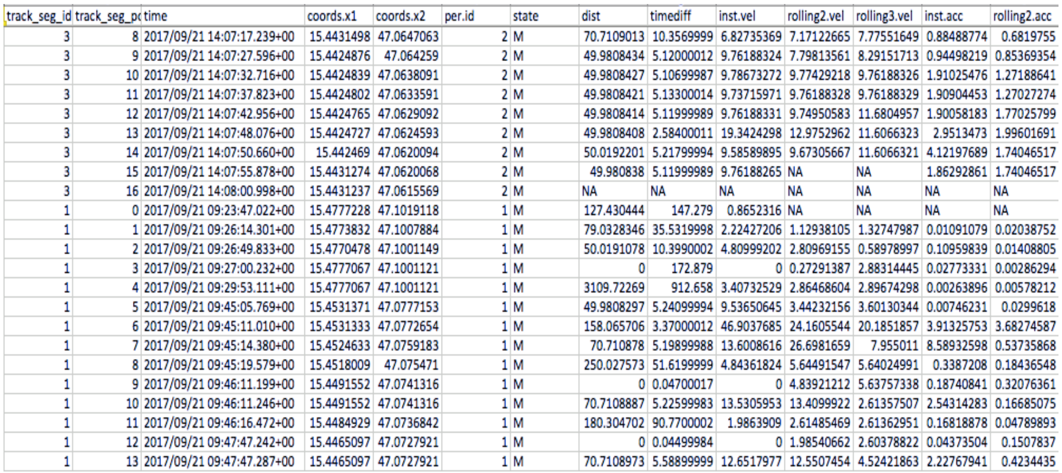
The next section describes the list of features that are extracted from the moving segments, which will be referred to as trajectories from now on. Each trajectory is made of a sequence of observations.

4.4.1 Features at the observation level

The following table describes the features extracted from each observation. Each feature extracted is for the observation o_i.

Attribute	Description
per_id	ID representing the user. Data generated by each user within a 24 hour window will have this ID
time	Date and time each point was generated [YYYY/MM/DD HH:MM:SS.MSMSMS]
coords.x1	Longitude
coords.x2	Latitude
track_seg_id	Segment ID of trajectory
state	State of trajectory, either M (Moving) or S (Stationary)
track_seg_point_id	ID of observations that make up each trajectory
dist	Great circle between observation o_i and o_{i+1}
timediff	Time elapsed between observation o_i and o_{i+1}
inst.vel	Velocity of o_i is calculated by $dist_{i,i+1}/timediff_{i, i+1}$
rolling2.vel	To smooth out measurement errors such as when huge jumps are made between observations, the velocity within a rolling window is calculated. Average velocity of 2 consecutive observations calculated by $(dist_{i,i+1} + dist_{i+1,i+2})/timediff_{i,i+2}$.
rolling3.vel	Average velocity of 3 consecutive observations calculated by $(dist_{i,i+1} +dist_{i+1,i+2} +dist_{i+2,i+3})/timediff_{i,i+3}$
Inst.acc	Acceleration of o_i calculated by $(inst.vel_{i,i+1} – inst.vel_{i+1,i+2})/timediff_{i,i+2}$
Rolling3.acc	Acceleration of o_i calculated using *rolling2.vel values*

Table 6: List of attributes extracted from each observation



track_seg_id	track_seg_pc	time	coords.x1	coords.x2	per.id	state	dist	timediff	inst.vel	rolling2.vel	rolling3.vel	inst.acc	rolling2.acc
3	8	2017/09/21 14:07:17.239+00	15.4431498	47.0647063	2	M	70.7109013	10.3569999	6.82735369	7.17122665	7.77551649	0.88488774	0.6819755
3	9	2017/09/21 14:07:27.596+00	15.4424876	47.064259	2	M	49.9808434	5.12000012	9.76188324	7.79813561	8.29151713	0.94498219	0.85369354
3	10	2017/09/21 14:07:32.716+00	15.4424839	47.0638091	2	M	49.9808427	5.10699987	9.78673272	9.77429218	9.76188326	1.91025476	1.27188641
3	11	2017/09/21 14:07:37.823+00	15.4424802	47.0633591	2	M	49.9808421	5.13300014	9.73715971	9.76188328	9.76188329	1.90904453	1.27027274
3	12	2017/09/21 14:07:42.956+00	15.4424765	47.0629092	2	M	49.9808414	5.11999989	9.76188331	9.74950583	11.6804957	1.90058183	1.77025799
3	13	2017/09/21 14:07:48.076+00	15.4424727	47.0624593	2	M	49.9808408	2.58400011	19.3424298	12.9752962	11.6066323	2.9513473	1.99601691
3	14	2017/09/21 14:07:50.660+00	15.442469	47.0620094	2	M	50.0192201	5.21799994	9.58589895	9.67305667	11.6066321	4.12197689	1.74046517
3	15	2017/09/21 14:07:55.878+00	15.4431274	47.0620068	2	M	49.980838	5.11999989	9.76188265	NA	NA	1.86292861	1.74046517
3	16	2017/09/21 14:08:00.998+00	15.4431237	47.0615569	2	M	NA	NA	NA	NA	NA	NA	NA
1	0	2017/09/21 09:23:47.022+00	15.4777228	47.1019118	1	M	127.430444	147.279	0.8652316	NA	NA	NA	NA
1	1	2017/09/21 09:26:14.301+00	15.4773832	47.1007884	1	M	79.0328346	35.5319998	2.22427206	1.12938105	1.32747987	0.01091079	0.02038752
1	2	2017/09/21 09:26:49.833+00	15.4770478	47.1001149	1	M	50.0191078	10.3990002	4.80999202	2.80969155	0.58978997	0.10959839	0.01408805
1	3	2017/09/21 09:27:00.232+00	15.4777067	47.1001121	1	M	0	172.879	0	0.27291387	2.88314445	0.02773331	0.00286294
1	4	2017/09/21 09:29:53.111+00	15.4777067	47.1001121	1	M	3109.72269	912.658	3.40732529	2.86468604	2.89674298	0.00263896	0.00578212
1	5	2017/09/21 09:45:05.769+00	15.4531371	47.0777153	1	M	49.9808297	5.24099994	9.53650645	3.44232156	3.60130344	0.00746231	0.0299618
1	6	2017/09/21 09:45:11.010+00	15.4531333	47.0772654	1	M	158.065706	3.37000012	46.9037685	24.1605544	20.1851857	3.91325753	3.68274587
1	7	2017/09/21 09:45:14.380+00	15.4524633	47.0759183	1	M	70.710878	5.19899988	13.6008616	26.6981659	7.955011	8.58932598	0.53735868
1	8	2017/09/21 09:45:19.579+00	15.4518009	47.075471	1	M	250.027573	51.6199999	4.84361824	5.64491547	5.64024991	0.3387208	0.18436548
1	9	2017/09/21 09:46:11.199+00	15.4491552	47.0741316	1	M	0	0.04700017	0	4.83921212	5.63757338	0.18740841	0.32076361
1	10	2017/09/21 09:46:11.246+00	15.4491552	47.0741316	1	M	70.7108887	5.22599983	13.5305953	13.4099922	2.61357507	2.54314283	0.16685075
1	11	2017/09/21 09:46:16.472+00	15.4484929	47.0736842	1	M	180.304702	90.7700002	1.9863909	2.61485469	2.61362951	0.16818878	0.04789893
1	12	2017/09/21 09:47:47.242+00	15.4465097	47.0727921	1	M	0	0.04499984	0	1.98540662	2.60378822	0.04373504	0.1507837
1	13	2017/09/21 09:47:47.287+00	15.4465097	47.0727921	1	M	70.7108973	5.58899999	12.6517977	12.5507454	4.52421863	2.22767941	0.4234435

Figure 21: Example of attributes extracted for each observation

4.4.2 Features at the trajectory level

From the segmented trajectories, each feature that will be used in the mode detection step is extracted. As mentioned in the literature review, common features extracted for the purposes of mode detection include, length of trip, maximum, average and median speed and acceleration (or 95th percentile speed for example), distance to transport lines, average heading change, whether trajectory intersects a pedestrian-only area (Gong et al., 2012; Gonzalez et al., 2010; Qu et al., 2015; Stenneth et al., 2011), to name a few.

Spatial features

Comparing the points with the existing transportation network, proximity values can be derived. The network data of Vienna is taken from the official government-run open data portal of Vienna[10]. Network data of Graz is extracted from OSM. Average distances are calculated as the average distance of each point in each trajectory to the closest link in the relevant transport network.

[10] https://open.wien.gv.at/site/open-data/

Spatial features	
Features	**Description**
Tramdist	Average distance to tram network
Busdist	Average distance to bus network
Sbahndist	Average distance to S-Bahn network
Ubahndist	Average distance to U-Bahn network (N.A. for Graz)

Table 7: List of spatial features extracted for each trajectory

Motion features

These features extracted are meant to describe the movement of the user during the course of the trajectory. As each trajectory is made of a number of observations, these features are usually averages of all the observations within a trajectory. For example, *vel.inst (*Table 7) of the trajectory is the average of all *inst.vel* values (Table 6) of the observations that make up the trajectory. Motion-based features include measures of speed, acceleration and their corresponding characteristics such as their deciles, mean, maximum and minimum etc. For these measures, to account for any more data anomalies that were not removed in the cleaning stage, values of rolling windows were also extracted. These consist of commonly used features in other mode detection studies using GPS as well as mobile phone data (Axhausen and Schüssler, 2009; Gong et al., 2012; Rasmussen et al., 2015) Other measures such as average speed of the entire trajectory, average distance between points and number of points were extracted, as there may be defined differences of these measures between each mode (Sohn et al., 2006). For example, one would expect the distance between points for someone on a high speed U-Bahn to be larger than that of someone who is strolling along a street. Similarly, it might be the case that someone on a bicycle might leave a lower number of data points than someone on a tram as they would not be able to surf the Internet whilst cycling, the same way a commuter and public transport can. Some other secondary features are also derived, such as the ratio of standard deviation to the mean (both speed and acceleration measures) to test if it can be a possible indicator that can differentiate between modes.

Some contextual data such as length of trip are also extracted as Wiener Linien has announced that the various modes have noticeably different average distance between stops. However, due to the temporal resolution (section 3.1.2), the segmentation may not reliably divide trips into individual journeys between successive stops. If they were, the ratio of stationary time to travel time could also be extracted. However, they can still be meaningful as a minimum distance travelled for a particular public transportation mode as the distance travelled should not be less than the distance between two stops. The list of features and descriptions can be found in Table 8.

Motion features	
Features	**Description**
duration	Duration of entire trajectory
Trip_dist	Distance traveled by all observations
vel.inst	Average inst.vel
vel.rolling2	Average of rolling2.vel
vel.rolling3	Average of rolling3.vel
Iqr.vel3	Inter-quartile range of rolling3.vel values
vel.var	Variance of instant velocity
vel.rolling2.var	Variance of rolling2.vel
vel.rolling3.var	Variance of rolling3.vel
vel.sd	Standard deviation of inst.vel
vel.rolling2.sd	Standard deviation of rolling2.vel
vel.rolling3.sd	Standard deviation of rolling3.vel
vel.median	Median of inst.vel
vel.rolling2.median	Median of rolling2.vel
vel.rolling3.median	Median of rolling3.vel
vel.sd.mean	Ratio of standard deviation to mean of rolling3.vel values (vel.rolling3.sd/vel.rolling3)
percentileXspeed	Xth percentile value of rolling3.vel
acc.inst	Average acceleration
acc.rolling2	Average rolling3.acc
Iqr.acc2	Inter-quartile range of acc values o
acc.var	Variance of inst.acc
acc.rolling2.var	Variance of rolling2.acc
acc.median	Median of inst.acc
acc.rolling2.var	Median of rolling2.acc
acc.sd	Standard deviation of inst.acc
acc.rolling2.sd	Standard deviation of rolling2.acc
acc.sd.mean	Ratio of standard deviation to mean of rolling3.acc values (acc.rolling2.sd/acc.rolling2)
percentileXacc	Xth percentile value of rolling2.acc
Abs_max_speed	Maximum inst.vel value
Rolling2_max_speed	Maximum rolling2.vel value
Rolling3_max_speed	Maximum rolling3.vel value
Total_speed	Speed of journey. Derived from trip_dist/duration

Eucdist_speed	Speed of journey as the crow flies. Derived from Euclidean distance between first and last point/ timediff between first and last point
Avg_dist	Average distance between each observation
Num_points	Number of points in the trajectory

Table 8 List of descriptive motion features extracted from each trajectory

4.5 Rule-based heuristics

For the mode detection methods of Group A, the primary step is a rule-based heuristic (RBH). The aim of this is to attempt to first identify modes based on human reasoning with simple rules of proximity to the transportation network and the assumption that certain modes produce certain speed values. A hierarchical set of rules was designed with an initial proximity to transport network split to identify rail features (S-Bahn/U-Bahn) with velocity measures. This is akin to Ramussen et al.'s paper (2015) and based on the rationale that because the rail usually runs independently of the road network, trajectories that have a lower average distance to these networks have a high probability of being a rail trajectory. Trip distance, acceleration, velocity and percentile velocity are used to filter out car trips based on the notion that while cars are able to generate low speeds, unlike larger modes like trams and buses, they are able to generate high speeds and acceleration as well. The remaining trajectories that do not have modes assigned are then fed into a FL system or Random Forest. The figure below shows the initial design of the rule-based heuristic (Figure 22). The design was initially based around thresholds and values obtained from existing literature as well as vehicle specifications from Wiener Linien, the public transport provider of Vienna. The motivation behind this was to keep the method less data driven, with the intention of increased generalizability. This would mean that other people would be able to adopt these methods for their own city's data with a few contextual inputs. The rules are first split by proximity to the public transport network, so as to ascertain if the user used public (tram/bus/U-Bahn/S-Bahn) or private (car/walk/bike) modes or transport. Studies use public transport proximity threshold values that range from 25m to 75m to 200m when using GPS data (Bohte and Maat, 2009; Gong et al., 2012; Rasmussen et al., 2015) to 500m when using CDR data (Qu et al., 2015). As the median distance to GPS data after outliers have been removed was found to be 188.8m (Table 5), the threshold here is set to 185m.

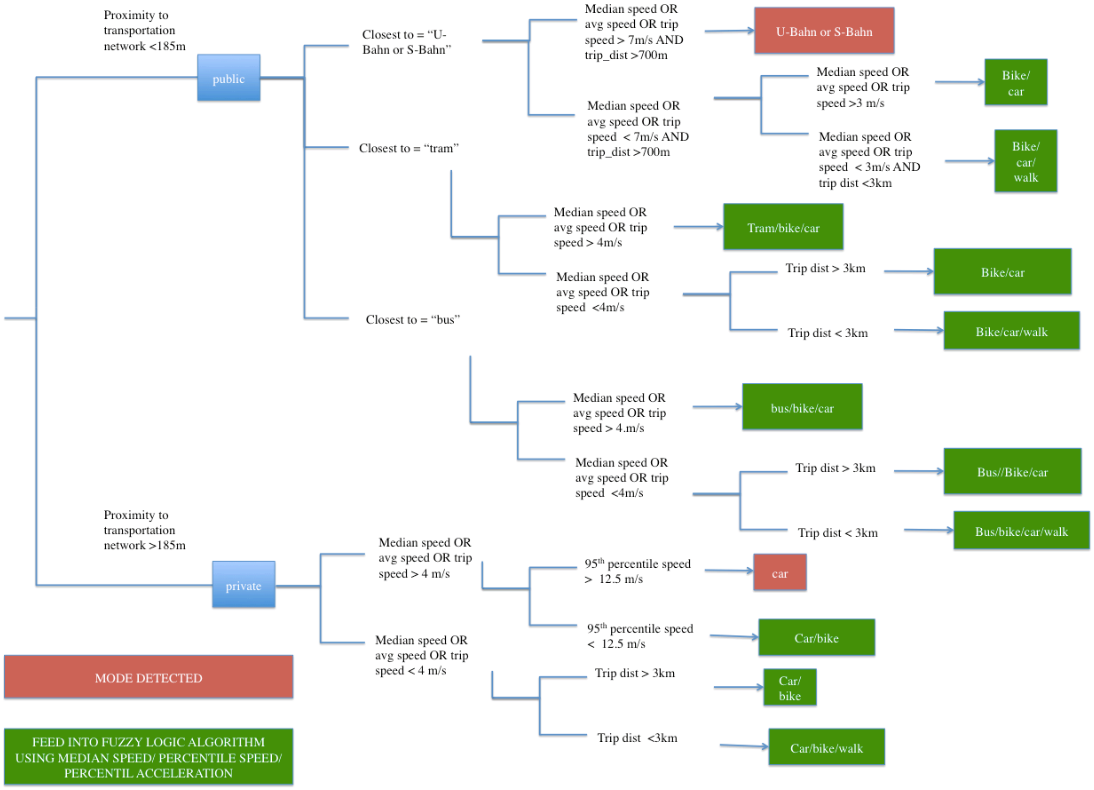

Proximity to transportation network <185m
public
Closest to = "U-Bahn or S-Bahn"
Median speed OR avg speed OR trip speed > 7m/s AND trip_dist >700m
U-Bahn or S-Bahn
Median speed OR avg speed OR trip speed < 7m/s AND trip_dist >700m
Median speed OR avg speed OR trip speed >3 m/s
Bike/ car
Median speed OR avg speed OR trip speed < 3m/s AND trip dist <3km
Bike/ car/ walk
Closest to = "tram"
Median speed OR avg speed OR trip speed > 4m/s
Tram/bike/car
Median speed OR avg speed OR trip speed <4m/s
Trip dist > 3km
Bike/car
Trip dist < 3km
Bike/car/walk
Closest to = "bus"
Median speed OR avg speed OR trip speed > 4.m/s
bus/bike/car
Median speed OR avg speed OR trip speed <4m/s
Trip dist > 3km
Bus//Bike/car
Trip dist < 3km
Bus/bike/car/walk
Proximity to transportation network >185m
private
Median speed OR avg speed OR trip speed > 4 m/s
95th percentile speed > 12.5 m/s
car
95th percentile speed < 12.5 m/s
Car/bike
Median speed OR avg speed OR trip speed < 4 m/s
Trip dist > 3km
Car/ bike
Trip dist <3km
Car/bike/walk
MODE DETECTED
FEED INTO FUZZY LOGIC ALGORITHM USING MEDIAN SPEED/ PERCENTILE SPEED/ PERCENTIL ACCELERATION

Figure 22: Version 1 of rule-based algorithm when all considerations as mentioned in section 4.1 are taken into account, and when concepts from other GPS studies using rule-based heuristics are borrowed

Just for illustration purposes, Figure 23 shows the cumulative distributive function plot of average distances of trajectories to the U-Bahn network of the various modes. The *trip_dist >* 700m rule is also introduced to supplement the proximity threshold, which is derived from the statistic provided from Wiener Lienien that the average distance between stops on the U-Bahn is 754m[11], as such, the rule removes trajectories that are shorter than this length as people cannot hop on and off between stops.

[11] https://www.wienerlinien.at/media/files/2017/facts_and_figures_2016_213708.pdf

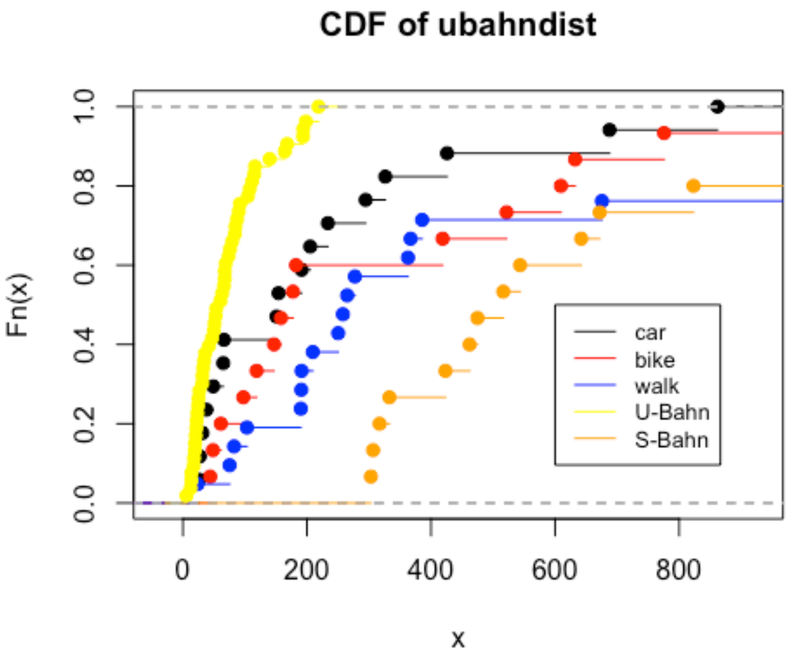


Figure 23: Cumulative distribution function of ubahndist. The diagram shows all U-Bahn trajectories have an average distance of less than 200m to the U-Bahn network for the Vienna dataset

The next split is using velocity, separating high-speed values from that of low speed ones, as high-speed values can be more informative due to the inherent fact that any mode of transportation can generate low speed values. This threshold value is derived from a publication by Wienier Lienien, which states the average speed of a U-Bahn journey is about 9m/s. Hence a conservative value of 7m/s was chosen as a minimum average to be considered as a U-Bahn mode. Two speed measures were used here, the median speed as well as the average speed of the entire trip. The same was done for the trams and buses as well, whose stated average speeds were between 4.1 m/s to 5.5 m/s, depending on the time of day. Again, a conservative estimate of 4 m/s was chosen as the minimum speed values to be considered for tram and bus modes. Those that do not meet this minimum are then split by total distance, with the assumption that people do not walk more than 3km to get to a destination if they could use another form of transportation. However, as taking the bus, riding a bicycle and walking all can be done at slow speeds (i.e. in congestion), and for a short distance (i.e. just one stop), trajectories with both very low average speeds and short distances cannot be assigned as a walk trajectory with high confidence. The green boxes in the Figure 22 represent the nodes where the choice of possible modes have been narrowed down but are still undetermined after all the relevant rules have been applied. This can still include U-Bahn and S-Bahn trajectories that were not captured by the initial rules due to larger inaccuracies in the data. Other studies with GPS data use proximity to start and stop points to filter out public transportation modes, but upon initial analysis of the segmented data, the accuracy of these points vary widely and thus are not used.

For trajectories that have an average distance exceeding 185m from the any transportation network, they are categorized as private modes. Similar to public modes, the next split is then a speed split, to sieve out non-motorized modes from motorized modes of private transport. The value of 4m/s is derived from the upper bound of walk threshold speeds used in existing studies. The highest is used by Bohte and Maat (2009), of 14 km/h or 3.83m/s. For those that exceed this value, car modes are singled out on the assumption that they are able to produce higher speeds that bicycles and walking cannot. Existing studies use maximum speeds for this. However, due to the noisy nature of this data type and its higher tendency to have outliers, a decile value is used. In this case, it is the 95th percentile speed. For those that do not exceed the 4 m/s threshold, the next split is on trip distance of 3km based on the assumption as described above. One potential area for improvement is in the initial velocity split, which could be done with percentile speeds

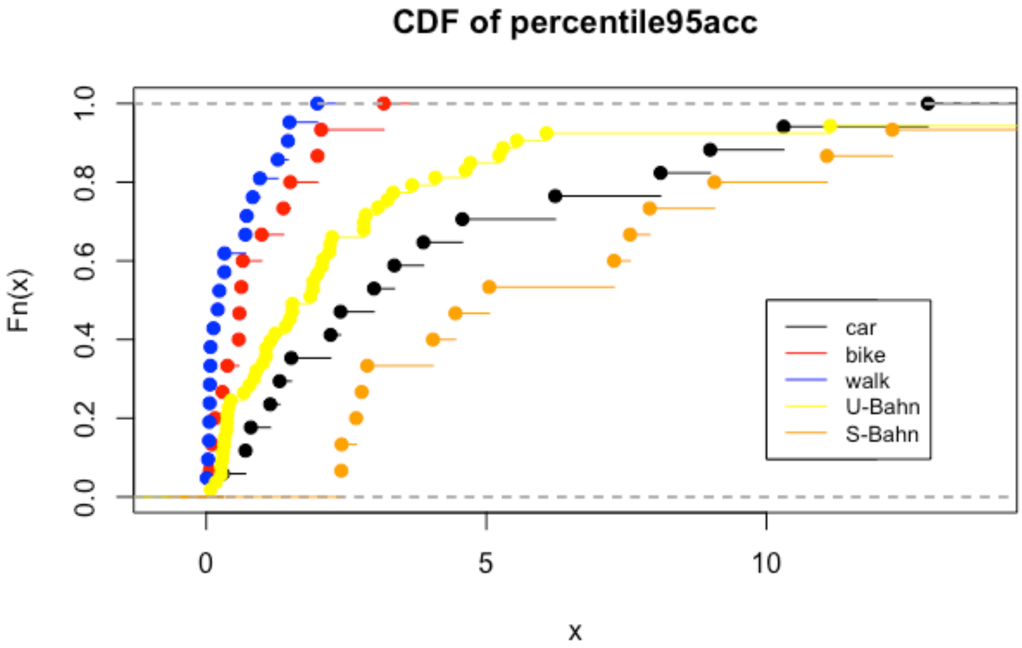


Figure 24 Cumulative distribution of *percentile95acc* of trajectories of various modes in Vienna

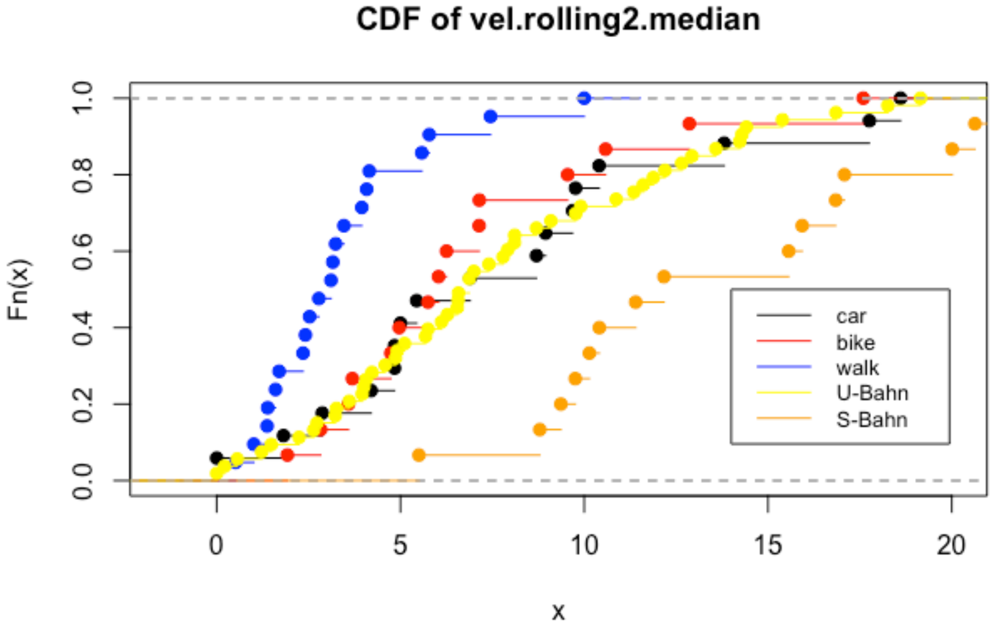


Figure 25 Cumulative distribution of *vel.rolling2.median* of trajectories of various modes in Vienna

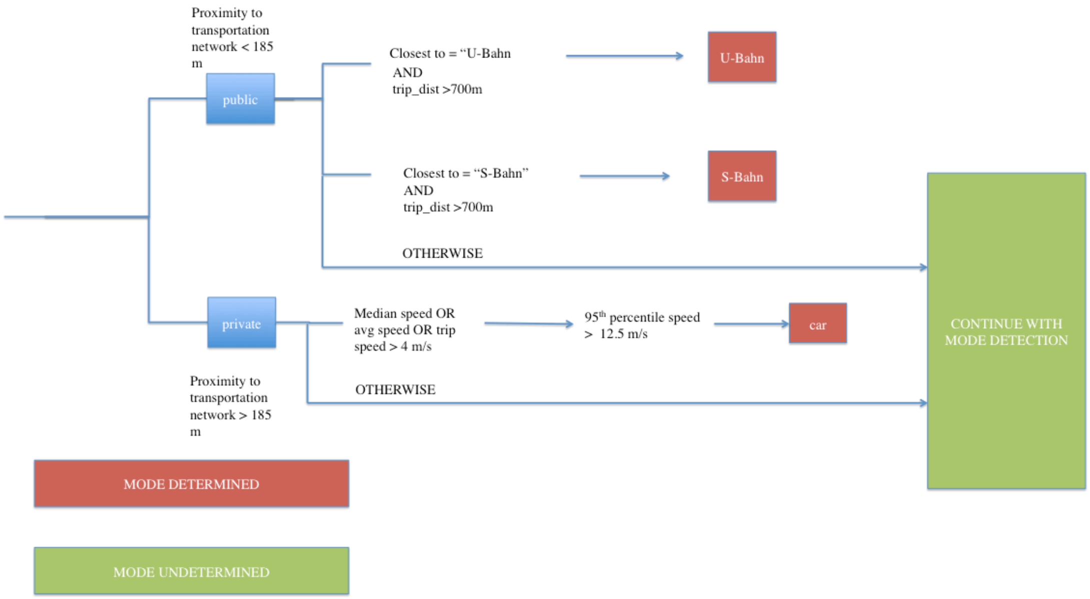

Proximity to transportation network < 185 m
public
Closest to = "U-Bahn
AND
trip_dist >700m
U-Bahn
Closest to = "S-Bahn"
AND
trip_dist >700m
S-Bahn
OTHERWISE
private
Median speed OR avg speed OR trip speed > 4 m/s
95th percentile speed > 12.5 m/s
car
OTHERWISE
Proximity to transportation network > 185 m
CONTINUE WITH MODE DETECTION
MODE DETERMINED
MODE UNDETERMINED

Figure 26 Second version of rule-based algorithm, simplified and improved

as supposed to average, so as to really tune in on high speed values. This is especially so for rail modes that are not at the mercy of road traffic. However, this might miss out rail trajectories that have a lower speed. On closer inspection to the speed and acceleration data, it is evident that the various modes do not have extremely clear distinctions, especially between car and bike for speed and bike and walk for acceleration (Figure 24 and Figure 25). Also, the initial assumption that the trains are high-speed modes, in that they do not produce low average speeds was not true when looking at the values extracted from the data. Even walk modes produced high average speed values, similar to the distribution of U-Bahns and cars. While there are some slight horizontal shifts in the functions, the largely similar distributions mean that single threshold values of speed and acceleration are difficult to implement into the set of rules. These insights were taken into account and the rules were improved by removing the speed thresholds (Figure 26).

A preliminary run of the data through the rule based showed that algorithm seems to perform similarly, if not better, without the added rules of speed thresholds. Both versions achieve the same proportion of correctly identified modes for those that are determined (are in the red boxes). However, the second version identifies more modes. This is because a lot of U-Bahn trajectories seem to produce speeds much lower than expected and very similar to other modes. This might be due to the stops that they make, reducing the recorded average speeds.

4.6 Fuzzy Logic Systems

One secondary step of the mode detection algorithm is the FL system. When compared with existing FL studies (Axhausen and Schüssler, 2009; Rasmussen et al., 2015), one way forward was to apply the data to the formulated rules and membership functions from existing literature (GPS studies). However, the velocity and acceleration values collected from the data were too different for them to be subjected to the same data ranges that were set by human experts in those studies (range of acceleration in GPS studies much lower than that in this study for example), possibly due to the differences in spatial and temporal granularity. Furthermore, as can be seen in the CDF plots of speed and acceleration, the differences are not as clearly defined to be incorporated into rules definitive rules, further making the case for the incorporation of an FL system. As such, a more suitable way to do this is to build our own rules and membership functions based on parameter values derived from the data. The *frbs* package in RStudio creates arbitrary membership functions with a user defined number of labels (5, very low, low, medium, high, very high for example), and subsequently assigns rules based on the data and these membership functions.

The FL system used to investigate the feasibility of this method is the fuzzy rule-based classification systems based on Chi's method to handle classification tasks (FRBCS.CHI). This is based on Wang and Mendel (1992)'s model that tackles classification problems. They approach the problem by creating the FL system through space partitioning in 4 main steps. The input variables will be used to define the output variable. Ultimately, the set of input variables will be chosen from the list of features extracted from the trajectories (Table 7 and Table 8).

1. For each input variable, the data is normalized and parameter values are arbitrarily set based on the normalized data (each value has the minimum subtracted from it and then divided by the range, resulting in a normalized range of 0 to 1). The input and output spaces (variables and labels) of the given numerical data are then divided into fuzzy regions, which refer to linguistic variables for each linguistic. For example, if there are 3 labels (low, medium, high), and the membership function is set as trapezoid, this will be drawn (Figure 27). The first row in Table 8 represents the label of each membership function, and the second onwards summarize the parameter values. For example, for the *total_speed* variable's "small" membership function, the left corner, upper left, upper right and right corner is not applicable, 0, 0,2 and 0,4 respectively. These are the normalized *total_speed* values (Table 9). The left and right top corners of the trapezoids are the range in which its degree of the membership function equals to 1. One assumption used here is that these spaces can be arbitrarily defined. Membership functions can be in the form of triangle, trapezoid, Gaussian etc.

2. Secondly, the IF-THEN rules are generated by the *frbs.learn* function. The fuzzy parameters from step 1 are used to partition the input-output space, which is then filled with the training data based on their values. The process is repeated for each observation in the training data to construct fuzzy rules covering the training data. Degrees of the membership function for all input and output pairs are calculated. Some examples of the rules generated for three of the modes are shown below:

a. *IF total_speed is small and percentile95acc is small and percentile95speed is small and vel.rolling2.median is small THEN label is 1. Degree = 1.00*

b. *IF total_speed is small and percentile95acc is small and percentile95speed is small and vel.rolling2.median is small THEN label is 3. Degree = 0.874*

c. *IF total_speed is large and percentile95acc is medium and percentile95speed is large and vel.rolling2.median is medium THEN label is 2. Degree = 0.786*

*1 = bike, 2 = car, 3 =walk

Fuzzy variables	total_speed			percentile95acc			percentile95speed			vel.rolling2.median		
MF* labels	S	M	L	S	M	L	S	M	L	S	M	L
MF* parameter *MF = Membership Function	0	0.23	0.6	0	0.23	0.6	0	0.23	0.6	0	0.23	0.6
	0.2	0.43	0.8	0.2	0.43	0.8	0.2	0.43	0.8	0.2	0.43	0.8
	0.4	0.53	1	0.4	0.53	1	0.4	0.53	1	0.4	0.53	1
	NA	0.73	NA	NA	0.73	NA	NA	0.73	NA	NA	0.73	NA

Table 9 Parameter values of membership functions of normalized data

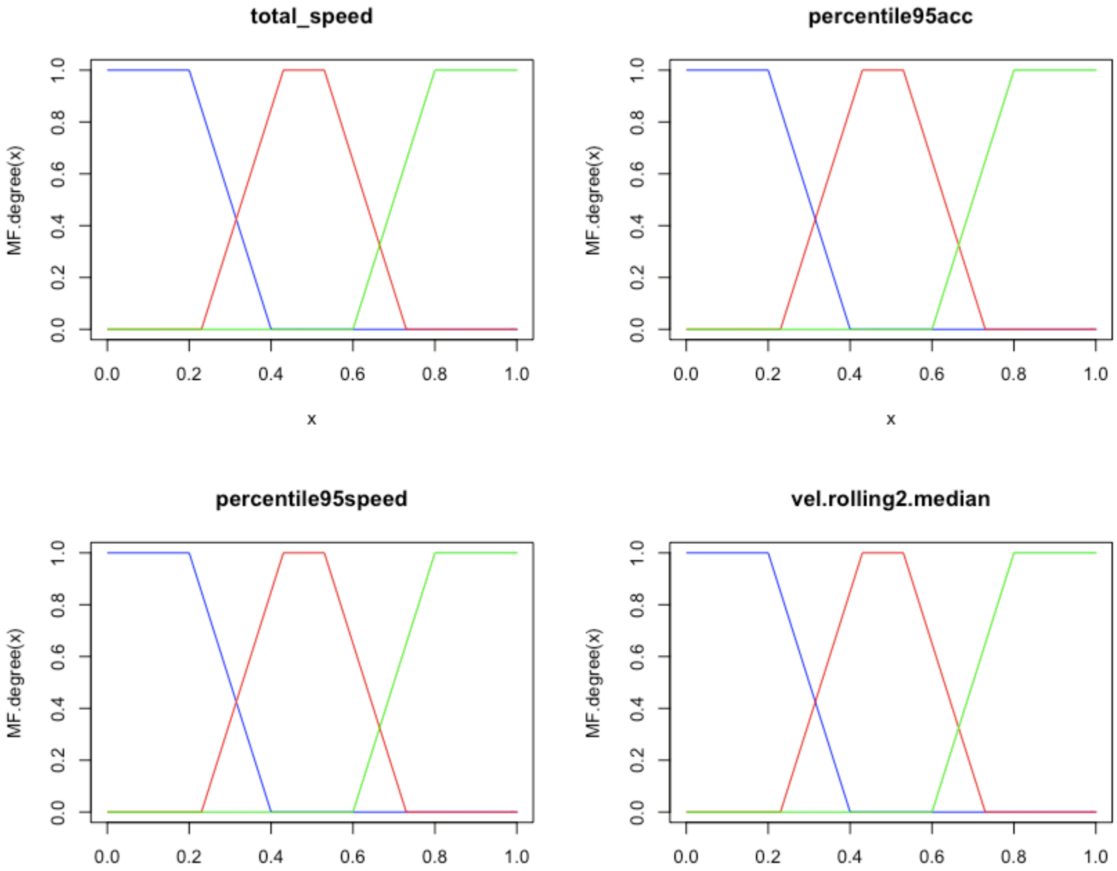


Figure 27 Trapezoidal membership functions and parameter values based on normalised data. Different memberships small, medium large from left to right

Each rule’s certainty degree is calculated by aggregating the degree of membership functions in the antecedent parts of the rule. Rule *a* and *b* have the same antecedent parts, but the degree differs. As a result, trajectories with characteristics fitting the antecedent parts of *a* and *b* will be preferably assigned to “bike” mode. Redundant rules are also deleted, resulting in the final fuzzy rule base (Chi et al., 1996).

Most earlier studies that use GPS information use trapezoidal membership functions (Axhausen and Schüssler, 2009; Das and Winter, 2016a; Rasmussen et al., 2015). Das & Winter found that due to the geometrical nature of the trapezoidal shape, there are cases where an input feature may fall outside the given range and may bear a zero membership value (Das and Winter, 2016b). On the other hand, as Gaussian functions are asymptotic in nature, there will always be a certain membership value in the range of [m,1], where $\lim_m \rightarrow 0$. In initial exploratory sensitivity tests, it is found that the Gaussian membership functions do perform markedly better in terms of accuracy, all other parameters being set the same (number of labels etc.) (Section 5.2).

4.7 Machine Learning

After the first stage of classification (RBH), an alternative to the FL system as the second stage is the Random Forest (RF) learning algorithm. RF is a popular learning algorithm due to its comprehensible concepts and that they perform well in a variety of problems and are easy to train (Hastie et al., 2009b; Montini et al., 2014; Prelipcean et al., 2017). Studies have also showed that despite the high performances, RF does not over fit the data even as more trees are added to the forest (Breiman, 2001; Cutler et al., 2012).

The method is based on a combination of bagging, the process of aggregating results of multiple trees and random subspace, which is the selection of random subset of variables as candidates for splitting at a particular node. In RF, optimal variables of each split are obtained from a random subset of all input variables (Breiman, 2001). The RF consists of multiple decision trees whose root nodes are bootstrap samples of the individuals. RF determines the splitting criterion based on a random subset of variables that are selected at each node. By considering only subsets of variables, RF reduces the correlation between trees. The final prediction is thus determined as the majority vote of all the trees (James et al., 2013). This classification algorithm is implemented in the R-package *randomForest*. Being reasonably fast, it can be run without tuning parameters and also outputs numerical estimates of variable importance.

4.8 Unsupervised Learning

A major hurdle in using CSD is the difficulty of obtaining ground truth data due to the inconveniences imposed by telecommunications provider in tracking an individual to their data trails. As mentioned in Chapter 2, many studies using cellular network data do not have the privilege of validating their methods with ground truth data, but only through comparisons with other proxies such as Google Maps and census data from travel surveys. As such, this study has also chosen to explore an unsupervised method to see if it is feasible to be applied in other studies should there only be unlabeled data available. In unsupervised learning the goal is to cluster the data and find patterns to see if the data falls into separate and interpretable groups. The common method of unsupervised learning in this context, and one that will be tested due to its popularity is K-means clustering. The K-means algorithm partitions the points into groups such that the sum of squares from each constituent point to the assigned centers is minimized. As using the squared distance assigns the highest influence to the largest distances, one drawback of the K-means method is a decreased robustness against outliers that produce high Euclidean distance values (Hastie et al., 2009a). One solution addressing this is the Partitioning Around Medoid technique (PAM). It is an extension of K-medoids clustering and compared to K-means, the PAM algorithm searches for *k* representative objects, or medoids instead of centroids, amongst each observation. *k* clusters are then generated by assigning each individual observation to the nearest medoid. The goal here is to minimize the sum of dissimilarities instead of the sum of Euclidean distances (Kaufman and Rousseeuw, 1990) to its nearest representative object. This dissimilarity measure can be derived from Random Forest predictors and has been used in unsupervised classification studies in the medical field (Dudoit and ridlyand, 2002; Shi and Horvath, 2006).

The PAM algorithm has two steps, build and swap. In the build phase, a collection of *k* objects is chosen to represent *k* clusters. The swap phase then tries to improve the clustering by exchanging these initially selected objects with potentially more representative ones, ultimately minimizing the average dissimilarity of objects to their closest selected object, or the medoids. One possible source of this dissimilarity input is from the proximity measure produced by RF without the use of labels. In an unsupervised RF, the original data is set considered as class 1, and a synthetic dataset, class 2, is generated by sampling at random from the original data. As a result, class 2 destroys the dependency structure in the original data and now a two-class problem can be fed to the RF algorithm. The rationale behind this is that the original observations will usually end up in the same terminal node of a tree, what the proximity matrix measures (Breiman, 2001; Liaw and Wiener, 2002). After each tree is generated, the data is fed down the tree and proximities are computed for every pair of observations. If the observations end up in the same terminal node, their proximity increases by one. At the end, proximities are normalized by dividing each proximity value by the number of trees. The result is an NxN matrix where each proximity value is between 0 and 1.Thus, this proximity matrix can be utilized as the dissimilarity matrix for clustering to divide the original data points into groups.

Some of the strengths of RF proximities are that it handles mixed variable types well, and is robust to outlying observations (Shi and Horvath, 2006; Liaw and Wiener, 2002). Similar observations should end up in the same terminal mode more often than dissimilar ones. As the similarity between an object and itself is one, the proximity matrix is symmetric. Finally, from this proximity matrix, we can derive the dissimilarity matrix with the equation (Shi and Horvath, 2006):

$$dissimilarity_{ij} = sqrt(1 - proximity_{ij})$$

The obtained dissimilarity matrix is then used as the input to the PAM clustering to obtain clusters represented by their medoids using the *pam()* function from the *cluster* package in R. Variables that are used in the clustering are those that are identified in PCA analysis as these are the variables that reflect the greatest variation in the data, and from existing literature. Next, modes are assigned to these clusters based on the characteristics of these medoids manually through expert and contextual knowledge derived by human reasoning. The results are then compared with the actual labels of the observations to evaluate performance of this method.

4.9 Variable selection

As mentioned earlier, there are three ways of selecting input variables. The first is by taking reference from current studies. As most, if not all the studies that employ this method of mode detection work with GPS trajectories, it is problematic to apply the same ranges that were set by human experts to the data in this study. Due to the poorer temporal and spatial accuracy and consistency of that accuracy, the resulting values of velocity and acceleration are not as fine-tuned as that of GPS data. As such, we will just be taking reference to the variables that all these studies have selected, which are median speed, 95th percentile speed and 95th percentile acceleration. In order to improve this, we add proximity variables (*ubahndist, tramdist, sbahndist, busdist*) as well as the *total_speed*. Due to the limitations of the segmentation and data cleaning process, *total_speed* can provide a more realistic picture of the trajectories velocity. This following part describes the two other variable selection procedures that are used to pick the input variables for the FL system and unsupervised clustering algorithms, RF and PCA respectively.

Random Forest

The data-driven variable selection method explored here is with the random forest algorithm. Section 2.5 has detailed some of the relevant strengths of analyzing large multi-dimensional datasets with random forest. This type of variable selection method is an entropy method (Guyon and Elisseeff, 2003). A variable with high entropy means it results in nodes that are high in uniformity. The variables that are considered as inputs in the RF algorithm or FL system are the best performing set of variables with high Gini coefficients (how much each variable contributes to the homogeneity of the nodes after the split, i.e. a more informative variable). The Boruta method is a process of variable selection with the RF algorithm that has been proven to be powerful and robust approach for its purposes (Degenhardt et al., 2017). The algorithm works as follows (Kursa et al., 2010):

1. For each attribute, a corresponding "shadow" attribute is created by shuffling values of the original attribute across all the data points.
2. Classification is performed on this extended database, and importance of all attributes are computed. In this case, the mean loss of accuracy metric is used and Z-scores are calculated by dividing the average loss by its standard deviation.
3. If the Z score of a shadow variable is significantly higher than that of its original variable, that variable is deemed as unimportant. This is because the importance of shadow variables will thus be non-zero only as a result of random fluctuations. Therefore, these importance values are used as a reference to deem if a variable is important or not. All unimportant and shadow variables are then removed.
4. The stability is increased as the steps are repeated over multiple random forest runs.

Figure 28 shows how the Z-scores vary amongst each variable and one can see that the important and unimportant variables are clearly separated by the variable that has the most important shadow variable. Finally, these variables that are deemed as important, as depicted in green, are then used as the input variables for the FL system, or the final RF algorithm.

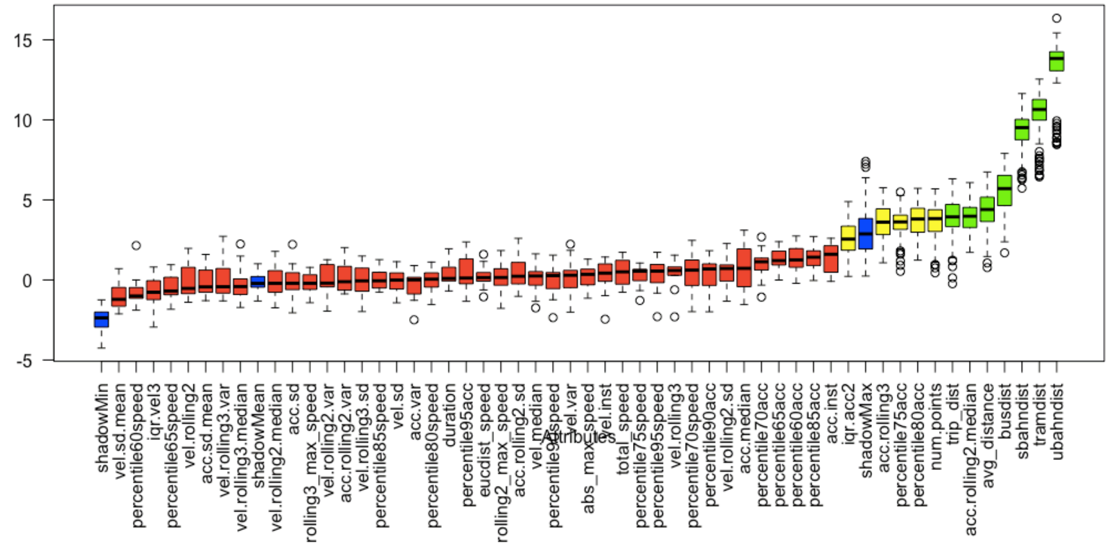


Figure 28 Plot of Z-scores after Boruta algorithm is run on variables. Blue boxplots represent to minimal, average and maximum Z score of a shadow attribute. Red and green boxplots represent Z scores of respectively rejected and confirmed attributes. Yellow are the tentative variables

Principal Component Analysis (PCA)

PCA is a process of transforming data by plotting it on different axes. As labels are not required, this method of variable selection is used for the unsupervised clustering algorithms. In order to find these PCs lines, one derives eigenvectors and values, which come in pairs. The principal component is thus the eigenvector with the highest eigenvalue. Before PCA is applied to the dataset, it is first scaled/normalized so as to increase their comparability with each other. Standardization is done by subtracting the mean from the value and dividing this value by the standard deviation. Standardization helps to understand how far above or below a value lies in relation to the other values in the distribution. Having a mean of 0 and a variance of 1, they have an equal scale and the same variance and all variables are therefore assumed to have equal importance and the same opportunity to be modeled (Bro and Smilde, 2014).

Table 10 shows the summary of the first 8 components when PCA is performed on all trajectories and corresponding variables of Vienna. The last row shows the cumulative proportion of variance explained in the data by that PC and the ones that came before it.

Importance of components	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Standard deviation	4.954	3.592	1.773	1.533	1.349	1.228	1.097	0.991
Proportion of Variance	0.463	0.243	0.059	0.044	0.034	0.028	0.023	0.019
Cumulative Proportion	0.463	0.707	0.766	0.810	0.844	0.873	0.896	0.914

Table 10 Importance of components in PCA of Vienna

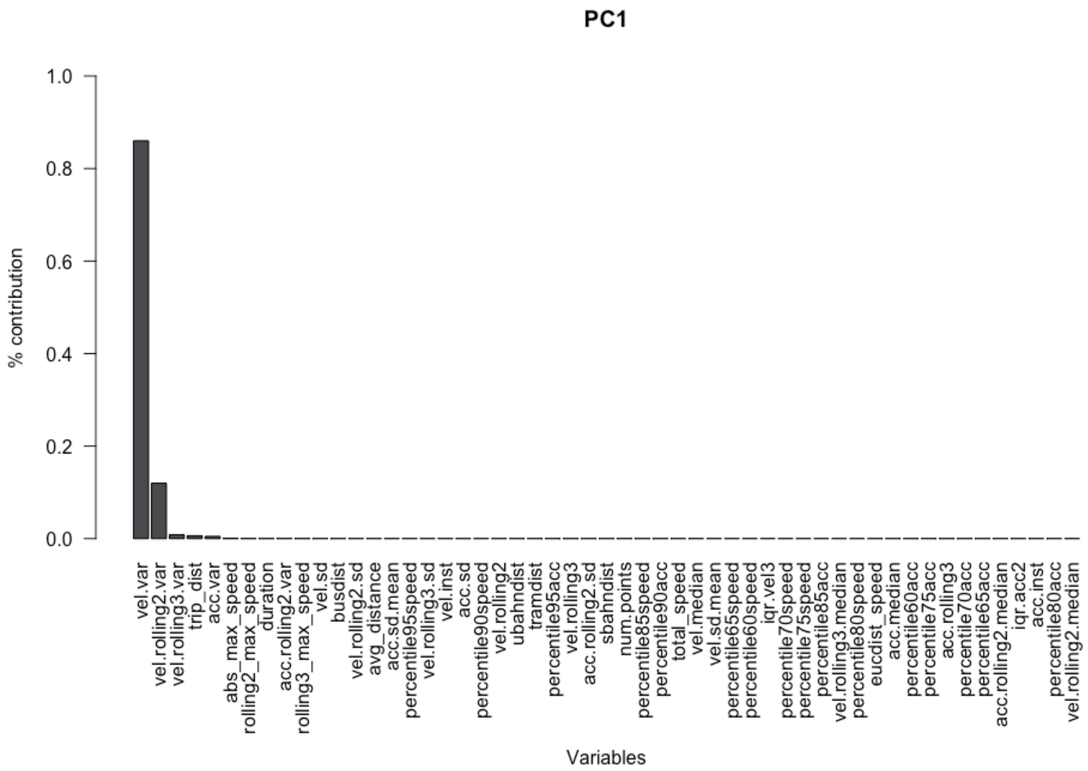


Figure 29 Variables ranked based on their contributions to the first principal component in the dataset for this study (Vienna).

As mentioned, each principal component will be dominated by a variety of input variables. For example, when looking at the first principal component (PC1) (Figure 29), it is dominated by *vel.var,* which contributes over 80% of the component. The subsequent variables are also other measures of variance of velocity. Furthermore, we can see from Table 10 that PC1 contributes to almost half of the variance observed in the data. This could indicate the importance of the variation in velocity when it comes to mode detection. It is interesting to note that the GIS information such as *tramdist*, *ubahndist* are not found to contribute highly to these first few PCs. This might be due to the low variability in our ground truth data, where most of these follow the same paths, or use the same mode. As such, there is a low range of average trajectory distances to the U-Bahn line, for example. Figure 30 shows PC2 and the contributing variables. The cumulative proportion of variance explained by the first two PCs is relatively high, at 70%. *trip_dist* contributes a vast proportion of PC2, which can also be an indicator of the importance of this particular variable.

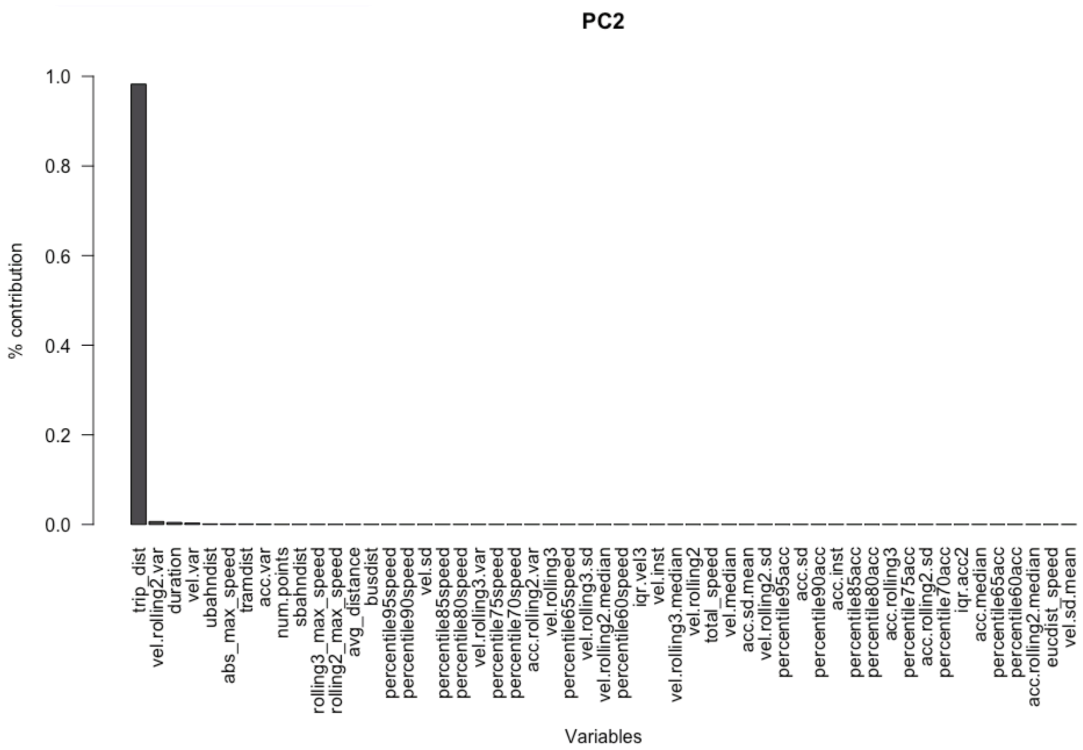


Figure 30 Variables ranked based on their contributions to the second principal component in the dataset for this study (Vienna)

Now with the PCA done, the next step is to the part of variable selection from the long list. Because the data is orthogonally transformed onto a new coordinate system, the values are scaled and variables do not explicitly represent the system-produced variables, hence applying PCA to the data set causes it to lose interpretability. However, PCA can still be used as a method of unsupervised variable selection. This can be done in a few ways (King and Jackson, 1999), but as mentioned in section 2.4.5, a recommended method of variable selection using PCA is a combination of the Broken-stick model and the B4 method.

The Broken-stick model assumes that the expected eigenvalue distribution will follow a Broken-stick distribution, meaning that observed eigenvalues that exceed the expected value generated by the broken stick model is deemed as interpretable. This model has been identified as a consistent approach to derive a cut off for eigenvalues (Bro and Smilde, 2014; Jackson, 1993). The B4 method is a relatively simple approach to variable selection. For example, if the Broken-stick model finds that the number of variables and hence PCs to use is *k*, the B4 method retains all *k* variables starting with the first component and keeping the variable with the greatest contributions to it. This is repeated *k* times for the *k* principal components to obtain the reduced subset of variables. While there are other methods of variable selection using PCA available, King & Jackson found that this combination was preferred because of its simplicity and performance, with good measures of fit and similarity (King and Jackson, 1999). Figure 31 shows the plot of eigenvalues using the data from the PCA of Viennese trajectories. According to the broken stick model, only the first two components are seen as interpretable. Next, to identify the two components, we look at the highest contributing variables of PC1 and PC2 each. This results in *trip_dist* and *vel.var*, which would then be included in the list of variables around which to perform unsupervised clustering.

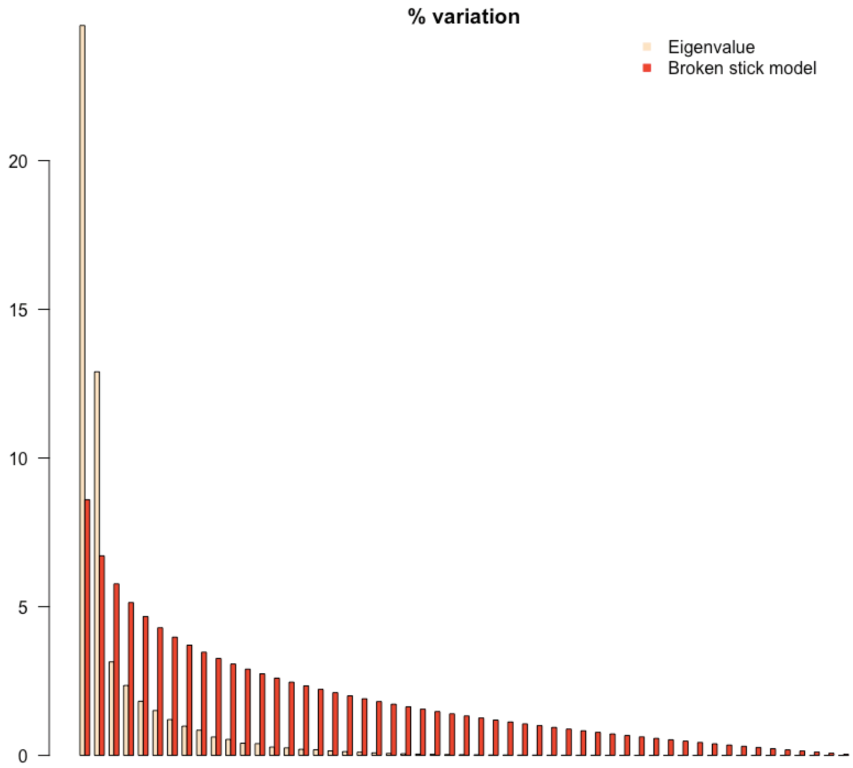


Figure 31 Comparison of eignevalues and values from the Broken-stick model

As the dataset we have is not large, the chosen method to evaluate the performances of the proposed algorithms is using a Leave-One-Out Cross Validation (LOOCV) approach. The LOOCV is a type of cross-validation that for *k* number of observations, the algorithm is run k times. At each iteration the i^{th} observation, where *i* = 1, 2, … *k*, is removed and the algorithm is tested on the i^{th} observation. For example, on the RBH + FL using existing literature method, in the i^{th} iteration, observation *i* will be the test data and the rest will be fed into the algorithm. So if *i* is not identified with a mode in the RBH step, only then will the algorithm continue into the fuzzy logic step, whereby all datapoints except *i* will be used to run the RF for variable selection and then generate the FL system. *i* will then be used as the test for the FL system to assign a mode.

CHAPTER 5
RESULTS

5.1 Validation

Validation will be the basis of several metrics of performances. Most mode detection studies use 4 particular measures, namely accuracy, precision, recall and F1-score. As the datasets provided were rather imbalanced in terms of class distribution, other metrics that try to account for this are also explored. They can be described as follows (Tan et al., 2006):

1. **Accuracy**: The percentage of correctly inferred modes out of the total number of trajectories, and is an intuitive performance measure. This refers to all modes
2. **Precision**: The percentage of correctly identified modes out of the total number of trajectories identified with that particular mode. For example, precision is the probability that a randomly selected trajectory identified as a car in the algorithm is actually a car trajectory. Each mode will have a precision value.
3. **Recall**: The percentage of correctly identified modes out of the number of trajectories of that particular mode. For example, recall is the probability that a randomly selected car trajectory is identified as a car in the algorithm, and is how good the algorithm is as detecting that particular mode. Each mode will have a recall value.
4. **F1 score**: A metric that is a combination of both precision and recall to give a better sense of the performance of the algorithm. It is the geometric mean of both precision and recall, and is especially useful when there is an uneven class distribution. A higher F1 score ensures that both precision and recall are reasonable high. Each mode will have a F1 value.
5. **Kappa statistic/Cohen’s Kappa**: An accuracy metric that is normalised by the imbalance of classes in the data, by taking into account the possibility of correct identification occurring by chance. It compares the actual accuracy with the expected accuracy, which is the accuracy that results from classification by random chance. For example, a classifier is designed to assign whether an object was an apple or a pear. Assuming there are 10 apples and 10 pears, and that the classifier is not that powerful, the results are:

	Apple	Pear
Apple	8	2
Pear	5	5

The observed accuracy is (8+5)/20 = 65%, which seems to be above average. To calculate expected accuracy, we use the number observations of each class as well as the number of identified instances of each class.

Probability of object being an apple = 10/20 = 0.5

Probability of classifier identifying an apple = 13/20 = 0.65

Probability of both agreeing = 0.4*0.65 = 0.325

Probability of object being an pear = 10/20 = 0.5

Probability of classifier identifying an pear = 7/20 = 0.35

Probability of both agreeing = 0.5 * 0.35 = 0.175

Expected accuracy is the probability of agreement for both apples and pears = 0.325 + 0.175 = 0.5

Equation for Kappa:

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e}$$

Where P_0 = observed accuracy, P_e = expected accuracy

Kappa = (0.65- 0.5)/ (1-0.5) = 0.3

This shows that the model does not actually perform as well as the 65% accuracy suggests, as it performs only marginally better than if the classifier chose how to identify the objects by random chance. The table below shows how the Kappa statistic can be interpreted.

Value of Kappa	Level of Agreement	% of Data that are Reliable
0–.20	None	0–4%
.21–.39	Minimal	4–15%
.40–.59	Weak	15–35%
.60–.79	Moderate	35–63%
.80–.90	Strong	64–81%
Above.90	Almost Perfect	82–100%

Table 11 Interpretation of Cohen's Kappa (McHugh, 2012)

5.2 Parameter settings

Parameter settings for FL systems and RF algorithms are chosen on the recommendation of sensitivity analyses. The following section details the sensitivity tests that they were subjected to.

5.2.1 Fuzzy Logic Systems

For FL systems, the parameters that are tested are the membership function shapes, and the number of linguistic terms used to describe the variables. Most earlier studies that use GPS data use trapezoid (Axhausen and Schüssler, 2009; Das and Winter, 2016a; Rasmussen et al., 2015) membership function. Das & Winter found that due to the geometrical nature of the trapezoidal shape, there are cases where an input feature may fall outside the given range and may bear a zero membership value (Das and Winter, 2016). On the other hand, as Gaussian functions are asymptotic in nature, there will always be a certain membership value in the range of [m,1], where $\lim_m \rightarrow 0$. The studies mentioned also use 3 levels of membership functions for the fuzzy variables (i.e. low, moderate, high). Figure 32 shows how the various performance metrics change with *Trapezoid* or *Gaussian* membership function shapes, and the number of linguistic terms from 2 to 6. The sensitivity analyses were done with variables from existing literature, though the input variables should be arbitrary as the aim is to compare the parameters. The graph shows that in general, as suggested by Das and Winter (2016b), results from using *Gaussian* membership functions seem to perform slightly better than that of *Trapezoid* membership functions. In terms of the number of levels, the results show that across accuracy and kappa, having 5 levels leads to the best performance for *Gaussian* membership functions. For *Trapezoid* membership functions, the best number of levels to use is not as clear. Having 2 levels yields the highest accuracy and having 5 yields the highest kappa. As such, the FL systems used in this study will be set to *Gaussian* membership functions and with 5 descriptive levels (i.e. very low, low, moderate, high, very high).

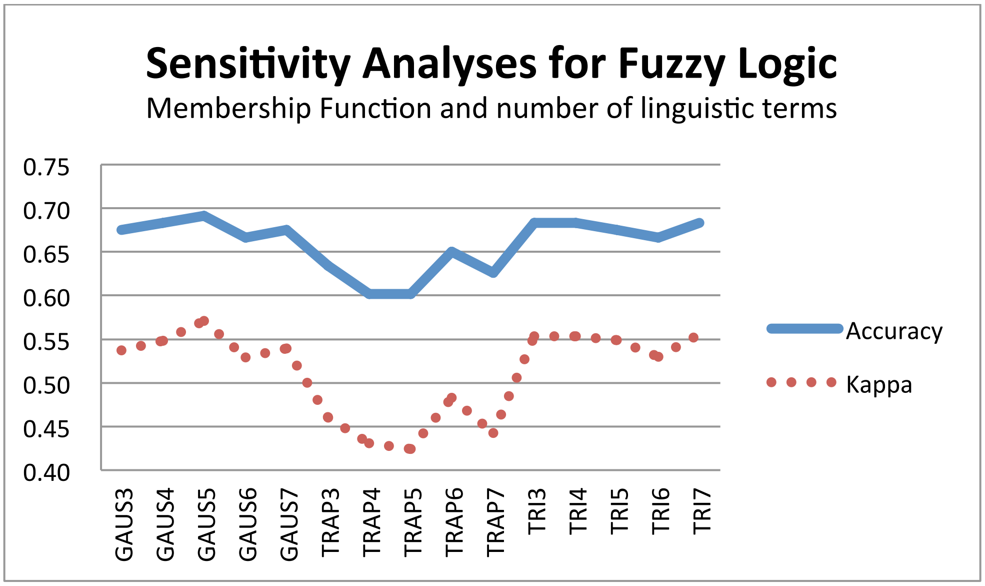


Figure 32 Sensitivity analysis for FL parameters of RB_FLEL method on the Viennese dataset.

5.2.2 Random Forest

For the random forest algorithms, the sensitivity analysis tests for the number of trees (*ntree*) to grow and the number of variables randomly sampled as candidates for each split (*mtry*). These parameters tested will include *ntree* = 100, 200, 400, 500, 600, 1000 and 1500 and *mtry* = 1, 2, 4, 5, 6 and 7. The results can be seen in Figure 33. There is a general increase in accuracy and Kappa for all *ntree*. There seems to be no clear pattern with *ntree*. The best performing permutation of these parameters is *ntree = 400, mtry = 6*, as it has the highest Kappa statistic and accuracy. As such, the parameters that will be set for all methods using RF as the main mode detection method will use these parameters. As Liaw and Wiener (2002) have found in their studies, that while various parameter settings in RF leads to different variable importance values, the ranking of the importance is quite stable. Furthermore, regardless of the parameters for RF and FL, the overall accuracy seems to vary with a relatively small range of 0.1 (out of 1). Consequently, while it is still important to validate results, it may not be of huge consequence to use parameter settings that deviate slightly from the optimal specification.

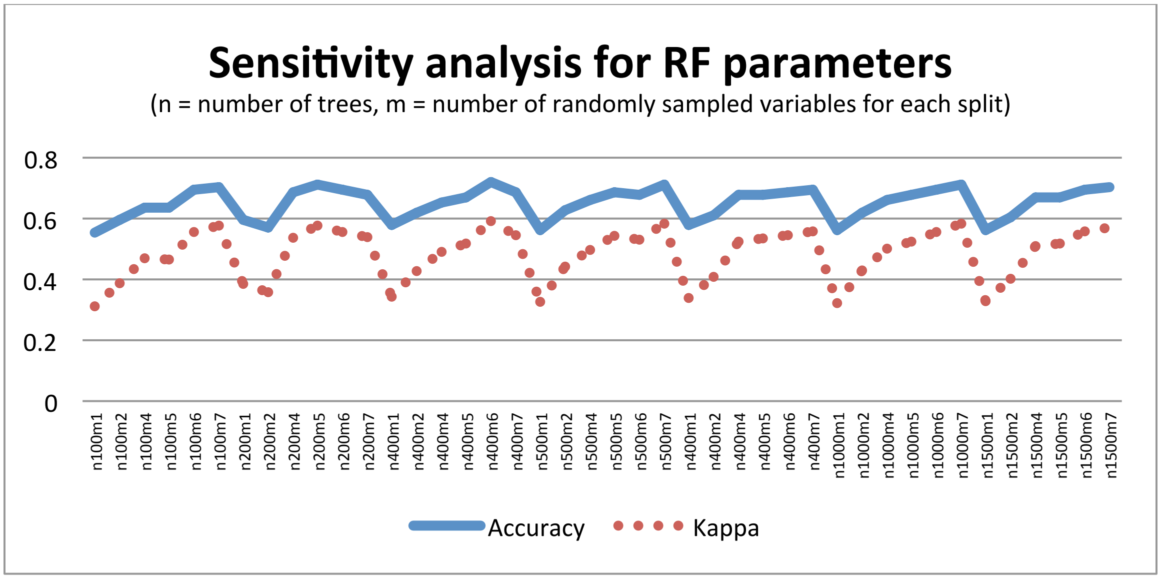


Figure 33 Sensitivity analysis for RF parameters

5.2.3 Unsupervised K-means and PAM with RF

For the unsupervised clustering methods, the number of clusters, *k*, will be set to the number of classes of interest. This would correspond to the number of classes in the data. For Vienna, the number of clusters set is 4. As we do not set the initial points for the building of the clusters, each algorithm is run 5 times. Each time the cluster centroids and medoids are analyzed and modes assigned manually. The performance metrics are then averaged over the 5 runs. While this leads to some bias in terms of knowing the number of modes that exist in the dataset (not known in real life if truly unsupervised), this is sufficient for the purposes of the experiment, which is to test the feasibility of this unsupervised learning technique on mode detection.

5.3 Pre-processing and trip-segmentation

The results are presented in this section in terms of precision, recall, F1 for the various modes, followed by accuracy and kappa. They are evaluated on their ability to correctly identify the mode of transport of the segments. Each method is referred to by an abbreviated code, as show in Table 12.

Code	Method
RB_prefix	Combined with rule-based heuristic
FLEL	Fuzzy Logic with variables from existing literature
FLPCA	Fuzzy Logic with variables selected from PCA
FLRF	Fuzzy Logic with variables selected from RF
RF	Random Forest
PAM	Partitioning Around Medoids
KMEANS	K-means clustering

Table 12 Code for methods

As precision, recall and F1 scores are calculated for each mode in each method, the averages of these measures for each method are calculated by summing the product of each class's value and the number of modes in that class, and dividing it by the total number of instances.

The pre-processsing, trip-segmentation and labeling led to the final dataset as shown in Table 15. In total there are 399 trajectories, 86 of which are non-trips. 144 trajectories belong to the Vienna dataset and 255 trajectories for Graz. The non-trips are moving trajectories whereby there were no corresponding GPS labels for mode of transportation that was reported by the participant. This could be due to underreporting, that was observed in GPS studies like Rasmussen et al. (2015) and Stopher et al (2008). Some of these non-trips were a result of the algorithm identifying periods where the participant is stationary as periods of moving due to the noisiness in the data. When a person is at work, for example, the CSD generated could still lead to speeds that are higher than walking due to errors in triangulation estimation, or due to jumping between cell towers. As the method of data cleaning here, like other studies using cellular network data, clean outliers using a speed threshold. Errors that are smaller but large enough to generate speeds that would be realistic in faster modes of transportation are not filtered out. For example, if a person were stationary, data points that have speeds of over 6m/s would not be removed, as it is still a realistic speed for a bicycle. As such, the algorithm would detect these as moving segments. Even though the processing step tries to account for this by dissolving short moving segments where the previous and subsequent stationary segments have centroids that are less than 500m apart, non-trips still occur as there is a trade off between missing out shorter trips or different parts of trips (i.e. when a journey consists of multiple modes) and having less non-trips. This is further compounded by the fact that the respondents in the second data collection campaign were instructed to actively collect data, and encouraged to collect data for varying transportation modes. This means that multiple changes can happen within a smaller area for the sake of data collection. This can also lead to another source of non-trips, whereby a single trajectory has multiple mode labels. In this scenario, if a moving segment has two labels and neither label has temporal coverage exceeding 80% of the duration of the moving segment, it will still be regarded as a non-trip. This study assumes that the segmentation in the first data collection campaign (from Invenium) yields no non-trips.

Despite efforts to collect varying modes, the resulting classes are still quite imbalanced, with U-Bahn trajectories making the majority in the Vienna dataset and bicycles and car trajectories making the majority in the Graz dataset. In both cases, there are modes where there are only one or two instances, like tram for Vienna and bus for Graz. Because of this, the tram modes will be excluded in the analysis for Vienna and similarly, the bus modes for Graz. This is partly due to the fact that in machine learning methods, one would need to have at least two instances of a particular mode to train and test a model. To maintain consistency amongst the city-wide analysis, these minority modes are excluded. It is worth noting that for the Viennese dataset, there is a more even split on private and public transportation modes, whereas in the Graz dataset, the significant majority belongs to private modes (car and bicycle).

Mode	Bicycle	Bus	Car	S-Bahn	Tram	U-Bahn	Walk	Total
Vienna	0	0	16	6	2	31	1	56
Graz	18	0	18	0	1	NA	6	43

Table 13 Table of number of trips extracted from first data collection (A)

Mode	Bicycle	Bus	Car	S-Bahn	Tram	U-Bahn	Walk	Non-trips	Total
Vienna	15	0	0	9	2	22	20	23	91
Graz	58	1	55	7	12	NA	4	63	200

Table 14 Table of number of trips extracted from second data collection (C)

Mode	Bicycle	Bus	Car	S-Bahn	Tram	U-Bahn	Walk	Non-trips	Total
Vienna	15	0	17	15	2	53	21	23	146
Graz	76	0	73	7	13	NA	10	63	255

Table 15 total mode shares of data (A + C)

5.4 Supervised methods

In this subsection, we will present the results of the supervised methods, both with and without the inclusion of RBH. For both Graz and Vienna, a similar pattern can be seen with regards to the overall performance of the methods. In both cities, RB_RF algorithm is the highest performing, in terms of both accuracy and Kappa statistic (Figure 34 and Figure 35). In Vienna, it achieved an accuracy of 0.73 and a kappa statistic of 0.61. While Cohen views anything above 0.61 as substantial, McHugh (2012) opines that it should be seen as moderate, as it implies that around to 40% of the information might be unreliable. Nevertheless, the study put 0.6 as a threshold for Kappa for placing confidence in the study results. For Graz, all the proposed methods perform poorer in comparison to the Vienna dataset, with the best faring RB_RF algorithm achieving an accuracy of 0.61 and has a corresponding Kappa of 0.36. There is lesser variation in the performance across the methods in the Graz dataset. Table 16 (Vienna) and Table 17 (Graz) show the corresponding precision, recall and F1 statistic measures when excluding the non-trips and when RBH is included or not. RB_RF has the highest F1-score compared to all other algorithms. The F1-score, which gives an indication of the values of both precision and recall, is high for all the public transportation modes (S-Bahn, U-Bahn), moderate for the walking modes and not as high for car and bike. This is also true for the Kappa score.

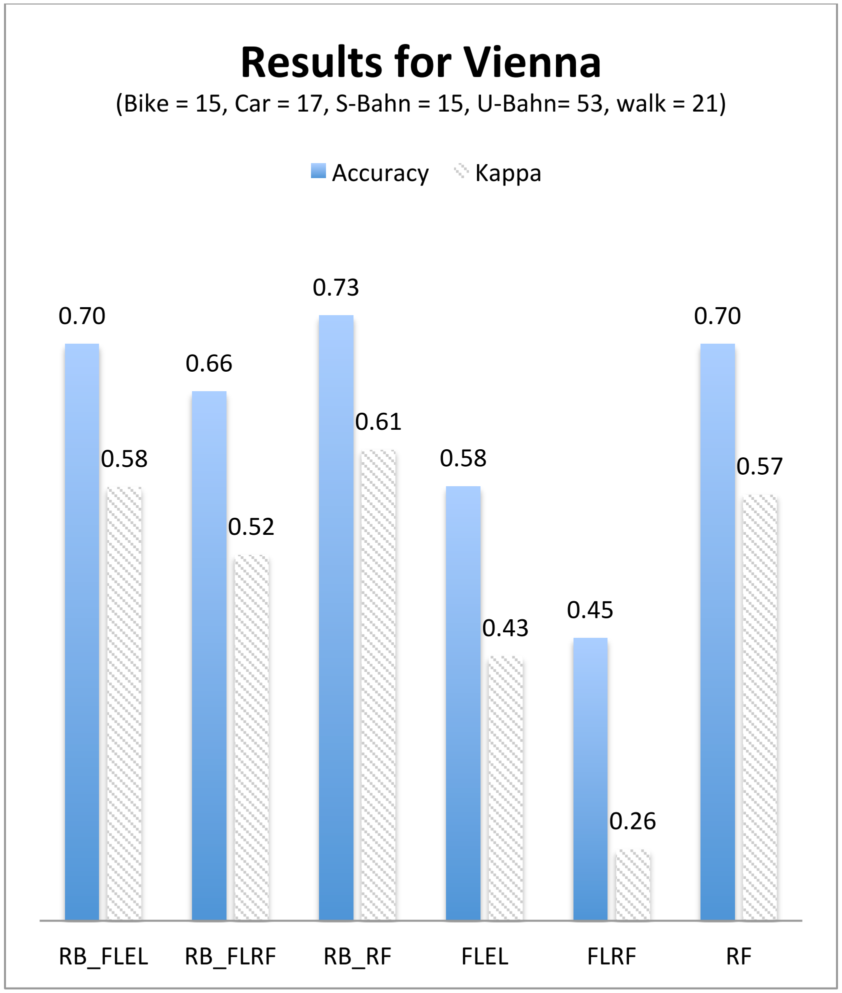


Figure 34 Results for Vienna dataset

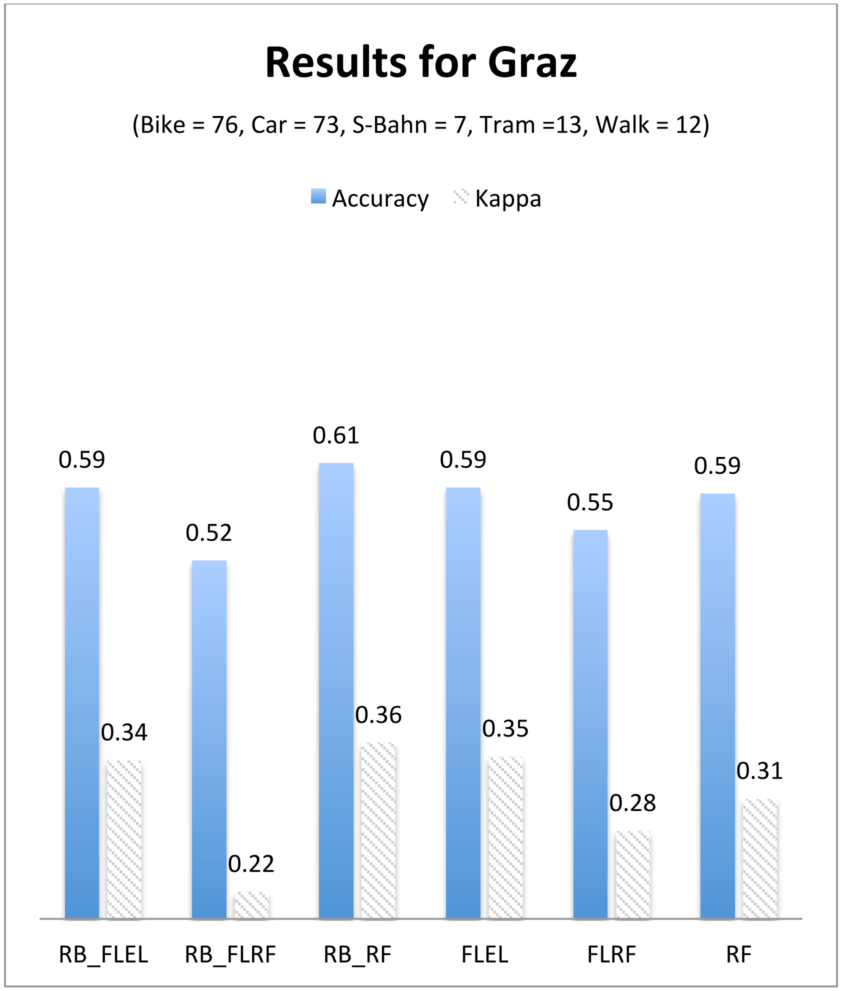


Figure 35 Results for Graz data

	RB_FLEL	RB_FLRF	RB_RF	FLEL	FLRF	RF	RB_FLEL	RB_FLRF	RB_RF	FLEL	FLRF	RF	RB_FLEL	RB_FLRF	RB_RF	FLEL	FLRF	RF
	Precision						Recall						F1					
Bike	0.43	0.45	0.60	0.27	0.29	0.63	0.40	0.33	0.40	0.40	0.40	0.33	0.41	0.38	0.48	0.32	0.33	0.43
Car	0.46	0.40	0.50	0.32	0.31	0.38	0.35	0.24	0.29	0.35	0.29	0.18	0.40	0.30	0.37	0.33	0.30	0.24
S-Bahn	0.76	0.70	0.88	0.63	0.44	0.88	0.87	0.93	0.93	0.67	0.53	0.93	0.81	0.80	0.90	0.65	0.48	0.90
U-Bahn	0.79	0.76	0.77	0.78	0.59	0.74	0.94	0.91	0.92	0.72	0.49	0.92	0.86	0.83	0.84	0.75	0.54	0.82
Walk	0.71	0.53	0.67	0.67	0.41	0.61	0.48	0.43	0.67	0.48	0.43	0.67	0.57	0.47	0.67	0.56	0.42	0.64
Average	0.68	0.62	0.70	0.61	0.46	0.65	0.70	0.67	0.69	0.58	0.46	0.61	0.68	0.63	0.71	0.59	0.45	0.52

Table 16 Precision, Recall and F1 values for each mode in Vienna

	RB_FLEL	RB_FLRF	RB_RF	FLEL	FLRF	RF	RB_FLEL	RB_FLRF	RB_RF	FLEL	FLRF	RF	RB_FLEL	RB_FLRF	RB_RF	FLEL	FLRF	RF
	Precision						Recall						F1					
Bike	0.61	0.57	0.62	0.61	0.56	0.60	0.75	0.86	0.74	0.75	0.83	0.74	0.67	0.68	0.67	0.67	0.67	0.66
Car	0.61	0.56	0.60	0.62	0.61	0.56	0.59	0.40	0.67	0.58	0.42	0.63	0.60	0.46	0.63	0.60	0.50	0.59
S-Bahn	0.67	0.00	1.00	0.50	0.56	1.00	0.86	0.00	0.71	0.43	0.71	0.29	0.75	0.00	0.83	0.46	0.63	0.44
Tram	0.00	0.00	0.33	0.00	0.00	0.50	0.00	0.00	0.08	0.00	0.00	0.15	0.00	0.00	0.13	0.00	0.00	0.24
Walk	0.14	0.11	0.00	0.42	0.13	0.00	0.08	0.08	0.00	0.42	0.08	0.00	0.11	0.10	0.00	0.42	0.10	0.00
Average	0.54	0.47	0.56	0.56	0.51	0.55	0.59	0.52	0.61	0.59	0.55	0.59	0.56	0.48	0.58	0.57	0.51	0.55

Table 17 Precision, Recall and F1 values for each mode in Graz

5.4.1 With RBH vs. without RBH

Figure 34 shows the accuracy and kappa of the dataset from Vienna. It can be observed that the combined methods (inclusion of RBH) yield consistently higher results, with the exception of when RF is used in isolation. The increase in accuracy when RBH is used is most pronounced in the two fuzzy logic methods, RB_FLEL and RB_FLRF, whilst having only a slight, but still positive change in RF algorithm. There is a huge improvement in Kappa scores, especially for RB_FLRF. In general, these metrics do not fare as well as when RBH is not used. For example, when RBH is included, average precision, recall and F1-score values are generally in the high 0.60s and exceed 0.7 for RB_RF and RB_FLEL. However, when RBH is not included, these values remain between 0.4 and 0.6. As expected, it is the public transportation modes that benefit the most from the inclusion of RBH since it exploits the fact that these public transportation modes follow fixed routes. The consistent better performance of the hybrid rule-based methods show promise in using this simple method as a primer for the secondary mode detection steps, especially when it comes to the rail modes. This is also consistent with results from other mode detection other studies using rule-based heuristics. Of all the S-Bahn and U-Bahn trajectories, less of them are assigned S-Bahn and U-Bahn modes when RBH is not used. Thus, it can be inferred that RBH is highly useful in the mode identification when there are public transportation modes that follow distinct routes. Consistent with this observation, S-Bahn modes fare particularly well in the Graz dataset, amidst the generally poor performance of the other modes. From this we can conclude that the inclusion of the transport network in the form of the RBH is effective for both private and public modes of transportation. The second best performing method, RB_FLEL, is 20.7% more accurate than its non-RBH counterpart, and experiences a 34.9% increase in the Kappa statistic. This means that with the inclusion of RBH, there is a higher probability that modes were assigned correctly by virtue of an accurate model rather than by pure chance.

Interestingly, despite the inference that public modes benefit more from the RBH step, the private modes seem to have a greater decrease in precision than that of public transportation. For example, in RB_FLEL, bike and car precision values drop from 0.43 and 0.46 to 0.27 and 0.32 respectively. On the other hand, S-Bahn and U-Bahn precision only drop from 0.79 and 0.76 to 0.78 and 0.6. This suggests that RBH is also important in making sure that S-Bahn, U-Bahn and walk trajectories are not mistakenly assigned as bike and car modes. Conversely, there recall measures for these private modes do not show a clear decrease when RBH is used, suggesting that many of the rail modes that would be identified by the RBH are identified as these private modes when RBH is not used.

For Graz however, this pattern is only seen when RF is used. For the FL methods (FLEL and FLRF), the results are poorer when RBH is used (when both accuracy and Kappa are taken into account). Again, this is likely to be a result of the bike and car heavy data in Graz as compared to rail mode heavy data in Vienna. The RBH step is only able to identify modes of a relatively low number of trajectories in the earlier step. Despite being able to detect some car trajectories, a lot of the car trajectories still do not reach the upper threshold set in the RBH (*percentile95speed* >12.5m/s). However, when this value is lowered, many other modes are misidentified as car modes, defeating the purpose of the RBH. This is due to the fact that there are quite a number of tram, bike and car trajectories whose *percentile95speed* values are found just below that threshold, leading to the tradeoff between precision and recall in the RBH step. In both cities, the best performing non-RBH method is also RF, which is expected, as RF is a popular choice and has many strengths (section 4.7). Figure 37 and Figure 38 show that the drop in performance is much clearer in the U-Bahn and S-Bahn modes, again with the exception of RF which still performs well without RBH. Car modes are also identified in the RBH step on the premise that it is able to generate high speeds. As such, the same figures also show a slight improvement, albeit much less evidently.

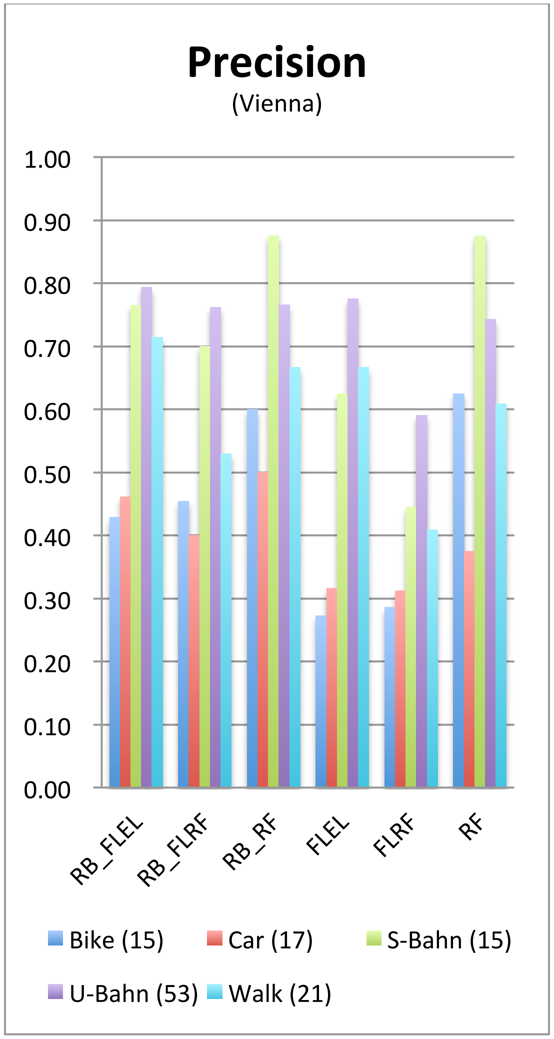


Figure 36 Precision of each mode in Vienna

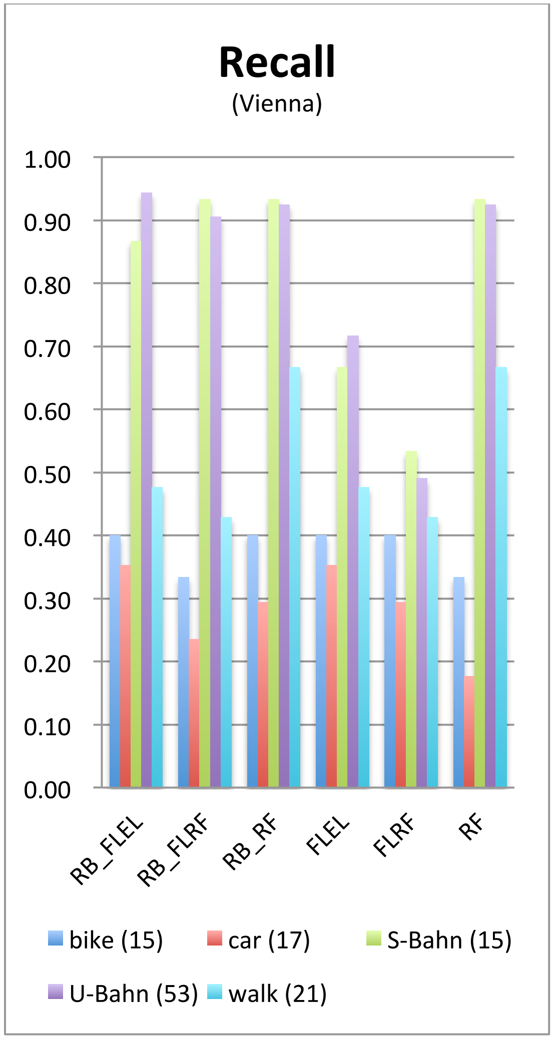


Figure 37 Recall of each mode in Vienna

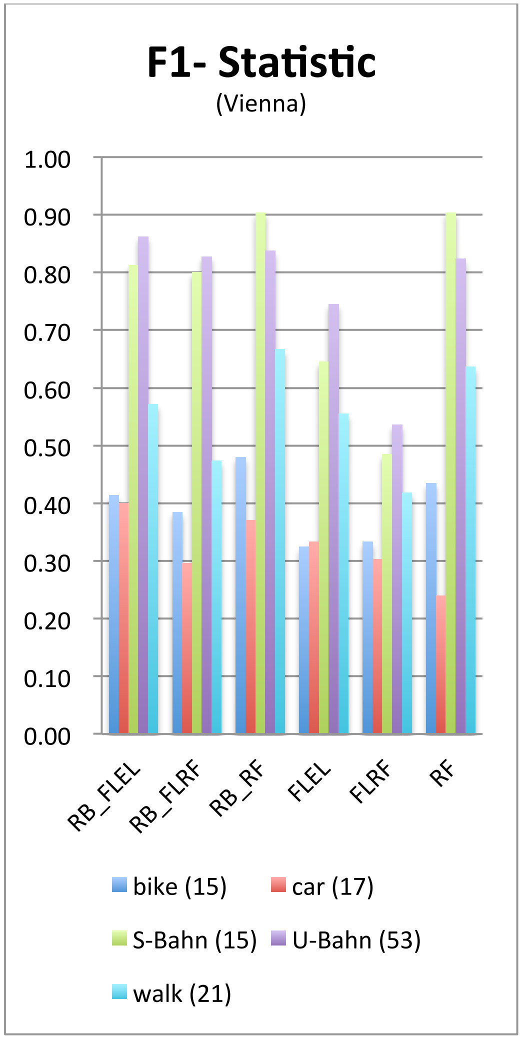


Figure 38 F1 of each mode in Vienna

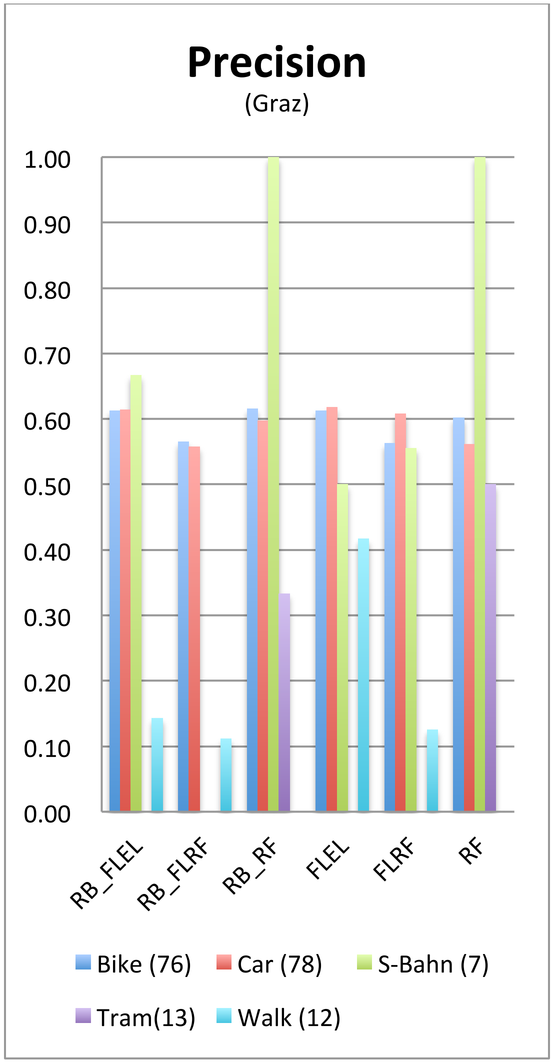


Figure 39 Precision of all modes in Graz

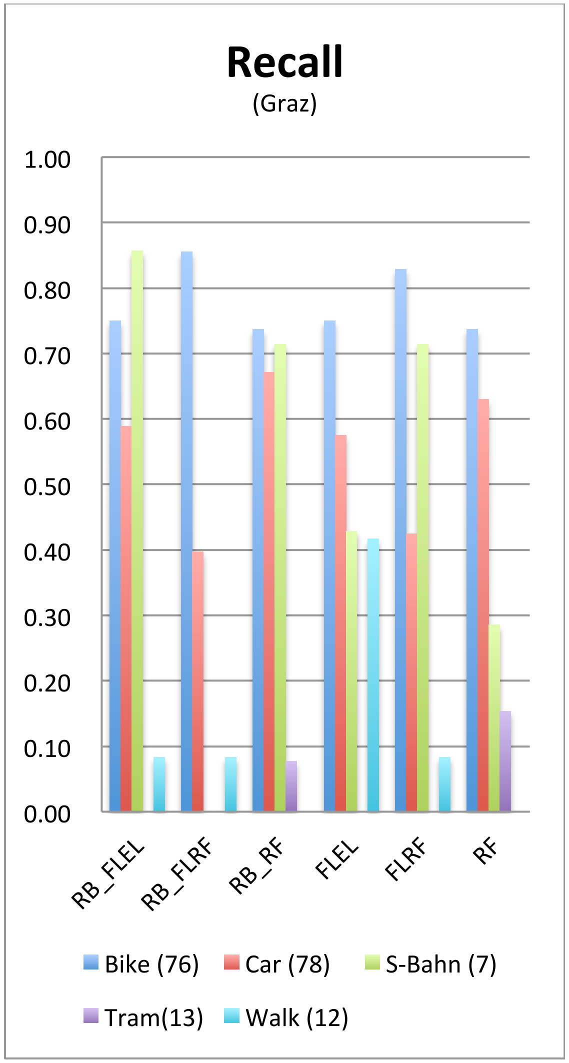


Figure 40 Recall of all modes in Graz

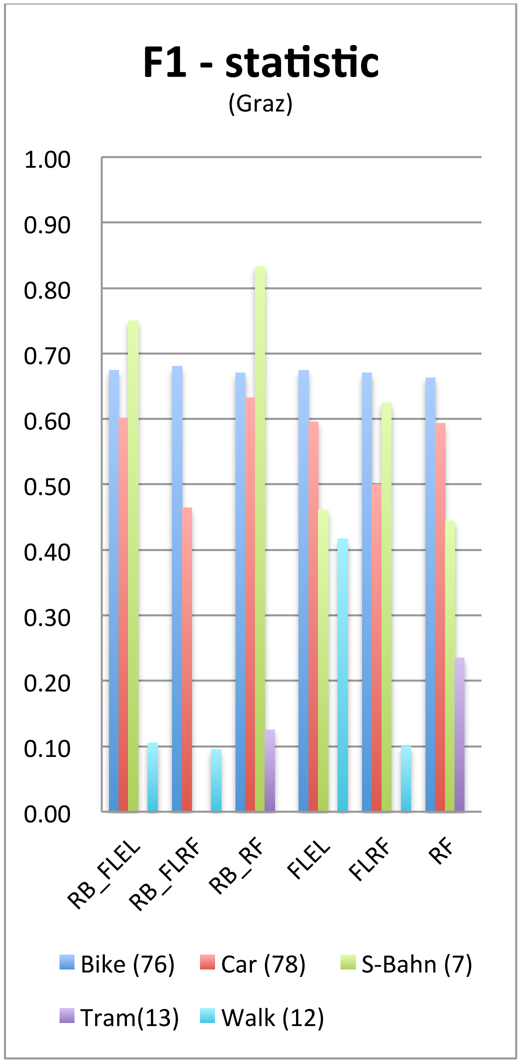


Figure 41 F1 statistic for all modes in Graz

The following tables go further in-depth to show the confusion matrices of each of the top performing combined methods in Vienna and Graz. Labels represent the transportation reported by the data collector and modes refer to what is inferred by the algorithm. “Total (actual)” refers to the actual frequency of each transportation mode while “Total (inferred)” refers to the frequency of modes inferred by the algorithm. The confidence rate can be understood as the precision when non-trips are also taken into account. For Vienna, the combined method of RBH and RF (Table 18) yields the best performance when the non-trips are excluded. This is followed closely by the combined method of RBH and FLEL. The confidence rates reflect the performance when the non-trips are taken into account. For example, when non-trips are also fed into the RB_FLEL algorithm (Table 19), 5 of the trajectories that are assigned mode “S-Bahn” are actually non-trips. As such, out of 22 trajectories identified as S-Bahn, (as supposed to 17), 13 are actually S-Bahn trajectories. The confidence rate is 0.59 as supposed to 0.76 if non-trips were not considered. As a result the total accuracy of the best performing method, RB_RF, is lowered from 0.73 to 0.61. These tables highlight a weakness of the segmentation approach, as reflected in the lower confidence rates.

Vienna							
RBH + RF							
	Mode						
Label	**Bike**	**Car**	**S-Bahn**	**U-Bahn**	**Walk**	**Total (actual)**	**Recall**
Bike	6	3	0	2	4	15	0.40
Car	1	5	0	10	1	17	0.29
S-Bahn	0	1	14	0	0	15	0.93
U-Bahn	1	0	1	49	2	53	0.92
Walk	2	1	1	3	14	21	0.67
Non-trip		2	5	7	9	23	
						121	**0.73**
Total (inferred)	10	12	21	71	30	144	0.61
Precision	0.60	0.50	0.88	0.92	0.77		Kappa =
Confidence	0.35	0.40	0.59	0.71	0.50		**0.61**

Table 18 Confusion matrix of Rule-Based Heuristics + Random Forest Vienna

Vienna							
RBH + FLEL							
	Mode					**Total**	
Label	**Bike**	**Car**	**S-Bahn**	**U-Bahn**	**Walk**	**(actual)**	**Recall**
Bike	6	4	0	3	2	15	0.40
Car	2	6	1	7	1	17	0.35
S-Bahn	0	1	13	1	0	15	0.87
U-Bahn	1	0	1	50	1	53	0.94
Walk	5	2	2	2	10	21	0.48
Non-trip	3	2	5	7	6	23	0.70
						121	
Total (inferred)	17	15	22	70	20	144	0.59
Precision	0.43	0.46	0.76	0.79	0.71		Kappa =
Confidence	0.35	0.40	0.59	0.71	0.50		0.58

Table 19 Confusion matrix of Rule-Based Heuristics + Fuzzy Logic with Existing Literature Vienna

The results for Graz are lower than that achieved by the Vienna dataset, which shows that these methods are more suitable to identifying other public transportation modes (Table 20 and Table 21). Again, the number of non-trips lowers the confidence greatly, to 0.41 from an accuracy of 0.61 when non-trips are not considered. Walk trips for Graz perform very poorly for these two methods, assigning all walk trajectories as bike and car modes. Compared to Vienna, the walk modes fare considerably worse, even when the same variables are used, i.e in RB_FLEL. On closer inspection of the trajectories, the speed and acceleration profiles of walk trajectories extracted in the Graz dataset are extremely different than those extracted in the Viennese dataset (Table 22). This could be due to the fact that the majority of respondents were based in Graz, meaning that when data was being collected in Vienna, more data points were generated as walking around an unfamiliar city might mean navigation applications might be open for a greater proportion of the time. The data generated in this case would be more reflective of the actual location and speeds. While both RB_RF and RB_FLEL display considerable confusion between car and bike modes, RB_RF is able to distinguish between the two modes to a slightly higher degree, as can be seen in the confusion matrices (Table 20 and Table 21).

Graz							
RBH + FLEL							
	Mode						
Label	**Bike**	**Car**	**S-Bahn**	**U-Bahn**	**Walk**	**Total**	**Recall**
Bike	57	16	0	2	1	76	0.75
Car	22	43	3	0	5	73	0.59
S-Bahn	0	1	6	0	0	7	0.86
Tram	10	3	0	0	0	13	0.00
Walk	4	7	0	0	1	12	0.08
Non-trip	42	32	1	7	0	82	
						181	**0.59**
Total	135	102	10	9	7	263	0.41
Precision	0.61	0.61	0.67	0.00	0.14		Kappa =
Confidence	0.42	0.42	0.60	0.00	0.14		**0.34**

Table 20 Confusion matrix of Rule-Based Heuristics + Fuzzy Logic with variables from existing literature Graz

Graz							
RBH + RF							
	Mode						
Label	**Bike**	**Car**	**S-Bahn**	**U-Bahn**	**Walk**	**Total**	**Recall**
Bike	56	20	0	0	0	76	0.62
Car	21	51	0	1	0	73	0.60
S-Bahn	0	2	5	0	0	7	1.00
Tram	9	2	0	2	0	13	0.33
Walk	4	8	0	0	0	12	NaN
Non-trip	60	26	1	0	0	87	
						181	**0.61**
Total	150	109	6	3	0	268	0.42
Precision	0.74	0.67	0.71	0.08	0.00		Kappa =
Confidence	0.37	0.47	0.83	0.67	0.00		**0.36**

Table 21 Confusion matrix of Rule-Based Heuristics + Random Forest Graz

		Min	1st Q	Median	Mean	3rd Q	Max
percentile95acc	**Graz**	0.27	0.58	1.44	1.49	2.34	3.21
	Vienna	0.01	0.07	0.23	0.53	0.83	1.98
percentile95speed	**Graz**	5.94	9.36	13.38	13.51	17.09	24.87
	Vienna	0.68	2.52	6.92	7.54	9.20	27.84
total_speed	**Graz**	1.44	3.66	5.86	5.76	7.50	12.41
	Vienna	0.45	1.05	1.96	2.77	2.99	8.33

Table 22 Comparison of speed and acceleration profiles of walk trajectories in Graz and Vienna

The results presented in this section indicate that the initial RBH step is a worthy inclusion in the proposed methods when CSD is concerned. By placing a higher importance on these proximity and 95th percentile speed and acceleration measures, the RBH step is useful in filtering out the rail and some car modes, in the process preventing these rail modes from being mistaken as other modes. Precision for rail modes and recall for other modes benefit slightly more from RBH. In Viennese dataset supports this conclusion more strongly as compared to the Graz dataset, likely due to the large number of rail modes in the former dataset. Despite this, both cities still show that there is a case for incorporating RBH into the mode detection step, especially when it is combined with RF in the RB_RF method.

5.4.2 Random Forest vs Fuzzy Logic

Now that it is clear that the inclusion of RBH is beneficial, further evaluation of results will focus on the combined methods. While RBH increases performance, some examples of modes that are not identified in the RBH step include rail segments that have lower *tramdist* values than *sbahndist/ubahndist* values (distance to appropriate network is not the smallest out of all public transportation modes) or car segments that are considered as part of the public transportation network due to their proximity to the network links, to name a few. In order to study the differences between the two secondary mode detection methods (Fuzzy Logic and Random Forest), Table 23 and Table 24 show the confusion matrices of trajectories that are identified in the secondary steps of RB_FLEL and RB_RF algorithm in Vienna.

RB_FLEL	Mode				
Label	**Bike**	**Car**	**S-Bahn**	**U-Bahn**	**Walk**
Bike	6	2	0	1	2
Car	2	1	1	1	1
S-Bahn	0	0	1	1	0
U-Bahn	1	0	0	7	1
Walk	5	1	1	1	10

Table 23 Confusion matrix of trajectories that are assigned modes in the FLEL step of RB_FLEL

RB_FLEL	Mode				
Label	**Bike**	**Car**	**S-Bahn**	**U-Bahn**	**Walk**
Bike	6	1	0	0	4
Car	1	0	0	4	1
S-Bahn	0	0	2	0	0
U-Bahn	1	0	0	6	2
Walk	2	0	0	2	14

Table 24 Confusion matrix of trajectories that are assigned modes in the RF step of RB_RF

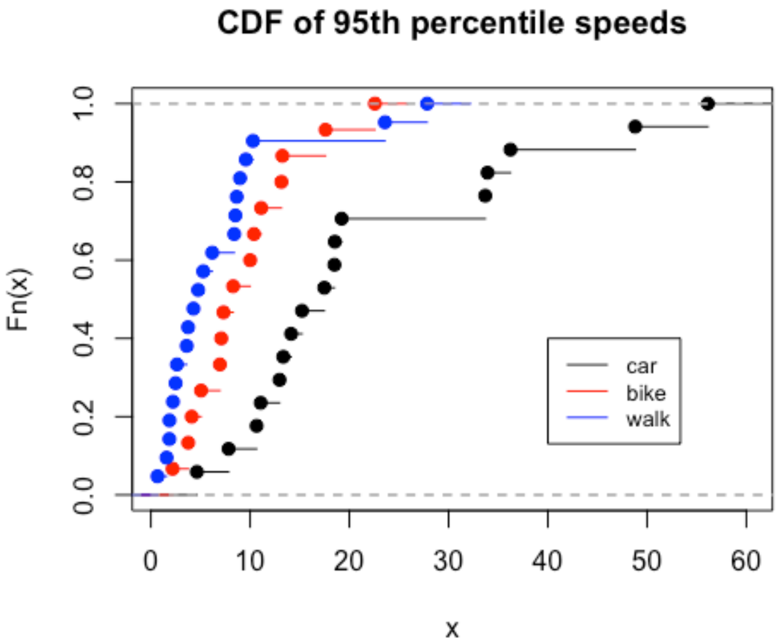


Figure 42 CDF for 95th percentile speeds in private modes in Vienna

When it comes to the trajectories that cannot be determined by proximity to the rail infrastructure and by high speeds of the car modes, RF fares considerably better in correctly identifying the walk modes. However, the recall for walk is lower in RF as it incorrectly identifies more bike modes as walk modes. The FLEL algorithm conversely misidentifies the walk modes as bike modes. Both fare poorly when it comes to identifying car trajectories. The variables used in the RB_FLEL algorithm attempt to distinguish the modes by their proximity to the public transportation network, the upper range of speed and acceleration values produced in the trajectories as well as the median speed. While these variables work well to account for outliers in the data, they may not work so well for the larger overlaps of the variable values across modes, as shown in the CDF plots in Chapter 4, and another example shown in Figure 42.

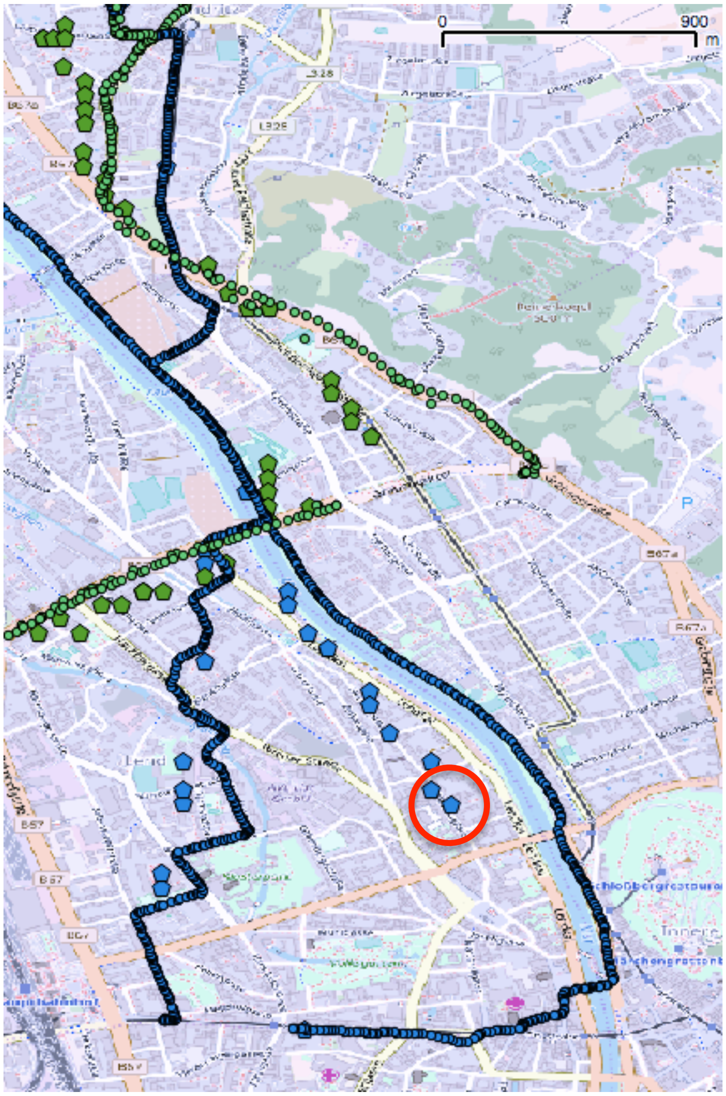


Figure 43 Example of unmatched bike start point (red circle). Small points represent GPS observations, larger pentagons represent CSD observations. Blue indicates bicycle and green indicates cars. (Source: OpenStreetMap)

These overlaps are accentuated even more due to the lower and consequently coarser temporal and spatial resolution. Furthermore, faster modes like cars that make a slower journey or slower modes like bicycles that go at a relatively quicker speed are not that easily distinguished when these percentile values are used.

Another possible reason for why FL may have worked better in GPS studies is that the higher spatio-temporal resolution of GPS data means that the trips that are extracted are more complete in terms of start and end points, as supposed to starting and ending mid-journey. This is very much a possibility in CSD as there may not have been data generated at times corresponding to the start and end of a moving segment. For example Figure 43 show the first point of the extracted bike trajectory coincides with points in the middle of a bike. While the segmentation approach attempts to account for this (segmentation assigns the last and first point of the previous and next stationary segment as the start and end point), this just serves as a stop-gap measure and a lot of information of the actual journey could still be missing. Despite the fact that fuzzy logic systems are meant to be able to account for overlaps and fuzzy sets, these overlaps seem to be almost complete, at least for the variables used. The randomness of RF and resulting diversity of trees leads to a higher performance in this case. The same can be said for the results in Graz, where the RF methods outperform the FL methods, though not by much (Table 20 and Table 21).

To summarize, both Graz and Vienna datasets show that RF outperforms FL. RF is able to better identify modes with a more similar motion profile, more specifically between cars and bicycles. Especially when combined with the RBH step, RF proves to be well suited for this task. However, it is not to say that FL is not a worthwhile option. RB_FLEL trails very closely behind RB_RF for Vienna, which shows that FL is still a feasible choice when it comes to CSD. The next section will explore in further detail the effect the different sets of variables used has on the overall performance of the proposed methods.

5.4.3 Variables for mode detection

This section compares the variable selection methods used in the FL systems, namely those selected by RF with Boruta and variables used in existing studies. Table 25 gives an overview of the variables that are selected by the RF Boruta method for the Graz and Viennese datasets. As the algorithm is iterated for each observation in the LOOCV method, the RF methods (RB_FLRF, RB_RF, RB_FLRF, FLRF) produce a different set of important variables at each iteration, though they are generally similar each time. The frequency of each variable selected is displayed in the table as well. As expected, decile variables like *percentile95acc and percentile95speed* as well as those that account for outliers such as *vel.rolling2.median* and *acc.median* are considered more important as inputs to the classifier than those that do not such as *vel.inst*. This makes sense as the data can still have inaccuracies and errors after the initial cleaning process. The RF variable selection method also seems to select variables that are found in existing literature, namely percentile values of speed and acceleration, as well as some measure of median and average speeds. In addition, it also consistently selects all the spatial features. Notably, there seems to be considerably more variables selected by RF in Vienna than in Graz. This could be due to the more even distribution of trajectories across the modes in Vienna than in Graz, leading to a greater number of variables being identified as important in accounting for these modes. For Graz where the dataset is car and bike heavy, there is an emphasis on velocity measures. This could give an indication that the best way to distinguish between bike and car is through a combination of these variants of velocity measures. Interestingly, RF does not identify *busdist* as an important variable in Graz. Possible reasons could be that bus modes are not present, or that car and bicycle that constitute the majority of the dataset, have trajectories that may coincide rather similarly with bus routes.

RF*				**Existing Literature***	**PCA**
Vienna	**Freq**	**Graz**	**Freq**	**Variables**	**Variables**
vel.rolling2.var	121	*vel.rolling3.median*	181	*percentile95acc*	*vel.var*
vel.rolling2.sd	121	*vel.rolling3*	181	*percentile95speed*	*trip_dist*
ubahndist	121	*vel.rolling2.median*	181	*vel.rolling2.median*	
trip_dist	121	*sbahndist*	181	*total_speed*	
tramdist	121	*percentile85speed*	181		
total_speed	121	*percentile80speed*	181		
sbahndist	121	*percentile70speed*	181		
rolling3_max_speed	121	*acc.median*	181		
rolling2_max_speed	121	*tramdist*	181		
percentile95speed	121	*rolling3_max_speed*	180		
percentile95acc	121	*percentile95speed*	132		
percentile90speed	121	*percentile65acc*	86		
percentile85speed	121	*percentile60speed*	44		
percentile80acc	121	*vel.rolling3.median*	43		
eucdist_speed	121				
busdist	121				
avg_distance	121				

percentile70speed	118		
percentile75speed	116		
percentile65speed	114		
acc.sd.mean	112		
vel.rolling2	104		
num.points	86		
duration	86		*Spatial variables are also included after variable selection
percentile90acc	62		
iqr.vel3	59		
percentile60speed	53		

Table 25 Variables selected for FL by RF and existing literature (in no particular order), and variables selected by PCA for unsupervised methods

Despite the better performance of RF as supposed to FL in both Vienna and Graz, the results show that when it comes tot the task of variable selection, RF is not such a good choice. Figure 34 and Figure 35 give side-by side comparisons of RB_FLEL, RB_FLRF, FLEL and FLRF for both cities. For Vienna, FLEL performs better than FLRF with or without RBH. Even though variables selected by RF contain those in EL, the inclusion of these additional variables derived from RFseem to have a negative effect on the FL system, a trend observed in both cities.

The FL system performed better on CSD when variables obtained from other GPS studies were used as supposed to those that were obtained from CSD itself. This is true despite the fact that the FLRF variables also contained a few of the variables used in FLEL. A common feature of the variables used in each case is they tend to be more robust against outliers or anomalies in the data such as *percentile95acc* or *vel.rolling2.sd*. Measures of maximum speed like *rolling3_max_speed* were identified as important variables but even then, this value was calculated using a rolling window of 3 consecutive observations (Section 4.4.2).

5.5 Unsupervised methods

This section will look at the two unsupervised methods Partioning around Medoids (PAM) and K-means. For PAM, as the dissimilarity measure is derived from the proximity matrix generated by an unsupervised random forest, each set of clusters will be different due to the random subsample of candidate variables considered for each split. While this method does not have a long run time, the process of assigning the modes to the clusters is done manually and is thus time-consuming. As such, the algorithm is run 5 times, and the results are averaged based on these 5 runs. The same is done for the K-means algorithm.

As mentioned in chapter 4, the variables used, as inputs for the clustering will be a combination of those derived from PCA and those from existing literature. For PCA, the variables that were selected in each iteration are very consistent, with *vel.var* and *trip_dist* always being selected (Table 25). When compared to the variables selected by the Boruta method, the only common variable between the two methods is *trip_dist*. This suggests that the spatial variables are generally important for splitting the data into homogenous groups, while *vel.var* and *trip_dist* account for the most variation in the dataset, regardless of the modes taken.

Like the supervised methods, using RBH in the initial step of identifying some rail and car modes with a substantial accuracy leaves less room for error in the secondary step, especially when no labels are used to build or train any model or system. From this set of results, PAM is preferred, as expected due, possibly due to its better ability to handle outliers (Section 4.8). Furthermore, as the cluster centers are medoids, actual observations and not centroids of the clusters, it is easier to assign mode classes. Being represented by actual observations makes them more interpretable and realistic. An example of the cluster centers of the PAM and K-means algorithm can be seen in Table 26 and Table 27**.** For example, if a majority S-Bahn cluster had both S-Bahns and cars, the centroid of the cluster may have a higher *sbahndist* value, but its medoid might have attributes that are more characteristic of S-Bahns. This can make it more challenging to confidently assign that cluster with the mode S-Bahn based on an average of the cluster's points. For the medoids, it is arguably easier to see the clusters that could be majority U-Bahn and S-Bahn clusters from the variables *ubahndist* and *sbahndist.* Conversely, the centroids for the K-means all have relatively high values (>150m) for all their proximity features, making the clusters less characteristic of individual modes. The mode column shows the modes that have been manually assigned to the clusters based on the medoids' and centroids' feature characteristics. While mode assignment was done with 5 modes in mind, the best mode for each cluster was assigned, even if it meant some modes were not represented. For example, in Table 26, both cluster 2 and 4 are assigned as U-Bahn due to the low *ubahndist* values, and as such, car modes are not assigned to any cluster. The same is done for mode assignment in the K-means algorithm.

Medoid	*sbahndist*	*ubahndist*	*total_speed*	*vel.var*	*trip_dist*	*percentile95acc*	*label*
1	88.96	332.94	8.63	504.10	4154.33	7.92	S-Bahn
2	279.00	107.60	2.53	10.19	650.17	0.29	U-Bahn
3	615.00	632.30	5.56	25.08	150.04	1.38	bike
4	1465.62	193.72	4.23	141.55	2028.13	1.90	U-Bahn
5	767.50	146.90	1.32	3.45	200.05	0.06	walk

Table 26 Example of medoids of each cluster for PAM and the mode assigned

Centroid	*sbahndist*	*ubahndist*	*total_speed*	*vel.var*	*trip_dist*	*percentile95acc*	*label*
1	225.35	163.40	4.66	114.12	478.45	1.32	U-Bahn
2	1350.14	174.88	4.83	170.39	1025.60	1.12	U-Bahn
3	787.40	316.22	2.96	17.35	354.77	0.45	Walk
4	398.23	2934.25	4.76	112.66	1345.13	1.08	Bike
5	636.65	728.41	6.12	282.57	6768.38	3.42	Car

Table 27 Cluster centroids for K-means

Table 28 and Table 29 show the confusion matrices as a result of the mode assignment for one of the better performing runs of the RB_PAM and RB_KMEANS algorithm respectively. It is important to note that the clustering was done on the entire dataset, not just those whose modes were assigned in the secondary step (PAM or K-means). For visualization purposes, and to understand the performance of these unsupervised methods, the tables below only show the labels and assigned modes of the trajectories whose modes were assigned in the PAM or K-means step. Just by looking at the clusters formed, it is evident that the RB_PAM algorithm is able to divide the trajectories into more distinct modes as there is less overlap actual modes in the clusters. For example, Table 29 shows that cluster 4 consists of a substantial number of almost all the other modes. Furthermore, looking at the actual labels, it seems that a single mode, like walk, is quite spread across a few clusters (1,2,4 and 5). Arguably, these overlaps occur to a slightly greater extent in the K-means clusters than that of the PAM ones.

It can also be observed that RB_PAM does a better job at drawing distinctions between the various modes, as can be seen in Figure 44. RB_PAM performs better in overall accuracy and has a higher Kappa coefficient, meaning that more of the results can be considered as reliable (Table 30).

PAM	**Assigned mode (cluster number)**			
Label	Bike (3)	S-Bahn (1)	U-Bahn (2, 4)	Walk (5)
Bike	3	0	4	4
Car	4	1	1	0
S-Bahn	0	2	0	0
U-Bahn	2	0	4	3
Walk	3	1	0	14

Table 28 Confusion matrix of trajectories whose modes are assigned by the PAM step

K-means	**Assigned mode (cluster number)**			
Label	Bike (4)	Car (5)	**U-Bahn** (1, 2,)	Walk (3)
Bike	10	0	0	1
Car	4	0	2	0
S-Bahn	2	0	0	0
U-Bahn	7	0	2	0
Walk	9	4	5	0

Table 29 Confusion matrix of trajectories whose modes are assigned by the K-means step

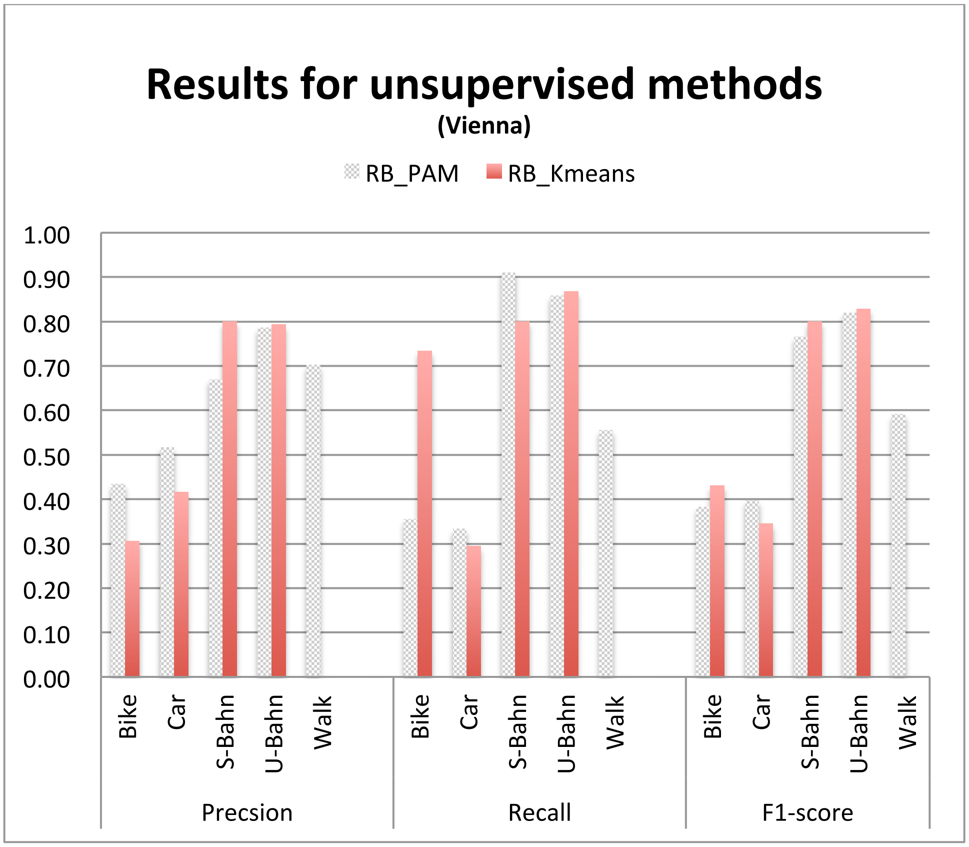


Figure 44 Precision, Recall and F1-score for RB_PAM and RB_KMEANS in Viennese dataset

Vienna	**RB_PAM**	**RB_KMEANS**
Accuracy	0.68	0.61
Kappa	0.55	0.46

Table 30 Performance of unsupervised methods in the Viennese dataset

The clustering however, does not seem to work very well when it comes to distinguishing between bike and car modes, as can be seen in the Graz dataset, where the majority of the trajectories consist of bicycle and car trajectories Table 31. Table 32 and Table 33 Table 33 shows the composition of each of the clusters formed. It illustrates that even if labels were present and the modes could be assigned to clusters to maximize accuracy, this accuracy will still be mediocre as the bike and car modes are quite spread across most of the clusters. Both the PAM and K-means method face a similar issue, though PAM still seems to perform slightly better than K-means.

Graz	RB_PAM	RB_KMEANS
Accuracy	0.56	0.50
Kappa	0.31	0.16

Table 31 Performance of unsupervised methods in the Graz dataset

RB_PAM	Cluster				
Label	**1**	**2**	**3**	**4**	**5**
Bike	4	28	17	25	0
Car	12	14	12	14	14
S-Bahn	1	0	0	0	1
Tram	2	2	1	8	0
Walk	2	2	0	2	2

Table 32 Overview of the composition of clusters generated in PAM on Graz dataset

RB_KMEANS	Cluster				
Label	**1**	**2**	**3**	**4**	**5**
Bike	27	8	20	19	0
Car	8	23	13	11	11
S-Bahn	0	1	0	0	1
Tram	10	1	1	1	0
Walk	5	1	0	0	2

Table 33 Overview of the composition of clusters generated in K-means on Graz dataset

5.6 Summary of main results

This section summarizes the key findings of the results presented in the chapter. The best performing method is RB_RF for both cities and RB_FLEL follows closely for the Viennese dataset. Both datasets indicate that the inclusion of the initial RBH step leads to a positive effect on the overall performance on the method chosen, especially when it is used in conjunction with RF. When it comes to the secondary mode detection step, RF outperforms FL. The benefit of RF over FL is more evident when it comes to the more complicated modes that produce similar values of speed and acceleration; bikes, cars and trams etc. A clear strength of RF is in identifying walk modes, a relatively poorly performing mode for the RB_FLEL method. However when there is a more even split between private and public transportation modes in the dataset, FL performs comparatively well too, as can be seen by RB_FLEL on the Viennese dataset.

For FL methods, the evidence shows that they perform better when variables from EL are used as supposed to that obtained from variable selection through RF. However, both cases use variables that are less sensitive to outliers, giving an indication of what type of variables are suitable to be used when dealing with the noisier CSD.

For unsupervised methods, RB_PAM proves to be a better choice than RB_KMEANS for CSD. By using a combination of variables obtained through PCA and EL, RB_PAM was able to achieve a commendable overall accuracy of 0.68 in the Viennese dataset, which is comparable to even the best accuracy of 0.73. Unsupervised clustering fails when it comes to very homogenous datasets like Graz, where the trajectories are mostly split between the more problematic bike and car modes. Using the RF proximity matrix as an input to measure dissimilarity in the PAM algorithm leads to better mode detection capabilities than when just normal sum of squared distances is used in K-means. However, the manual assignment of modes limits the number of runs that this method can be tested and as such, needs some further work to truly understand its value and shortcomings.

CHAPTER 6
DISCUSSION

This section will discuss the results presented in the previous chapter, in response to the research questions that guide this thesis with the hopes of contributing to the existing work in the field of transportation mode detection using mobile phone data.

6.1 Research Question 1

Development and Implementation: *How can various modes of transportation (walk, bus, tram, car) be detected from cellular signaling data (CSD)?*

From the results presented in Chapter 5, the best method for detecting cellular data is RB_RF for both cities. In Vienna, when non-trips are not considered, the accuracy of the algorithm manages to distinguish between bike, car, S-Bahn, U-Bahn and walk modes up to an accuracy of 73% and Kappa of 0.61. In the Graz dataset, the algorithm manages to distinguish between bike, car, S-Bahn, tram and walk modes for 61% of the trajectories with a Kappa of 0.36. While not as high as that of GPS studies, the results are still comparable to existing research using GPS data, considering the lower spatial and temporal resolution of the data. For example, Rasmussen et al.'s (2015) baseline algorithm manages to achieve an 81.7% accuracy when distinguishing between similar modes and non-trips are not considered. Their baseline algorithm is similar to RB_FLEL. Other GPS studies using RBH achieve accuracies between 70% (Bohte and Maat, 2009), 78-86% (Gong et al., 2012) to more than 90% (Biljecki, 2010). Those that use machine learning methods achieve accuracies up to 93% (Stenneth et al., 2011).

It is challenging to make direct comparisons with existing mobile phone transport mode detection studies due to the differences in data available and validation approach. The data used in this particular thesis is actively sought and, as such, does not follow what would otherwise be a more repetitive pattern of a home to work routine, for example. This means that unlike existing mobile phone studies, OD matrices cannot be extracted and aggregated to any meaningful degree, and it would be futile to compare it with existing census data. For example, Qu et al (2015) used a large amount of data with IDs that did not reset every 24 hours, meaning they were able to extract home-work trips of individuals and this is how the data was cleaned. In that respect, only the home and work trips (derived from the inferred location of home and work, and not directly from any one CDR observation) were considered. The studies are thus only isolated to routine home and work trips. Wang et al (2010) also do the same.

Many existing studies do not have ground truth data to compare their results to, and as a result validate their findings with official census data (Kasahara et al., 2017; Qu et al., 2015; Wang et al., 2010). To our knowledge, Sohn et al. (2006) is the only mode detection study using cellular data that compares the result to ground truth. While the paper achieves 85% accuracy using machine-learning methods, the study only distinguishes between walk, drive and stationary. Because of the improved spatio-temporal granularity as compared to CDR, this introduces greater information to work with. A further significant difference is that these studies group public transportation into one mode (i.e. distinguish between walking, driving and public transportation).

Due to the reasons stated, the methods proposed by this research is a substantial improvement to the state of the art and contributes to the current work in mode detection using mobile phone data, with a greater number of modes being detected with reasonable accuracy.

6.2 Research Question 2

Evaluation and comparison: *How do these proposed methods (RQ1) perform and compare against each other? Which is the best method of mode detection for detecting these modes of transportation?*

This section explores how different aspects of the proposed methods address the spatial and temporal challenges posed by CSD. Consistent with findings from existing studies, the results show that incorporating GIS data in the form of the transportation network in the RBH generally leads to a better overall result. The RBH step only consists of a few rules, and yet is still able to effectively filter out certain modes. Existing RBH studies have a larger variety of rules when GPS data is used, such as individual speed thresholds for each mode. Due to the lower resolution of

CSD data, the trips may not be as intricately represented. Subsequently, the comparatively simpler and rather general set of rules is sufficiently open to encompass the nuances in the data and preemptively extract rail and car modes in CSD. Compared to many GPS studies, the hierarchical RBH detects the slower walking modes first. Due to the lower data quality of lower speed CSD trajectories, a RBH that filters out the quicker modes first is more suitable for this data type. The initial RBH step also means that a certain priority is placed on spatial features for the rail modes, and upper speed thresholds for private car modes. The assumption that proximity to transportation networks takes precedence over many other speed and acceleration variables for rail modes have proven to work in the two datasets.

Between the two secondary mode detection methods, fuzzy logic and random forest, both Graz and Vienna datasets show a better performance of RF than FL, regardless of the variables used in the latter. This is expected as RF has proven to be a popular choice amongst many existing studies. RF is able to better handle imbalanced class conditions, which is usually unavoidable in many real life situations like in this scenario. These generated forests can also be saved and transferred for application to other sets of data (Breiman, 2001). RF seems to do a better job at distinguishing between the more problematic modes, car and bicycles. From the results, it can be observed that small differences that exist in these problematic segments be more accurately defined by the forest generated in RF than by the rules generated by FL. It proves to be even more difficult to generate any sort of hard rule through human reasoning to distinguish between these two modes in the RBH step. As such, RF is a suitable supplementary step to deal with the more complicated differences that the initial human-reasoning-derived RBH cannot handle. With that said, the FL option still proves to be a viable one, with RB_FLEL trailing closely behind RB_RF in performance. In terms of transferability, the fuzzy rules and random forest generated from studies like these have the potential to be applied to other cities, for example. The benefit of FLS as predictive models is how they handle uncertainty that is understandable and more interpretable by humans. Transferability of these fuzzy rules and random forests however, was difficult to investigate due to the different mode compositions of the two datasets.

The FLRF methods were an attempt to customize FL methods to suit CSD. It was expected to perform better than FLEL as variables used in the latter was thought to be more suited to GPS data than cellular phone data, seeing that they were borrowed from existing GPS studies. However, contrary to what was anticipated, variables taken from existing GPS studies seem to lend themselves to better results than those extracted from the CSD through RF. One possible implication of this could be that CSD data is more similar to GPS data than previously thought. Despite the lower data quality of CSD, the distinguishing power of the GPS studies' variables still has a place and can still be used. It is also worth noting that the run time for extracting the variables using RF is substantial, especially when compared to using EL, which has no time cost whatsoever. Each iteration in Boruta requires a large number of trees to be iteratively generated with the inserted shadow variables and this leads to a higher computational complexity in time taken and memory used, making the FLRF option even less desirable.

A further reason these methods are suited to CSD data is the fact that the features are extracted from the segmented trajectories, and not individual observations like in most GPS studies (instant velocity, instant acceleration). There is too much variation in the sampling rate and spatial accuracy for instantaneous features to be used effectively here. This was proven in the variable selection outcomes of PCA and RF, where all the variables that were deemed important or reflected a wide variation in the data were those that were less sensitive to outliers like percentile values. In addition, these percentile values of velocity and acceleration are based on windows of multiple points instead of just instantaneous values (*rolling3.vel* and *inst.acc2*). The RF variable selection method also included all the spatial variables (*ubahndist, sbahndist* etc.) and these values are average distance values of all observations to the corresponding networks. As the spatial accuracy in cities is a lot higher than that of outside cities, these average values for distances to the transportation network seem to be adequate for the effectively detecting rail modes.

Overall, variables that are more suitable to deal with the spatial and temporal characteristics of CSD data are variables that are less sensitive to outliers. The coarser granularities mean the values extracted have a greater range of error. 95th percentiles can be compared to maximum values to shed some information on higher speed modes. By using the percentile values instead of absolute values, relevant information on these maximum speeds can be drawn from CSD. Mode specific variables like distance to the rail network are highly useful for the rail modes. Trip distance is also useful for detecting rail and car modes. The consistently longer distances of these mode types set them apart from the greater variation in distance of the others. For rail modes especially, the user is free to use their mobile phone, increasing the amount of event driven data generated. As a result, less estimation of what is actually happening is required. In cars, it is not uncommon to have navigation apps open and running during journeys. Compared to bike modes, it is difficult, or at least highly discouraged, to use a mobile device whilst riding, limiting the CSD generated to network driven data. This may also be the case for walk trips. The shorter trip distances of these two modes also mean that the network driven data generated is much less than when trip distances are longer, as the chances and frequency of passing through different location areas and switching between cell towers is lower. *total_speed* is another important variable, a velocity measure that aggregates the total distance travelled over the duration of the trip. The inconsistencies of the data make this approach of obtaining average velocity measures more suitable as it evens out the over and underestimations. *inst.vel* that averages all the instantaneous velocity values calculated for the entire trip does not give much information due to the variation and noise in the data. With that said, many cases are not well captured by these variables alone. For example, the faster car modes that make a slower journey or slower modes like bicycles that go at a relatively quicker speed are not that easily distinguished when these percentile values are used.

The poorer performance on the Graz dataset reveals that applying these variables to the fuzzy logic systems still does not uniquely separate all modes; car and bike trips have highly overlapping profiles of these speed and acceleration profiles. This is also seen in GPS studies that use FLS to detect transportation modes (Bolbol et al., 2012; Tsui & Shalaby, 2006). The additional step that Rasmussen et al. (2015) used to combat this problem involves individual stops at bus stops, a level of detail that is not attainable using CSD. A different approach will be required to supplement the proposed CSD methods to separate bike and car trips.

As it is often easier to obtain unlabeled data than labeled data, there is a huge motivation for developing a workable unsupervised method to detect modes of transportation from CSD. The unsupervised methods explored here are a hybrid approach of RBH and two types of clustering; PAM and K-means. The input variables are obtained through PCA. Between the two types of clustering, PAM edges out K-means slightly. As K-means is based on means, it is highly sensitive to outliers. This could contribute to the less distinct clusters produced by K-means. RB_KMEANS performs particularly poorly for the walk modes, though rail modes seem to be comparable to that in RB_PAM. Using RF for the dissimilarity matrix in PAM leads to more meaningful clusters. By using the random forest dissimilarity matrix as opposed to just Euclidean distance, the former is based on the ranks of the variables and is scale independent. As a result, and in agreement with Shi and Horvath (2006), the results show that using the RF dissimilarity in PAM is useful for clustering data into groups that can be interpreted as thresholds, such as the likely maximum distance of *sbahndist* of a S-Bahn segments. Like the supervised methods, the unsupervised methods also perform better on the Viennese dataset than on the Graz, demonstrating that the unsupervised methods work better on distinguishing public transportation modes as opposed to private transportation. Again, this is likely to be because the distinct routes of the rail network sets it apart from the other modes in terms of its spatial distribution, allowing more distinct and meaningful clusters to form on this basis. On top of this, having medoids representing the clusters as supposed to centroids makes the mode assignment more intuitive, which is important as this step is done manually and through human reasoning. While this manual assignment can be perceived as a strength, where human expertise is tapped into to identify nuances that machines may not, the manual assignment of modes is also a source of subjectivity, whereby the capacity of the algorithm to correctly identify modes relies greatly on how well the human expert can assign the modes to the clusters. This becomes especially problematic for cases like the Graz dataset where the majority of modes are bike and car, modes that have similar speed and acceleration profiles.

6.3 Limitations of the study

A major limitation of the study is the amount of data. While there have been huge improvements in accessibility to CSD, the current situation still puts considerable strain on the process of gaining access to CSD datasets, let alone those that are trackable and identifiable to a specific person so that ground truth can be collected. This study and its findings are therefore limited to the ground truth that is provided by the respondents. As the majority of the ground truth data was generated for another study, we had no input on how, when, what and where the data is collected. Furthermore, the low number of relevant data (located in Graz and Vienna) meant there was a need to combine data from two different collection campaigns. The data from both scenarios were different in terms of how they were collected. The former was collected whilst respondents were going about their day, leading to very many similar trips due to routines, and the latter, where the participants were encouraged to collect more and varied types of data across various parts of Austria and to use their devices more to increase the amount of data generated. This could lead to discrepancies in variables like *num.points* within the same modes. Another data inconsistency is that the participants of both data collection campaigns were based in Graz, meaning that their traveling behavior and phone usage patterns are different when in Vienna, a place that is likely to be less familiar. Chances of having navigation apps turned on while driving or walking around the city are higher, leading to a higher frequency and density of observations. It is also possible that different cities have differing degrees of car-friendliness, resulting in different needs for using these navigation systems to aid a journey. For the purposes of this research however, these possible inconsistencies were kept in mind but not considered a big problem as the same inconsistencies would probably be present in real life, in addition to individual differences in phone use. The different labeling procedures of the two data collection campaigns also mean that assumption that no non-trips were generated from the first data collection campaign had to be made. This could have inflated the success of the methods proposed.

The imbalanced dataset might also be a source of bias in the methods. While this was addressed by removing modes with only one or two trajectories, some modes like public U-Bahns in the Vienna dataset and private bike and car modes in the Graz dataset are over represented, leaving the rest underrepresented. Nevertheless, this noticeably different modal make up in the Vienna and Graz datasets make them both good case studies to evaluate the effectiveness of the methods on the various modes; between public and private transportation in Vienna and between the car and bike modes in Graz.

CHAPTER 7
CONCLUSION AND FUTURE WORK

In this thesis, we proposed various novel methods to detect modes of transportation in CSD with inconsistent sampling rates and lower spatial precision of location estimates. The presented approaches consist of hybrid methods combining an initial rule-based heuristic step with a secondary fuzzy logic, supervised random forest or unsupervised clustering step. The variables tested were those used in existing literature as well as data-driven ones extracted from the CSD data using PCA and RF. The results have shown promise in using this newer data type for the purposes of transport mode detection. The best performing method is a hybrid method of RBH and RF, where a success rate of 73% is achieved in the Viennese dataset, a result comparable to existing GPS studies and a step forward in mobile phone data studies in terms of the number of modes detected and the performance achieved. While the results for RB_RF in the Graz dataset is lower at 0.61, RB_RF still outperforms all of the other proposed methods.

Consistent with existing studies, this thesis found the inclusion of RBH to be beneficial for mode detection tasks when it comes to CSD data. Rail modes are reliably detected based on proximity measures and minimum trip distances and certain car modes can be identified with the 95thpercentile speed and acceleration values with relative efficacy. However, with car modes, there is a more delicate balance to be struck when deciding the percentile value that is used in the RBH, as the similarities between certain private modes like cars and bikes are quite substantial. Overlaps in locations of private and public transportation modes like cars, buses and trams make it difficult to use proximity to public transportation network alone to confidently identify bus and tram modes. New methods need to be explored in order to achieve that. While RBH in combined methods leads to reasonable mode detection results, the combination of RBH with RF is more advantageous than the combination of RBH with FL. The randomness of RF makes it more suitable in accounting for the small differences in certain modes.

Furthermore, the study also found that when it comes to speed and acceleration measures, the ones that are less sensitive to outliers are almost always chosen as input variables in all the methods. These, along with spatial proximity features, are regarded as informative and important variables when the goal is to distinguish between modes of transportation with CSD. This thesis also explores the unsupervised alternative in mode detection methods, comparing the ever-popular K-means algorithm with the PAM clustering method. The results presented in the thesis show that PAM is a better fit for CSD. As the PAM uses a dissimilarity measure derived from the proximity matrix from an unsupervised RF, the resultant clusters are more homogeneous in their modes than when K-means is used. Also, as the clusters are represented by medoids in PAM and not centroids, the assignment process becomes much more intuitive and straightforward. However, the process of assigning modes needs to be improved and perhaps automated. The current approach of manual assignment makes this method a considerably bigger undertaking, especially when the algorithm requires several iterations to ensure reliability of the result. That being said, the preliminary results of unsupervised learning for mode detection in CSD shows that there is huge potential in these methods when combined with the initial RBH, which is especially useful due to the hassle of gaining access to this CSD.

The main challenge still lies in the more complicated and problematic modes of bike, car and tram; modes that use the road and will have heavily overlapping speed and acceleration profiles, especially during congestion. As there has been insufficient data on buses, this mode has been left out of the study, but it is likely that it would have performed similarly to the tram modes in Graz. GPS studies have found ways to deal with this similarity in speed and acceleration profiles by incorporating things like bus schedules and bus stops. Future works in this area should add strategies to further distinguish between these complicated modes, keeping the spatial and temporal challenges of CSD in mind. For example, the irregular temporal resolution means that it might be difficult to detect any individual bus or tram stops, as such using live schedules might be more suitable. This would be a bigger undertaking but might be necessary to achieve higher rates of correct mode identification. Future work developing the unsupervised methods should incorporate strategies to automate the process of assigning modes to these clusters so as to increase the stability and lower the sensitivity of these unsupervised methods.

Trip segmentation could be another focus for future work. Depending on the ultimate goal of mode detection, the trade-offs between decreasing the number of non-trips and sacrificing the detection of shorter actual trips must be negotiated and a balance must be struck. For example, when the goal is studying urban activity, activity locations are used as the start and end points of trips (Widhalm et al., 2015). This approach tackles the issue of overlooking short trips that may be part of multi modal trips, effectively reducing the dilemma between minimizing non-trips and maximizing actual trips. Transferability of these methods should also be explored in greater detail. Applications of these methods and their usage will benefit greatly should these algorithms be transferable across cities, making transportation mode detection much more accessible to many more groups. When transferring the model across different scenarios and case studies, it is vital to contemplate the impact of the differences of the built environment, the transportation network etc. These may be significant, or have little effect on the overall algorithm and thus will have a considerable consequence on the generalizability and thus potential users of the proposed methods. Future works could investigate these properties further, and if necessary, proposed improvements to the methods to increase the transferability. This could be in the form of generalizable algorithms or those that use a set of city-specific input parameters that are easily attainable.

While the proposed methods may not perform well in distinguishing between bike and car modes, they are able to distinguish between different categories of modes to a reasonable degree. Depending on the application and intended purpose of mode detection, the data can be processed in a certain way that is useful to the user. For example, if companies collect data from travellers' phones with the intention of estimating road usage, it would be sufficient to differentiate between rail modes, walk and the others (grouping all modes that use the road network, like bikes, cars and trams, together). This level of distinction enables the application to filter out the irrelevant walk and rail modes from their usage reports, and with this purpose in mind, the proposed methods enable such information to be reliable extracted from CSD. Furthermore, despite some modes having an appreciable level of confusion, the methods can still be useful when it comes to individual modes of interest, depending on what the real world application is. The results of this study show that there is huge promise and potential in using CSD for transport mode detection, as a gateway to many other applications.

References

Abdi, H., Williams, L.J., 2010. Principal component analysis: Principal component analysis. Wiley Interdiscip. Rev. Comput. Stat. 2, 433–459. https://doi.org/10.1002/wics.101

Asgari, F., Sultan, A., Xiong, H., Gauthier, V., El-Yacoubi, M.A., 2016. CT-Mapper: Mapping sparse multimodal cellular trajectories using a multilayer transportation network. Comput. Commun. 95, 69–81.

Axhausen, K.W., Schüssler, N., 2009. Processing GPS raw data without additional information. ETH Zurich. https://doi.org/10.3929/ethz-a-005652342

Biljecki, F., 2010. Automatic segmentation and classification of movement trajectories for transportation modes, in: Workshop on Modelling Moving Objects. Informatics Institute, University of Amsterdam, Science Park.

Bivand, R., Keitt, T., Rowlingson, B., Pebesma, E., Sumner, M., Hijmans, R., Rouault, E., Ooms, J., 2018. rgdal: Bindings for the "Geospatial" Data Abstraction Library.

Bivand, R., Rundel, C., Pebesma, E., Stuetz, R., Hufthammer, K.O., 2017. rgeos: Interface to Geometry Engine - Open Source ('GEOS').

Blondel, V.D., Decuyper, A., Krings, G., 2015. A survey of results on mobile phone datasets analysis. EPJ Data Sci. 4, 10.

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. Transp. Res. Part C Emerg. Technol. 17, 285–297. https://doi.org/10.1016/j.trc.2008.11.004

Bolbol, A., Cheng, T., Tsapakis, I., Haworth, J., 2012. Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification. Comput. Environ. Urban Syst. 36, 526–537. https://doi.org/10.1016/j.compenvurbsys.2012.06.001

Breiman, L., 2001. Random Forests. Mach. Learn. 45, 5–32. https://doi.org/10.1023/A:1010933404324

Bro, R., Smilde, A.K., 2014. Principal component analysis. Anal Methods 6, 2812–2831. https://doi.org/10.1039/C3AY41907J

Calabrese, F., Di Lorenzo, G., Liu, L., Ratti, C., 2011. Estimating Origin-Destination flows using opportunistically collected mobile phone location data from one million users in Boston Metropolitan Area. IEEE Pervasive Comput. 10, 36–44.

Calabrese, F., Ferrari, L., Blondel, V.D., 2015. Urban sensing using mobile phone network data: a survey of research. Acm Comput. Surv. Csur 47, 25.

Carter, A., Liddle, J., Hall, W., Chenery, H., 2015. Mobile Phones in Research and Treatment: Ethical Guidelines and Future Directions. JMIR MHealth UHealth 3. https://doi.org/10.2196/mhealth.4538

Chen, C., Ma, J., Susilo, Y., Liu, Y., Wang, M., 2016. The promises of big data and small data for travel behavior (aka human mobility) analysis. Transp. Res. Part C Emerg. Technol. 68, 285–299.

Chi, Z., Yan, H., Pham, T., 1996. Fuzzy algorithms: with applications to image processing and pattern recognition. World Scientific.

Chung, E.-H., Shalaby, A., 2005. A trip reconstruction tool for GPS-based personal travel surveys. Transp. Plan. Technol. 28, 381–401.

Cutler, A., Cutler, D.R., Stevens, J.R., 2012. Random Forests, in: Ensemble Machine Learning. Springer, Boston, MA, pp. 157–175. https://doi.org/10.1007/978-1-4419-9326-7_5

Cutler, F. original by L.B. and A., Wiener, R. port by A.L. and M., 2018. randomForest: Breiman and Cutler's Random Forests for Classification and Regression.

Das, R., Winter, S., 2016a. Automated Urban Travel Interpretation: A Bottom-up Approach for Trajectory Segmentation. Sensors 16, 1962. https://doi.org/10.3390/s16111962

Das, R., Winter, S., 2016b. Detecting Urban Transport Modes Using a Hybrid Knowledge Driven Framework from GPS Trajectory. ISPRS Int. J. Geo-Inf. 5, 207. https://doi.org/10.3390/ijgi5110207

de Montjoye, Y.-A., Hidalgo, C.A., Verleysen, M., Blondel, V.D., 2013. Unique in the Crowd: The privacy bounds of human mobility. Sci. Rep. 3. https://doi.org/10.1038/srep01376

Degenhardt, F., Seifert, S., Szymczak, S., 2017. Evaluation of variable selection methods for random forests and omics data sets. Brief. Bioinform. https://doi.org/10.1093/bib/bbx124

Dudoit, S., ridlyand, J., 2002. A prediction-based resampling method for estimating the number of clusters in a dataset 21.

Elkan, C., Berenji, H., Chandrasekaran, B., De Silva, C., Attikiouzel, Y., Dubois, D., Prade, H., Smets, P., Freksa, C., Garcia, O., others, 1994. The paradoxical success of fuzzy logic. IEEE Expert 9, 3–49.

Faragher, R., 2012. Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation [Lecture Notes]. IEEE Signal Process. Mag. 29, 128–132. https://doi.org/10.1109/MSP.2012.2203621

Fiadino, P., Valerio, D., Ricciato, F., Hummel, K.A., 2012. Steps towards the extraction of vehicular mobility patterns from 3G signaling data, in: International Workshop on Traffic Monitoring and Analysis. Springer, pp. 66–80.

Gong, H., Chen, C., Bialostozky, E., Lawson, C.T., 2012. A GPS/GIS method for travel mode detection in New York City. Comput. Environ. Urban Syst. 36, 131–139. https://doi.org/10.1016/j.compenvurbsys.2011.05.003

Gong, L., Sato, H., Yamamoto, T., Miwa, T., Morikawa, T., 2015. Identification of activity stop locations in GPS trajectories by density-based clustering method combined with support vector machines. J. Mod. Transp. 23, 202–213. https://doi.org/10.1007/s40534-015-0079-x

Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.-L., 2008. Understanding individual human mobility patterns. nature 453, 779.

Gonzalez, P.A., Weinstein, J.S., Barbeau, S.J., Labrador, M.A., Winters, P.L., Georggi, N.L., Perez, R., 2010. Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks. IET Intell. Transp. Syst. 4, 37. https://doi.org/10.1049/iet-its.2009.0029

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182.

Hasan, S., Schneider, C.M., Ukkusuri, S.V., González, M.C., 2013. Spatiotemporal patterns of urban human mobility. J. Stat. Phys. 151, 304–318.

Hastie, T., Tibshirani, R., Friedman, J., 2009a. Unsupervised Learning, in: The Elements of Statistical Learning. Springer New York, New York, NY, pp. 1–101. https://doi.org/10.1007/b94608_14

Hastie, T., Tibshirani, R., Friedman, J., 2009b. Unsupervised learning, in: The Elements of Statistical Learning. Springer, pp. 485–585.

Horn, C., Gursch, H., Kern, R., Cik, M., 2017. QZTool—Automatically Generated Origin-Destination Matrices from Cell Phone Trajectories, in: Advances in Human Aspects of Transportation. Springer, pp. 823–833.

Horn, C., Klampfl, S., Cik, M., Reiter, T., 2014. Detecting Outliers in Cell Phone Data: Correcting Trajectories to Improve Traffic Modeling. Transp. Res. Rec. J. Transp. Res. Board 2405, 49–56. https://doi.org/10.3141/2405-07

Iovan, C., Olteanu-Raimond, A.-M., Couronné, T., Smoreda, Z., 2013. Moving and calling: Mobile phone data quality measurements and spatiotemporal uncertainty in human mobility studies, in: Geographic Information Science at the Heart of Europe. Springer, pp. 247–265.

Jackson, D.A., 1993. Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches. Ecology 74, 2204–2214.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. An Introduction to Statistical Learning, Springer Texts in Statistics. Springer New York, New York, NY. https://doi.org/10.1007/978-1-4614-7138-7

Janecek, A., Hummel, K.A., Valerio, D., Ricciato, F., Hlavacs, H., 2012. Cellular data meet vehicular traffic theory: location area updates and cell transitions for travel time estimation, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing. ACM, pp. 361–370.

Jiang, S., Fiore, G.A., Yang, Y., Ferreira Jr, J., Frazzoli, E., González, M.C., 2013. A review of urban computing for mobile phone traces: current methods, challenges and opportunities, in: Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing. ACM, p. 2.

Kalatian, A., Shafahi, Y., 2016. Travel Mode Detection Exploiting Cellular Network Data. MATEC Web Conf. 81, 03008. https://doi.org/10.1051/matecconf/20168103008

Kaplan, E., Hegarty, C., 2005. Understanding GPS: principles and applications. Artech house.

Kasahara, H., Iiyama, M., Minoh, M., 2017. Transportation mode inference using environmental constraints. ACM Press, pp. 1–8. https://doi.org/10.1145/3022227.3022309

Kaufman, L., Rousseeuw, P.J., 1990. Finding groups in data: an introduction to cluster analysis, Wiley series in probability and mathematical statistics. Wiley, New York.

King, J.R., Jackson, D.A., 1999. Variable selection in large environmental data sets using principal components analysis. Environmetrics 10, 67–77.

Krumm, J., 2009. A survey of computational location privacy. Pers. Ubiquitous Comput. 13, 391–399. https://doi.org/10.1007/s00779-008-0212-5

Kursa, M.B., Rudnicki, W.R., 2018. Boruta: Wrapper Algorithm for All Relevant Feature Selection.

Kursa, M.B., Rudnicki, W.R., others, 2010. Feature selection with the Boruta package. J Stat Softw 36, 1–13.

Leisch, F., Dimitriadou, E., 2018. flexclust: Flexible Cluster Algorithms.

Letouzé, E., Vinck, P., Kammourieh, L., 2015. The Law, Politics and Ethics of Cell Phone Data Analytics. Data-Pop Alliance White Pap. Ser. Data-Pop Alliance World Bank Group Harv. Humanit. Initiat. MIT Media Lab Overseas Dev. Inst. April.

Liao, L., Fox, D., Kautz, H., 2007. Extracting places and activities from gps traces using hierarchical conditional random fields. Int. J. Robot. Res. 26, 119–134.

Liaw, A., Wiener, M., 2002. Classification and Regression by randomForest 2, 5.

Liu, L., Hou, A., Biderman, A., Ratti, C., Chen, J., 2009. Understanding individual and collective mobility patterns from smart card records: A case study in Shenzhen, in: Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference On. IEEE, pp. 1–6.

Maechler, M., original), P.R. (Fortran, original), A.S. (S, original), M.H. (S, maintenance(1999-2000)), K.H. (port to R., Studer, M., Roudier, P., Gonzalez, J., Kozlowski, K., 2018. cluster: "Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.

McHugh, M.L., 2012. Interrater reliability: the kappa statistic. Biochem. Medica 22, 276–282.

Mendel, J.M., 1997. Designing Fuzzy Logic Systems. IEEE Trans. CIRCUITS Syst. 44, 11.

Miao, G., Zander, J., Sung, K.W., Slimane, S.B., 2016. Fundamentals of mobile data networks. Cambridge University Press, United Kingdom.

Milborrow, S., 2017. rpart.plot: Plot "rpart" Models: An Enhanced Version of "plot.rpart."

Montini, L., Rieser-Schüssler, N., Axhausen, K.W., 2014. Personalisation in multi-day GPS and accelerometer data processing, in: 14th Swiss Transport Research Conference (STRC).

National Geospatial-Intelligence Agency, 2017. About NGA [WWW Document]. URL https://www.nga.mil/About/Pages/Default.aspx (accessed 4.14.18).

Pebesma, E., Bivand, R., Rowlingson, B., Gomez-Rubio, V., Hijmans, R., Sumner, M., MacQueen, D., Lemon, J., O'Brien, J., O'Rourke, J., 2018. sp: Classes and Methods for Spatial Data.

Prelipcean, A.C., Gidófalvi, G., Susilo, Y.O., 2017. Transportation mode detection – an in-depth review of applicability and reliability. Transp. Rev. 37, 442–464. https://doi.org/10.1080/01441647.2016.1246489

Qu, Y., Gong, H., Wang, P., 2015. Transportation Mode Split with Mobile Phone Data. IEEE, pp. 285–289. https://doi.org/10.1109/ITSC.2015.56

Rasmussen, T.K., Ingvardson, J.B., Halldórsdóttir, K., Nielsen, O.A., 2015. Improved methods to deduct trip legs and mode from travel surveys using wearable GPS devices: A case study from the Greater Copenhagen area. Comput. Environ. Urban Syst. 54, 301–313. https://doi.org/10.1016/j.compenvurbsys.2015.04.001

Reades, J., Calabrese, F., Sevtsuk, A., Ratti, C., 2007. Cellular census: Explorations in urban data collection. IEEE Pervasive Comput. 6.

Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., Srivastava, M., 2010a. Using mobile phones to determine transportation modes. ACM Trans. Sens. Netw. TOSN 6, 13.

Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., Srivastava, M., 2010b. Using mobile phones to determine transportation modes. ACM Trans. Sens. Netw. 6, 1–27. https://doi.org/10.1145/1689239.1689243

Ricciato, F., Widhalm, P., Pantisano, F., Craglia, M., 2017. Beyond the "single-operator, CDR-only" paradigm: An interoperable framework for mobile phone network data analyses and population density estimation. Pervasive Mob. Comput. 35, 65–82. https://doi.org/10.1016/j.pmcj.2016.04.009

Riza, L.S., Bergmeir, C., Herrera, F., Benitez, and J.M., 2015. frbs: Fuzzy Rule-Based Systems for Classification and Regression Tasks.

Rojas, M.B., Sadeghvaziri, E., Jin, X., 2016. Comprehensive Review of Travel Behavior and Mobility Pattern Studies That Used Mobile Phone Data. Transp. Res. Rec. J. Transp. Res. Board 2563, 71–79. https://doi.org/10.3141/2563-11

Schlaich, J., Otterstätter, T., Friedrich, M., others, 2010. Generating trajectories from mobile phone data, in: Proceedings of the 89th Annual Meeting Compendium of Papers, Transportation Research Board of the National Academies.

Schulze, G., Horn, C., Kern, R., 2015. Map-matching cell phone trajectories of low spatial and temporal accuracy, in: Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference On. IEEE, pp. 2707–2714.

Sevtsuk, A., Ratti, C., 2010. Does urban mobility have a daily routine? Learning from the aggregate data of mobile networks. J. Urban Technol. 17, 41–60.

Shah, R.C., Wan, C., Lu, H., Nachman, L., 2014. Classifying the mode of transportation on mobile phones using GIS information, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, pp. 225–229.

Shen, L., Stopher, P.R., 2014a. Review of GPS Travel Survey and GPS Data-Processing Methods. Transp. Rev. 34, 316–334. https://doi.org/10.1080/01441647.2014.903530

Shen, L., Stopher, P.R., 2014b. Review of GPS Travel Survey and GPS Data-Processing Methods. Transp. Rev. 34, 316–334. https://doi.org/10.1080/01441647.2014.903530

Shi, T., Horvath, S., 2006. Unsupervised Learning With Random Forest Predictors. J. Comput. Graph. Stat. 15, 118–138. https://doi.org/10.1198/106186006X94072

Sohn, T., Varshavsky, A., LaMarca, A., Chen, M.Y., Choudhury, T., Smith, I., Consolvo, S., Hightower, J., Griswold, W.G., Lara, E. de, 2006. Mobility Detection Using Everyday GSM Traces, in: UbiComp 2006: Ubiquitous Computing, Lecture Notes in Computer Science. Presented at the International Conference on Ubiquitous Computing, Springer, Berlin, Heidelberg, pp. 212–224. https://doi.org/10.1007/11853565_13

Stenneth, L., Wolfson, O., Yu, P.S., Bo, X., 2011. 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2011): November 1-4, 2011, Chicago, Illinois.

Stopher, P., Clifford, E., Zhang, J., FitzGerald, C., 2008. Deducing mode and purpose from GPS data. Inst. Transp. Logist. Stud. 1–13.

Tan, P.-N., Steinbach, M., Kumar, V., 2006. Introduction to data mining, 1st ed. ed. Pearson Addison Wesley, Boston.

Tennekes, M., 2018. tmaptools: Thematic Map Tools.

Tettamanti, T., Demeter, H., Varga, I., 2012. Route choice estimation based on cellular signaling data. Acta Polytech. Hung. 9, 207–220.

Therneau, T., Atkinson, B., port, B.R. (producer of the initial R., maintainer 1999-2017), 2018. rpart: Recursive Partitioning and Regression Trees.

Thiagarajan, A., Ravindranath, L., Balakrishnan, H., Madden, S., Girod, L., 2011. Accurate, low-energy trajectory mapping for mobile devices.

Tsui, S., Shalaby, A., 2006. Enhanced System for Link and Mode Identification for Personal Travel Surveys Based on Global Positioning Systems. Transp. Res. Rec. J. Transp. Res. Board 1972, 38–45. https://doi.org/10.3141/1972-07

Wang, H., Calabrese, F., Di Lorenzo, G., Ratti, C., 2010. Transportation mode inference from anonymized and aggregated mobile phone call detail records, in: Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference On. IEEE, pp. 318–323.

Wang, L.-X., Mendel, J.M., 1992. Generating Fuzzy Rules by Learning from Examples 22, 14.

Weiner, E., 1986. Urban transportation planning in the United States: an historical overview (revised edition). Department of Transportation, Washington, DC (USA). Office of the Assistant Secretary for Policy and International Affairs.

Wickham, H., Chang, W., RStudio, 2016. ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics.

Wickham, H., Francois, R., Henry, L., Müller, K., RStudio, 2017. dplyr: A Grammar of Data Manipulation.

Widhalm, P., Yang, Y., Ulm, M., Athavale, S., González, M.C., 2015. Discovering urban activity patterns in cell phone data. Transportation 42, 597–623.

Xiao, G., Juan, Z., Zhang, C., 2015. Travel mode detection based on GPS track data and Bayesian networks. Comput. Environ. Urban Syst. 54, 14–22. https://doi.org/10.1016/j.compenvurbsys.2015.05.005

Zook, M., Kraak, M.-J., Ahas, R., 2015. Geographies of mobility: applications of location-based data. Taylor & Francis.

Personal Declaration

I hereby declare that the submitted thesis is the result of my own, independent work. All external sources are explicitly acknowledged in the thesis.

June 25, 2018
Kimberley Chin