TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

HEIKKI MÄENPÄÄ

USER TRACKING AND INCENTIVE MANAGEMENT IN SMART MOBILITY SYSTEMS

Master of Science thesis

Examiners: Prof. Jose L. Martinez Lastra and Dr. Andrei Lobov
Examiners and topic approved by the Dean of the Faculty of Engineering Sciences
on 29th of March 2017

ABSTRACT

HEIKKI MÄENPÄÄ: User tracking and incentive management in smart mobility systems
Tampere University of Technology
Master of Science thesis, 58 pages, 2 Appendix pages
April 2017
Master's Degree Programme in Automation Technology
Major: Automation Software engineering
Examiners: Prof. Jose L. Martinez Lastra and Dr. Andrei Lobov
Keywords: Machine learning, index structures, travel mode recognition

A system for offering incentives for ecological modes of transport is presented. The main focus is on the verification of claims of having taken a trip on such a mode of transport. Three components are presented for the task of travel mode identification: A system to select features, a means to measure a GPS (Global Positioning System) trace's similarity to a bus route, and finally a machine-learning approach to the actual identification.

Feature selection is carried out by sorting the features according to statistical significance, and eliminating correlating features. The novel features considered are skewnesses, kurtoses, auto- and cross correlations, and spectral components of speed and acceleration. Of these, only spectral components are found to be particularly useful in classification.

Bus route similarity is measured by using a novel indexing structure called MBR-tree, short for "Multiple Bounding Rectangle", to find the most similar bus traces. The MBR-tree is an expansion of the R-tree for sequences of bounding rectangles, based on an estimation method for longest common subsequence that uses such sequences. A second option of decomposing traces to sequences of direction-distance-duration-triples and indexing them in an M-tree using edit distance with real penalty is considered but shown to perform poorly.

For machine learning, the methods considered are Bayes classification, random forest, and feedforward neural networks with and without autoencoders. Autoencoder neural networks are shown to perform perplexingly poorly, but the other methods perform close to the state-of-the-art.

Methods for obfuscating the user's location, and constructing secure electronic coupons, are also discussed.

TIIVISTELMÄ

HEIKKI MÄENPÄÄ: Käyttäjien jäljittäminen ja kannusteiden hallinta älykkäissä liikennejärjestelmissä
Tampereen teknillinen yliopisto
Diplomityö, 58 sivua, 2 liitesivua
Huhtikuu 2017
Automaatiotekniikan koulutusohjelma
Pääaine: Automaation ohjelmistotekniikka
Tarkastajat: Prof. Jose L. Martinez Lastra ja Tri. Andrei Lobov
Avainsanat: Koneoppiminen, hakurakenteet, kulkuneuvon tunnistus

Tässä diplomityössä esitellään järjestelmä tarjoamaan kannustimia ympäristöystävällisten ajoneuvojen käyttöön. Pääpaino on matkan väitetyn kulkuneuvon tai kulkuneuvojen tarkistamisessa. Tarkistamiseen esitellään kolme komponenttia: Menetelmä piirteiden valitsemiseen koneoppimista varten, menetelmä jäljen vertaamiseksi bussiaikatauluun, ja koneoppimisjärjestelmä itse kulkuneuvon tunnistamiseen.

Piirteiden valinta tapahtuu järjestämällä piirteet tilastollisen testiarvon mukaan ja eliminoimalla keskenään korreloivat piirteet. Uusina harkittuina piirteinä esitellään nopeuden ja kiihtyvyyden vinoudet, huipukkuudet, kurtoosit, auto- ja ristikorrelaatiot, ja spektrikomponentit. Näistä vain spektrikomponentit havaitaan hyödyllisiksi.

Bussiaikatauluun vertailuun käytetään uutta hakurakennetta nimeltä MBR-puu (Multiple bounding rectangle). MBR-puu on R-puuhun perustuva rakenne, jossa jäljet lajitellaan jäljen sisältävän suorakulmiosarjan perusteella. Ajatuksen taustalla on tapa estimoida LCSS-mittaa (Longest Common SubSequence) suorakulmiosarjojen avulla. Tämä hakurakenne osoitetaan nopeammaksi kuin lineaarinen haku. Vaihtoehtona esitellään M-puu joka perustuu ERP-mittaan (Edit Distance with Real Penalty) suunta-pituus-kesto-kolmikoiden sarjojen välillä, mutta tämän rakenteen näytetään toimivan huonosti.

Koneoppimiseen harkitut menetelmät ovat Bayes-luokittelija, random forest ja feedforward-neuroverkko autoenkooderilla tai ilman. Autoenkooderilla neuroverkon osoitetaan hämmästyttävän huonosti toimiviksi, mutta muut menetelmät yltävät alan nykytilaa vastaaviin tarkkuuksiin.

Lisäksi työssä esitellään järjestelmät käyttäjän sijainnin peittämiseen ja turvallisten elektronisten alennuskuponkien luomiseen.

PREFACE

This thesis was written as part of the MUSA (MUlti-cloud Secure Architecture) project, in particular the Tampere Smart Mobility Engine use-case.

I'd like to thank my thesis supervisor Andrei Lobov for his guidance in the writing and research process, and for agreeing to grade this thesis in a rush to make it to the next faculty meeting. I'd also like to thank my coworkers in the MUSA project, and FAST laboratory in general. The weekly gatherings FAST M.Sc. Students club helped me keep on track to finishing this thesis, and I'd like to thank everyone who attended. Also, I appreciate everyone who took the time to participate in the travel survey.

Two anonymous reviewers for Transportation Research Part B: Methodological offered valuable insights for the travel mode recognition part of this thesis. Another three from Transportation Research Part C: Emerging Technologies also gave feedback that improved the presentation of the research in this thesis.

I should also acknowledge the OpenStreetMap project for the GPS traces they made available, and the cartography used in figures 6.1, 6.2, 6.3, 6.4, 6.5 and 6.6. The cartography is provided by OpenStreetMap contributors under Creative Commons Attribution Share-Alike 2.0, and the aforementioned figures are therefore also released under the same license.

Tampere, 17.3.2017

Heikki Mäenpää

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS AND SYMBOLS

AED	Average Euclidean Distance, a flexible similarity measure
DTW	Dynamic Time Warping, a flexible similarity measure
ERP	Edit distance with Real Penalty, a flexible similarity measure
GPS	Global Positioning System, a satellite navigation system
HMAC	
LCSS	Longest Common Sub-Sequence, a flexible similarity measure
MBR	Minimum Bounding Rectangle, the series of narrowest minimum/-maximum pairs to contain an object. Also Multiple Bounding Rectangle in the term "MBR-tree".
QR-code	Quick Response code. A type of two-dimensional barcode.

a	The first parameter of an elliptic curve. Also used for a Boneh-Boyen signatures.
b	The second parameter of an elliptic curve.
A^*	The probability that the current mode of transport is classified as A
B_{pr}	The event that the previous mode of transport is B
c_i	an i:th counter in a cryptography-based loyalty point system.
D	A divisor
E	An elliptic curve
$e(P, Q)$	A pairing
$\mathbb{F}_p$	A finite field of order p
$\mathbb{G}_1, \mathbb{G}_2$	Groups of elliptic curve points over finite fields, generated by g_1 and g_2, respectively.
$\mathbb{G}_T$	The codomain of a pairing.
γ	A secret key
k	Order of $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$
λ	The slope of a chord or tangent of an elliptic curve
$\mathbb{M}$	The set of all (considered) modes of travel
μ_Y	Mean of feature vectors belonging to class Y
n_P	A multiplier of point P in a divisor.
ν	The y-intercept of a chord or tangent of an elliptic curve
P	A point on an elliptic curve with coordinates (x_P, y_P)
p	a prime
Q	A point on an elliptic curve with coordinates (x_Q, y_Q)

R	A time series of length n. Also used as a point on an elliptic curve with coordinates (x_R, y_R)
$R_{1..n}$	Series R without its first element
r_0	First element of R
r_i	A random integer used to calculate the i:th counter c_i in an Enzmann-signature.
S	A time series of length m
$S_{1..m}$	Series S without its first element
s	a serial number
s_0	First element of S
Σ_Y	Covariance matrix of feature vectors belonging to class Y
w	A public key

1. INTRODUCTION

The goal of this thesis is a system to offer incentives for ecological modes of transport. The research problem considered is verifying that a journey has taken place, with the stated mode of transport, without unduly compromising the user's privacy.

1.1 Motivation

A certain inability to delay gratification is hard-wired into the human brain[21]. However, the rewards from reducing greenhouse gas emissions will be reaped largely by future generations. For this reason, offering more immediate rewards for eco-friendly modes of transport may be necessary to reduce traffic-related greenhouse gas emissions.

To this end, a system is needed to identify the user's mode of transport, and to confirm that the stated journey has taken place. In particular, it would improve the system's usability if the system could infer the user's mode of transport without the user's input.

A system to actually deliver the incentives to users is also needed. The envisioned system will involve several vendors associated with the service provider, and it would be impractical to individually integrate each vendor's system into the incentive system.

1.2 Travel mode detection

As mentioned in the research problem, a system is required to verify whether or not the user has completed the journey with a given mode of transportat. For this reason, a machine learning system was trained to recognize the user's mode of transport.

Knowing the mode of transport currently employed by the user has many other applications. Among these are tracking the user's exercise goals [33], more accurate

household travel surveys [20] and better informed participant selection for such surveys[36], and even targeted advertisements[41].

The features considered for this task are mainly characteristics of distribution of speed and absolute acceleration, as well as spectral components and auto-and cross correlations of the same.

An algorithm to select a limited amount of features is described in section 5.3. Three different statistical tests are used for the selection of the most significant features for the machine learning algorithm. The tests used are Welch's t-test, Mann-Whitney U-test, and the F-test.

After the feature selection, the features are used to train three different machine learning algorithms. These are four-class feed-forward neural network, a four-class Bayes classifier, and a random forest. To validate the feature selection, a deep autoencoder was trained and used in conjunction with a neural network. The principle behind these classifiers is explained in section 5.2.

The training data consisted of GPS traces of Tampere city buses and GPS traces acquired from the OpenStreetMap and Microsoft GeoLife projects. The preprocessing for this data is described in 3.

The neural network, Bayes classifier and random forest produced roughly comparable results. The autoencoder produced a fairly poor result, and only succeeded in differentiating between muscle-powered and motorized modes of transport. More in depth results are outlined in section 6.2, and discussion of the results is given in section 7.1.

1.2.1 Bus route similarity

Experimentation showed that buses as a mode of transport tend to confound the machine learning algorithms. For this reason, measuring a trajectory's similarity to a bus route could be expected to be useful.

The bus route similarity metric makes use of the flexible similarity measures described in section 2.2. A novel data structure, based on an R*-tree, is used to index these, and is described in section 5.4.

1.3 Incentive management

Incentive management has been utilized in various fields, from customer loyalty programs to peer-to-peer networks. Much research has been done on the subject, and most of it has shown incentive management to be a cost-effective means to the desired end.

Most research on incentive management toward ecological goals has been targeted at business and land owners. In particular, a city in Korea managed to cut the cost of reducing marine litter by a factor of ten by buying fishermen's trash from them[12]. A study in Australia offered land-owners an economic incentive to preserve critically endangered forest types on their land, which resulted in generally positive effect on biodiversity[26].

Research has also been done on using incentives to affect private citizens' behavior. A study of the staff of an institution for mentally retarded children discovered that not only did rewarding lack of unscheduled leave reduce absenteeism, it reduced the disruptive behavior by the patients[16]. This suggests that incentivising one facet of the desired behavior will positively affect the others.

1.4 Hypotheses

Research and experimentation on the subject of travel mode detection yielded the following four hypotheses.

Hypothesis 1 *Frequency-domain features can be used to differentiate between modes of transportation.*

A number of factors could be expected to create fluctuation in the speed at which a person or vehicle is traveling at. For buses, there are stops along the way which would necessitate coming to a full stop to load and unload passengers. All traffic would be expected to pause or slow down at intersections, but cars and bicycles would reach the intersections faster than walkers.

Frequency domain features have been used in travel mode recognition with accelerometer data[24], and on dense GPS data [35]. However, little research was found studying the spectral features of sparse GPS data.

Hypothesis 2 *Auto- and cross correlation of velocity and acceleration can be used to differentiate between various modes of transport.*

A person walking would maintain a fairly steady speed from minute to minute. Similarly, a car that is moving slower than the speed limit would be expected to accelerate, as can a bus at a stop.

Features such as this seemed conspicuously absent from prior research, and are therefore considered here.

Hypothesis 3 *Skewness and kurtosis of speed and acceleration can be used to differentiate between modes of transport.*

A car's velocity distribution would be expected to be skewed toward the speed limit, only dropping for intersections and such, whereas a walker's speed would remain fairly constant and a bicycle would accelerate and decelerate as the road's inclination and other traffic allow.

Lower statistical moments and other features of the distribution of acceleration and velocity have been studied [41, 35, 39]. However, skewness and kurtosis seem not to have been considered as features before.

Hypothesis 4 *Bus route similarity will be an useful tool in differentiating between buses and non-buses.*

Scatter plots from an initial experiment, presented and discussed on page 47, showed that the bus class overlaps with the car and bicycle classes, with the other classes being almost linearly separable.

To verify these four hypotheses, statistical tests will be used to identify the most suitable features, as proposed by Bolbol et al.[9]. A number of features considered by previous research, namely the mean, median and variance of speed and acceleration, will be considered alongside these for reference.

1.5 Structure of this thesis

Chapter 2 presents the theoretical background of this thesis. Section 2.1 discusses the methods of obfuscating an user's location to improve privacy. Section 2.2 introduces the reader to the concept and examples of flexible distance measures. Section 2.3 discusses prior work on travel mode detection. Section 2.4 discusses the M-tree and R-tree. Section 2.5 presents the current state of research into secure mobile coupons, and the mathematical basis thereof. Section 2.6 gives the definition of a smart mobility system.

Chapter 3 pertains to the sources and pre-processing of the training data used for training the machine learning. Section 3.1 introduces the sources of the data. Section 3.2 discusses the issues with the labeling of the data, and the measures taken to circumvent these issues. Section 3.3 discusses the travel survey application created to gather data from within the city of Tampere.

Chapter 4 describes the approach taken to the research problem. Section 4.1 discusses the approach to travel mode detection. Section 4.2 describes the rationale behind creating the MBR-tree. Section 4.3 discusses the approach to incentive management.

Chapter 5 presents the implementation of the above. Section 5.1 will describe the location obfuscation system used. In section 5.2, the inner workings and training algorithms of the used classifiers are described. Some thought is also given to the classification of a continuous multimodal trace. The algorithm for selecting features for the classifiers is described in section 5.3. The MBR-tree is described at length in section 5.4. Section 5.5 describes the initially considered but ultimately unsuccessful second approach to bus schedule indexing, and presents the shared parameters of various flexible distance measures. Finally, section 5.6 will describe the implementation of the incentive management.

Chapter 6 will give the results of experiments run on these. The results of experiments in travel mode recognition are presented in two parts, first the feature selection experiments in section 6.1, and then the actual classification results in section 6.2. Section 6.3 will present the performance metrics of the bus route indexing methods.

Chapter 7 will give discussion on the results. Section 7.1 will discuss the travel mode detection, and section 7.2 the bus route indexing. Finally, section 7.3 will discuss the results of the travel survey.

Conclusions and future work will be given in chapter 8.

2. BACKGROUND

2.1 User Tracking

A major issue in user tracking is privacy. Most mobile app users are uncomfortable with letting an app track their precise location.[19, 34]

As privacy safeguards, the following methods have been used in the past:

Perturbation The introduction of random or pseudorandom noise to the location signal to mask the signal.[37]

Cloaking region Reporting the user's location only at precision low enough to always match at least k other users' location. [22]

Silent period Tracking is stopped at certain times or in certain areas, and resumed with a new identifier when the period expires or the user leaves the area.

On the other hand, systems have been proposed to track an user's location from very rough data. One method involves repeatedly querying a routing engine to discover a likely route traveled, using cell tower's signal strength to produce a rough fix[4].

2.2 Flexible similarity measures

A trajectory is the path that a moving object follows through space as a function of time. Trajectory recognition has been considerably successful in handwriting recognition[30]. It has also been used to index videos according to the movement of objects on screen[25]. Also, much like any pair of real numbers can be expressed as a point in a two-dimensional plane and vice versa, much of trajectory recognition can be generalized to time-series.

For the purposes of this thesis, a similarity measure between two trajectories is sufficient and necessary. Magdy et al.[27] made a broad literature review, and produced a taxonomy of measures for trajectory similarity.

Of particular interest in this thesis are flexible similarity measures, or measures intended to gauge the similarity of time series of different lengths. Flexible similarity measurements tend to adhere to the template

$$dist(R,S) = min \begin{cases} f(r_0, s_0) + dist(R_{1..n}, S_{1..m}) \\ g(r_0) + dist(R_{1..n}, S) \\ g(s_0) + dist(R, S_{1..m}) \end{cases} \tag{2.1}$$

Where R is a time series of length n, S is a time series of length m, r_0 and s_0 are the first elements of the respective series, and $R_{1..n}$ and $S_{1..m}$ are the time series with the first element removed. Function f is the cost of altering one argument to the other, and function g is the cost of deleting the argument.

Where flexible similarity measures differ is the sorts of changes allowed, and the cost function for these changes, and the distance between and empty and non-empty series. The plain edit distance simply counts the amount of changes. A slightly more refined measure, Edit distance with real penalty (ERP) also factors in the magnitude of these changes[10], and some variants of dynamic time warping account for the distance in time[29].

Most flexible similarity measures require recursion over every possible combination of the edit actions to find the cheapest one. This means that it is necessary to have a lower bound for these. Marteau[29] recommends that the metric used should satisfy the triangle inequality $||a-c|-|c-b|| \leq |a-b| \leq |a-c|+|c-b|$. Using this inequality, the difference between distances to a trivial case such as an empty sequence can be used to establish a minimum distance, permitting pruning recursions that can not produce a smaller distance than already completed recursions.

It is useful if an edit distance is a metric. The definition of a metric is

$$\begin{aligned} d(x,y) &\geq 0 && (2.2) \\ d(x,y) = 0 &\Leftrightarrow x = y && (2.3) \\ d(x,y) &= d(y,x) && (2.4) \\ d(x,z) &\leq d(x,y) + d(y,z) && (2.5) \\ & && (2.6) \end{aligned}$$

If the similarity measure used is a metric, the set of trajectories becomes a metric space which, among other things, can be indexed with an M-tree.

2.2.1 Edit distance

Edit distance measures the amount of elements that need to be changed in one time series to match another. This is not a metric, which is why Edit distance with real penalty is used in this thesis. Edit distance with real penalty(`ERP`)[10] is a similarity measurement for two time series R and S, of the form[10]

$$erp(R,S) = min \begin{cases} f(r,s) + erp(R_{2..n}, S_{2..m}) \\ f(r,g) + erp(R_{2..n}, S) \\ f(s,g) + erp(R, S_{2..m}) \end{cases} \tag{2.7}$$

Where the element g is a gap element, or an element used as a reference point.

In the special case that one series (marked below with S) is empty, the distance becomes

$$erp(R,S) = \sum_{i=0}^{n} f(r_i, g) \tag{2.8}$$

Where r_i is the i:th element of R, which is the non-empty series. Intuitively, it follows that if both sets are empty, the distance is zero.

ERP has been shown to satisfy the triangle inequality [10]. Therefore, the triangle inequality can be used to eliminate recursions with help of a reference value. The simplest reference value, and the one used in this paper, is an empty series.

The function is also trivially symmetric as long as $f(r,s)$ is a metric. It is also plain that $erp(S,R) = 0 \equiv S = R$ if f is a metric. Therefore, ERP is a metric. This enables the use of an M-tree to index bus schedules.

2.2.2 Longest Common Sub-Sequence

Longest common sub-sequence (LCSS) is a special case of edit distance that discovers the length of the longest sequence that can be converted into either of the sequences

compared through insertions. This is not a proper metric, either, but there is a method to find a lower bound for it[40].

The lower bounding method is based on sequences of bounding rectangles, and is the basis of the MBR-tree.

The equation for LCSS is

$$lcss(R,S) = \begin{cases} 1 + lcss(R_{2..n}, S_{2..m}) & \text{if} r = s \\ max \begin{cases} lcss(R_{2..n}, S) \\ lcss(R, S_{2..m}) \end{cases} & \text{otherwise} \end{cases} \tag{2.9}$$

2.2.3 Dynamic Time Warping

Dynamic time warping (DTW) stretches and contracts the elements of one series to fit the other series as close to another as possible. It is challenging since it neither satisfies the triangle inequality nor has a finite trivial case [23, 29].

DTW's main strength is that it matches sequences that are out of phase in time[23], which makes it useful for measuring similarity of a segment of a bus route to the entire bus route.

Methods to find a lower bound for a DTW distance have been developed[23].

The equation for Dynamic Time Warping is as follows

$$dtw(S,R) = min \begin{cases} f(r_0, s_0) + dtw(R_{1..n}, S_{1..m}) \\ dtw(R_{1..n}, S) \\ dtw(R, S_{1..m}) \end{cases} \tag{2.10}$$

2.2.4 Average Euclidean Distance

In this thesis, a flexible metric called Average Euclidean Distance (AED) is used as a metric unrelated to LCSS, DTW or ERP, for comparison of the indexing methods' performance. This is similar to the modified Hausdorff distance used by Atev et al.[5], except instead accounting for the order this metric accounts for the time coordinate.

The distance is calculated as follows:

$$aed(R,S) = \frac{\sum_{i=1}^{n} min_{j\in[1,m]}(d(s_i, r_j))}{n} \tag{2.11}$$

where the function d is a weighed sum of spatial distance and difference in time of day.

This is not transitive if $m \neq n$, although this can be worked around as defining the output as $aed(S,R)$ if $n > m$. A counter-example to the triangle inequality can be constructed by taking two distinct sub-sequences of a sequence, and measuring the AED between the three sequences. This counter-example also disproves $aed(R,S) = 0 \Leftrightarrow R = S$. Therefore, AED is not a metric.

2.3 Travel mode recognition

Travel mode recognition refers to classifying the kind of vehicle a trip has been made on. This has been of interest for various application, and there is a large body of research available on the subject. For instance, Su et al.[38] made a fairly broad review of the literature in 2014, with an eye to using the entire sensor suite of a smartphone for this task. Their study did not touch on the subject of using only sparse GPS data.

Bolbol et al. have both shown that a 30 to 60 second sampling period is sufficient for travel mode recognition[8], and provided an useful methodology for the selection of most discriminating features[9]. Another purely GPS study was carried out by Gong et al.[20] who combined the GPS data with information about public transport stops in order to identify modes of transport in New York City.

State of the art for travel mode recognition is a 2016 study by Zhu et al.[41], in which a deep learning algorithm achieved a 93% overall accuracy. They compared their algorithm, a deep neural network with a stacked auto-encoder with several other state of the art methods. A similar study from eight years before[35] had found that a decision tree combined with a hidden Markov model performed best.

A study comparing a Bayes classifier, a feedforward neural network and a random forest has been submitted to Transportation Research Part C[31]. The findings of the article are contained in this thesis, particularly in sections 1.2, 5.3 and 6.2

A common feature of studies on travel mode recognition is that walking can be easily identified. The other modes of transport tend to be harder to discern.

2.4 Index structures

2.4.1 M-tree

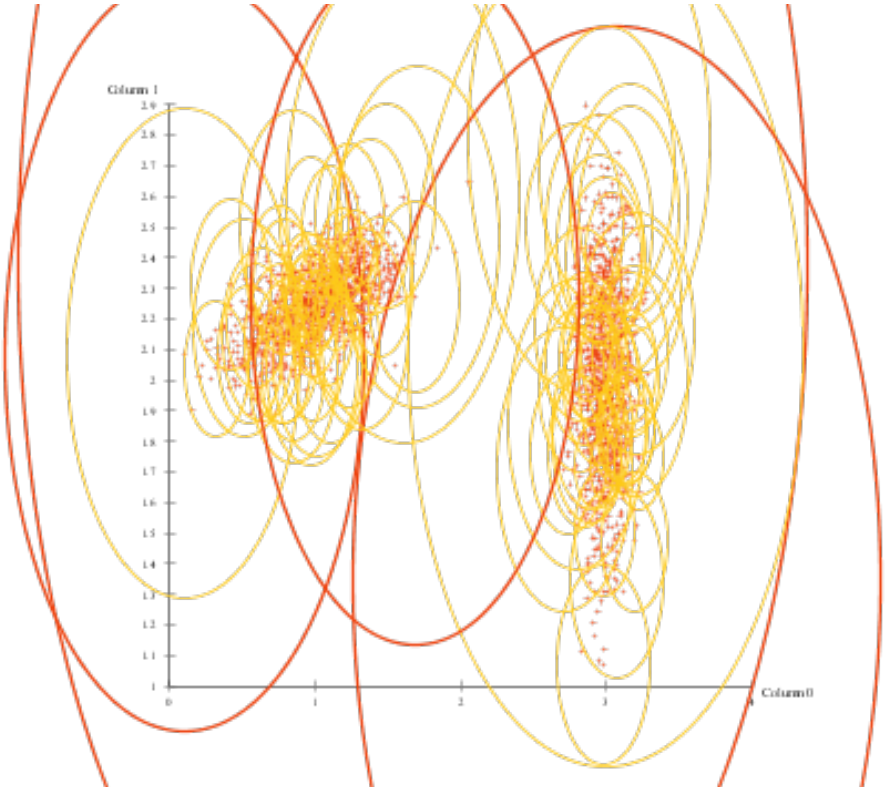

Figure 2.1 *An M-tree constructed with random data. Index spheres appear elliptical due to scaling.*

An M-tree is an data structure for indexing metric spaces[13]. Nodes of an M-tree are defined by a centroid and a radius. The centroid is a reference element, and the radius is the maximum distance the node's descendants will be from the centroid, as determined by the associated metric.

Figure 2.1 shows an M-tree constructed out of two-dimensional random data. The circles represent the index spheres, with red as the first layer and yellow as the second layer. The index spheres appear elliptical due to the coordinates' scaling. The rash of red points almost obscured by the index spheres is the underlying data. As can be seen, the index spheres overlap quite a bit, and overflow the training data's boundaries.

In this thesis, a metric space of bus routes measured with ERP will be indexed with M-tree.

During insertion, the element is inserted into the subtree whose centroid is closest to the element, recursively until the correct leaf node is found.

If a node's child count is above a pre-set maximum, the node is split. In a split, two child nodes are selected, and their centers are used as the centers of new nodes. The split node's children are then inserted under the new nodes, and the new nodes are inserted into the split node's parent.

Dividing a metric space that is not spatial is not an intuitive process, since things like coordinates do not exist. The implementation used in this thesis splits the node by selecting the child node furthest from the parent's center as the first pseudo-medoid, and then selecting the child node furthest from the first one as the second pseudo-medoid. The children are then divided according to the pseudo-medoid closest to them, and the medoid of each group is set as the corresponding new node's center.

2.4.2 R-tree

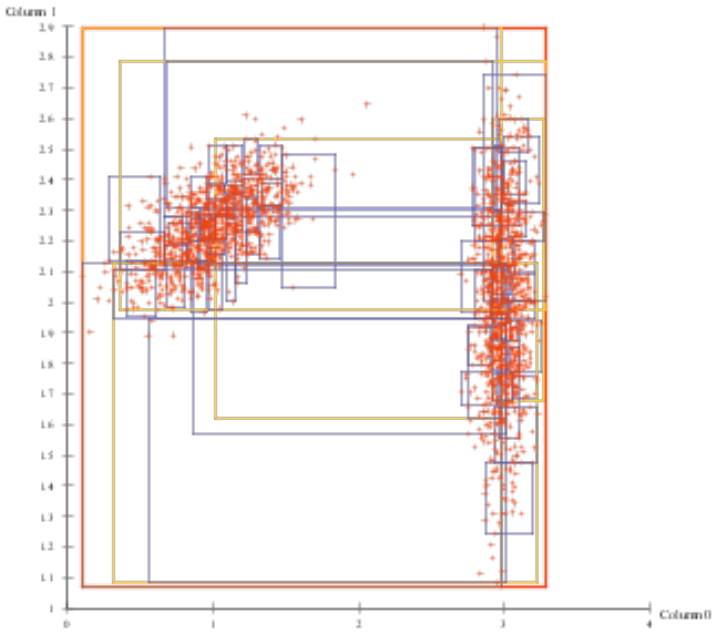

Figure 2.2 *An R-tree constructed with random data*

The R-tree is the basis on which the MBR-tree is built. An R-tree is a data structure for storing spatial data[6]. Nodes of an R-tree are defined by their bounding rectangles. The R*-tree is a particularly well-performing variant of the algorithm.

Figure 2.2 shows an R*-tree constructed out of random data. The large red rectangle represents the parent node of the whole tree, the yellow rectangles are the first layer

and the blue rectangles the second layer. As can be seen, there is some overlap in the index rectangles, but not much. The rash of red points is the underlying data.

During insertion, the element is inserted to a sub-tree selected depending on whether the sub-tree contains only leaf nodes or other branches.

If the sub-tree contains other branches, the element is inserted into the sub-tree whose bounds require the least expansion. If the sub-tree contains only leaf nodes, the sub-tree whose bounds need to be expanded to add the least overlap with its neighbors is selected.

If a node's child count is over a pre-set number, the node is either split or undergoes forced reinsertion. During a split, the coordinate axis is found for which the total width of the child nodes' bounds is smallest, and the child nodes are ordered according to their bounds' position on the axis. The children are then divided at the location where the division results in the least overlap between the two new nodes, with ties broken by lesser total volume.

Forced reinsertion is carried out the first time a node on a given level of the tree is split during an insertion. In a forced reinsertion, the node's children are sorted by their bounding rectangle's center's distance to the node's bounding rectangle's center, and the most distant child nodes are removed and inserted back into the tree. Beckmann et al. found that the amount of reinserted nodes should be 30% of the node's size, and that the best performance is achieved when the reinsertion order is in ascending order of distance.

2.5 Incentive management

The incentives the system will offer are discount coupons. Most research on electronic coupons has focused on creating privacy-preserving schemes that permit coupons to be validated without the issuer being able to track the coupons.

Nguyen[32], whose work built on Chen et al. [11], approaches the issue through pairing-based cryptography. Enzmann et al.[17] used elliptic curve cryptography. Since cryptographic pairings only exist (so far) for elliptic curve points, it is safe to say that elliptic curve cryptography is a common theme in the literature.

2.5.1 Elliptic curve cryptography

An elliptic curve is of the form

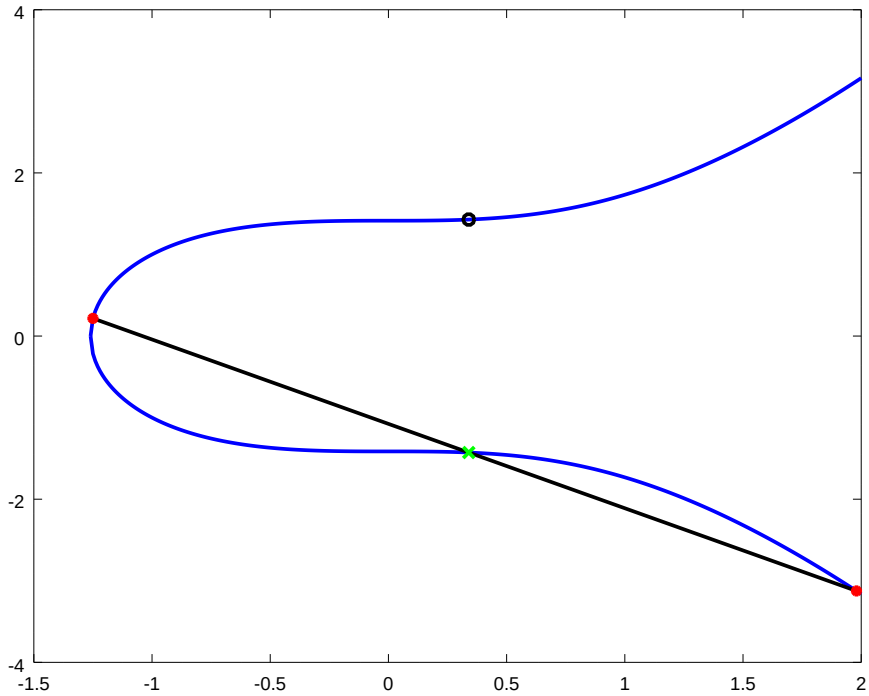

Figure 2.3 *Illustration of the chord and tangent rule*

$$y^2 = x^3 + ax + b \tag{2.12}$$

The field law for points on an elliptic curve is defined by the chord and tangent rule: draw a line passing through the two points (or a tangent at the single point being squared) and negate the y-coordinate of the third point the line passes through. If there is no such third point, the result is a point at infinity $\mathcal{O}$, which is considered to be directly above any point.

This is illustrated in figure 2.3. The two round dots in red are the operands, the green cross is the third point, and the black circle is the end result.

The chord and tangent rule can be expressed mathemathically as

$$x_R = \lambda^2 - x_P - x_Q \tag{2.13}$$
$$y_R = -(\lambda x_R + \nu) \tag{2.14}$$

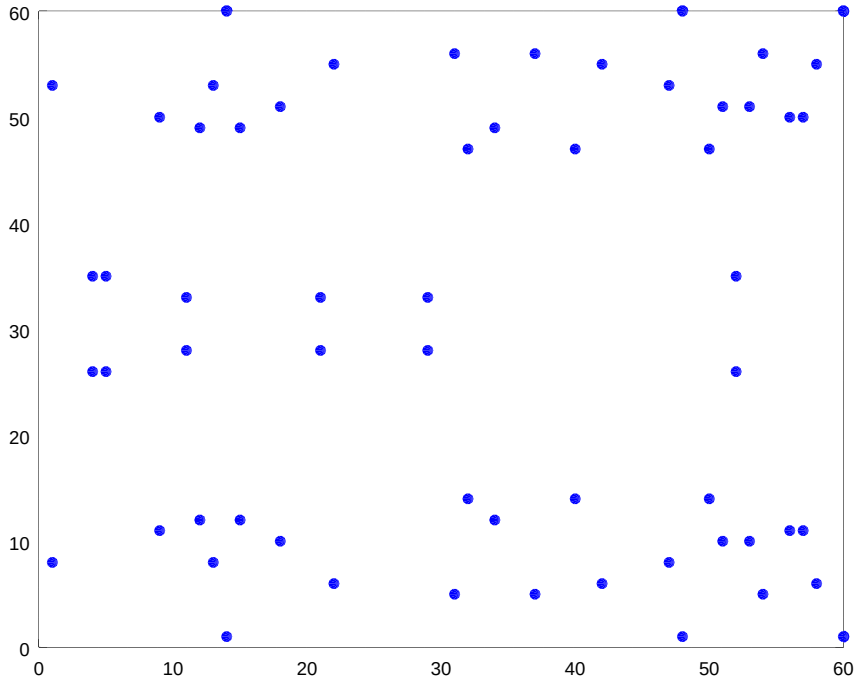

Figure 2.4 *The elliptic curve from figure 2.3 over* $\mathbb{F}_{61}$

Where the chord or tangent's equation is $y = \lambda x + \nu$, the operands are (x_P, y_P) and (x_Q, y_Q) and the output is (x_R, y_R).

For cryptographic purposes, elliptic curves over finite fields are used. In finite fields of prime power, equation 2.12 turns to

$$y^2 \equiv x^3 + ax + b \bmod p \tag{2.15}$$

for a finite field of prime order p. Equations 2.13 and 2.14, and the chord's formula, are similarly altered. Figure 2.4 shows a plot of elliptic curve points over $\mathbb{F}_{61}$, which is illustrative of the magnitude of the effect of these changes.

Enzmann et al.'s[17] signature method is as follows: to initialize the system, the vendor selects a secret key γ, an elliptic curve point g, and a public key $w = g^\gamma$. The user selects a serial number s for their "zero-counter", and hashes it to the same elliptic curve to produce $c_0 = H(s)$.

To increment the counter, the customer selects a random scalar r_i, and calculates $b_i = c_{i-1}g^{r_i}$, which they send to the vendor. The vendor calculates b_i^γ, and returns that to the customer, who calculates

$$c_i = b_i^\gamma V^{-r} = c_{i-1}^\gamma g^{\gamma r_i} g^{-\gamma r_i} = c_{i-1}^\gamma \tag{2.16}$$

When the customer wishes to redeem the points, they send c_n, n and s to the vendor, who verifies $c_n = H(s)^{\gamma^n}$.

2.5.2 Pairing-based cryptography

At time of writing, pairing-based cryptography is a subset of elliptic curve cryptography. It forms the basis of Nguyen's[32] electronic coupon scheme.

A pairing is a non-degenerate bilinear function $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, i.e. it satisfies the conditions

$$\begin{aligned} e(P, QR) &= e(P, Q)e(P, R) && (2.17)\\ e(PQ, R) &= e(P, R)e(Q, R) && (2.18)\\ e(P, Q) &\neq 1 && (2.19) \end{aligned}$$

It follows from equations 2.17 and 2.18 that $e(P^n, Q^m) = e(P, Q)^{nm}$. Because of the requirements, the definitions of pairings tend to become complicated.

In particular, the pairing definitions rely on divisors, which are a "convenient way to denote a multi-set of points"[14] on an elliptic curve over a finite field. A divisor is written as the formal sum

$$D = \sum_{P \in E(\mathbb{F}_p)} n_P(P) \tag{2.20}$$

If D is the divisor of a function, a positive n_P indicates a zero of order n_P, and a negative n_P a pole of order $-n_P$. A function for elliptic points is evaluated at a divisor as

$$f(D) = \prod_{P \in E(\mathbb{F}_p)} f(P)^{n_P} \tag{2.21}$$

Pairings rely on evaluating a function, whose divisor is defined according to the other operand, at one operand (and vice versa, in the case of the Weil pairing).

As an example of the use of pairings in cryptography, the Boneh-Boyen signature is calculated as follows: g_1, g_2 are the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, γ is the secret key, $w = g_2^{\gamma}$ is the public key and r is the message. The signature $a = g_1^{\frac{1}{\gamma+r}}$, and it is verified by $e(a, wg_2^r) = e(g_1, g_2)$.

To compute $g_1^{\frac{1}{\gamma+r}}$, the following must be done: since $\mathbb{G}_1$ is of order k, find l so that $\frac{lk+1}{\gamma+r}$ is an integer, which would leave the verification equaling $e(g_1, g_2)^{lk+1}$. Because $\mathbb{G}_T$ is also cyclic with order k, this is equal to $e(g_1, g_2)$.

l can be found by the following:

$$\begin{aligned} lk + 1 &\equiv 0 \bmod \gamma + r && (2.22)\\ lk &\equiv -1 \bmod \gamma + r && (2.23)\\ l &\equiv -k^{-1} \bmod \gamma + r && (2.24) \end{aligned}$$

The signature scheme of Nguyen[32] is considerably more involved.

2.6 Smart Mobility Systems

Smart mobility systems are services that provide various types of real-time information on the traffic conditions, to the citizenry at large [28]. The information provided is typically aggregated from multiple sources. The EU directive 2010/40/EU defines Intelligent Transport Systems as "...advanced applications which without embodying intelligence as such aim to provide innovative services relating to different modes of transport and traffic management and enable various users to be better informed and make safer, more coordinated and 'smarter' use of transport networks"[18].

2.7 Summary

Random forest classification has been found to perform well in travel mode recognition, as have neural networks[41, 35]. In this thesis, these two methods will used

as a classification

To measure bus route similarity, an index of bus routes will be used. For this purpose, an index structure based on the LCSS estimation by Keogh et al.[23] and the R-tree will be used. For comparison, an M-tree will also be used. The performances of these will be measured by LCSS, DTW and AED between the query and the 5NN result.

The incentive management portion of this thesis will make use of elliptic curve cryptography.

3. DATA ACQUISITION AND PREPARATION

In order to train machine learning algorithms, a large data set is generally required. For this reason, GPS traces for each mode of transport were acquired from three sources: The OpenStreetMap project[3], Microsoft's Geolife[2], and the Innovative Tampere Site[1].

3.1 Sources of data

OpenStreetMap is an open-source cartography project. Its purpose is to provide and maintain an open-data map of the world[3].

The project has amassed a large collection of GPS traces, some of which were tagged with a mode of transport. Unfortunately, some of the traces were tagged with more than one mode so an initial heuristic was used to reduce the amount of false positives.

Microsoft GeoLife is a location-based social networking service[2]. The purpose of the service is to mine multiple users' data for typical travel sequences and to use individual location histories to measure similarity between users and provide friend- and location recommendations.

The project has made the GPS traces of 182 users from a period of April 2007 to August 2012 available. The dataset was collected by Microsoft Research Asia.

The Innovative Tampere Site provides, among other things, real-time data on bus locations[1]. These were logged for three hours on a working day to collect bus traces.

3.2 Pre-processing

To improve the trained system's reliability for short segments, and to further increase the size of the training set, the traces were split into ten minute chunks. Tracks were also split if the location didn't move $50m$ in three minutes, corresponding to $1^{km}/_{h}$

For each mode of transport, a trace was rejected if the speed, inferred from location change and timestamps, stayed over v_{max} for thirty consecutive seconds. Also, if the inferred speed did not exceed v_{min} for thirty consecutive seconds at some point, the trace was rejected. Individual track points were rejected as corrupt if the inferred speed was over $200^{km}/_{h}$, and the entire trace was rejected if there were ten consecutive corrupt track points. The parameters are in table 3.1.

Mode	v_{max}	v_{min}
Bicycle	$40^{km}/_{h}$	$10^{km}/_{h}$
Walking	$15^{km}/_{h}$	0
Car	$80^{km}/_{h}$	$40^{km}/_{h}$

Table 3.1 *The parameters for initial classification*

If any trace was over three standard deviations away from the mean for any of the features described in 5.3, they were discarded as outliers. The final filtration results are in table 3.2. Finally, because the walking class was excessively large, it was truncated to two times the size of the car class before eliminating outliers.

For the initial experiment, five percent of the size of the car class, 70 traces, were set aside as a final validation set and to draw visualizations of the results.

Mode	N	$v > v_{max}$	$v < v_{min}$	corrupt	3σ	final
Bicycle	3086	384	276	161	564	2449
Walking	10438	144	0	180	550	2250
Car	1403	1580	24369	3417	168	1094
Bus	1826	0	0	0	239	1465

Table 3.2 *Number of accepted and rejected segments by mode and rejection criterion*

Of the filtered traces, 453 walking-, 107 bicycle- and 40 car traces were within Tampere and could be used to train classifiers in conjunction with the MBR tree. For this experiment, each class was trimmed until it was no more than twice the size of the smallest class, which were the car traces.

3.3 Travel survey application

Because the OpenStreetMap project's GPS traces were from all around the world, and the Geolife traces were from Beijing, China, they could not be used to train the travel mode detection algorithm in conjunction with the bus route similarity measurement. For this reason, an Android application was built, with a corresponding server backend.

The application collected the user's location data from the system's location service. It then used a neural network trained without bus route similarity to gauge when the user had changed mode of transport. The user was then given the opportunity to correct any mislabeled segments before finally submitting the data to a central server, where the traces were anonymously stored.

The central server was built with Node.js. The traces were anonymized by generating a pseudorandom filename for each uploaded trace. Requests to download traces were authorized by HMAC[7] with a hard-coded application secret as the key.

The participants pool consisted of the employees at FAST laboratory at TUT, and the students of a single course in Factory Automation and Industrial Informatics.

4. APPROACH

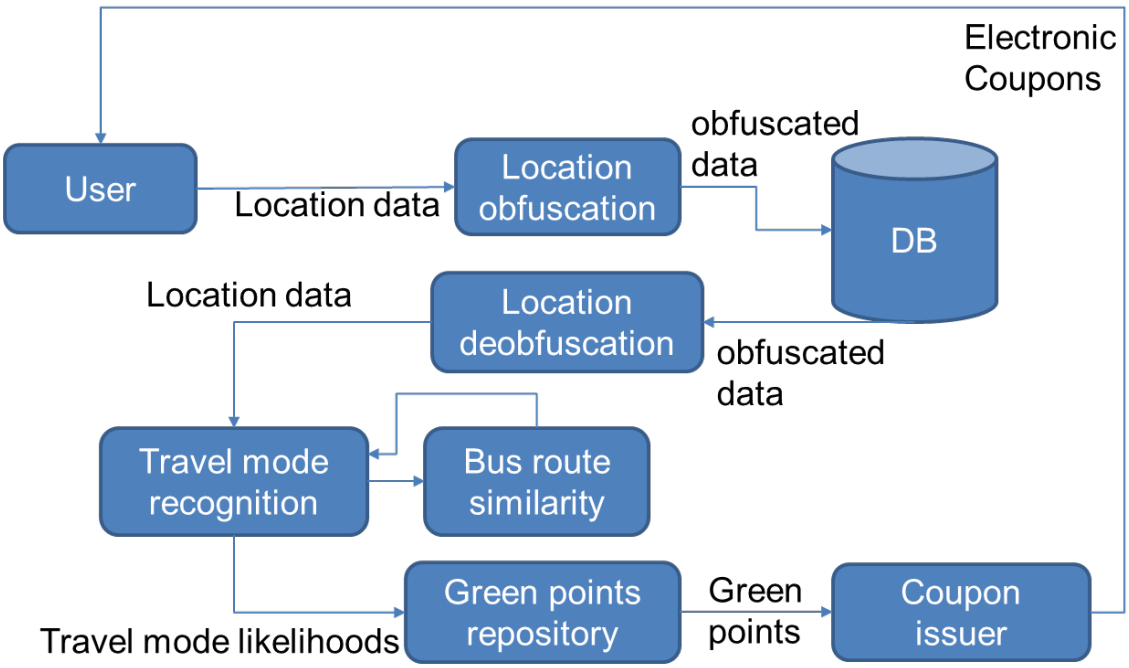

Figure 4.1 *The overall architecture of the system*

The overall architecture of the system is given in figure 4.1. The arrows represent the flow of data, and are labeled with the type of data carried. The blocks represent the modules of the system. Requests sent by the user have been omitted for clarity.

The user's data is passed through location obfuscation into the DB, from where it will be read with deobfuscation when travel mode recognition becomes necessary. The travel mode recognition consults bus route similarity for one feature in the recognition task, and passes the likelihoods of each travel mode to the green points repository which allocates green points according to the "greenness" of the trip. When the user wishes to redeem green points for real world benefits, the green point repository asks the coupon issuer to issue an electronic coupon to the user.

4.1 Travel mode estimation

The experiments in this thesis were carried out using reasonably precise and accurate GPS data. Such data is also required for the bus route similarity measurement. These two factors necessitate a reversible obfuscation of location data.

4.2 Bus route similarity

During experimentation with the travel mode detection algorithms it became apparent that buses are difficult to differentiate from the other modes of transport, and especially that the other three modes considered were fairly distinct. For this reason, a bus route similarity metric is needed.

The main weakness of indexing a sequence of MBRs in a regular R tree, as in [40] is that it is sensitive to the starting times and/or -points of the traces being compared. In the case of bus-route similarity measurement, the start of the query trace will very likely be after the start of the scheduled trace, which would result in the first bounding box of the query trace corresponding to the middle of the bounding boxes of the scheduled trace, which in turn would tend to produce a large distance between the coordinates corresponding to the first bounding boxes.

A different granularity in the query trace and the scheduled trace could also impact the construction of the bounds, especially when coupled with the difference in length and duration between the traces. Therefore, the method doesn't lend itself very well to the task at hand.

A modification of the algorithm needs to be undertaken. Two indexing methods of non-metric similarity measures have been proposed, both dealing with sequences of minimal bounding rectangles. This suggests that the R-tree could be generalized to use a sequence of bounding rectangles in order to efficiently index trajectories.

For comparison, an M-tree using Edit distance with real penalty will be used. Because there is no trivial and useful gap variable for spatio-temporal traces, a decomposition of the trace to compass directions, durations and distances will be used as the basis for the M-tree.

4.3 Incentive management

The incentives that can be purchased with green points are virtual coupons. A major risk of such a system is coupon forging, which will be mitigated by cryptographically

signing each issued coupon.

The delivery of coupons to the user's device is a second concern. The most universal way to approach this would be displaying the coupon's data in an optically-readable format on the device's screen, i.e. as a QR-code.

5. IMPLEMENTATION

5.1 Location obfuscation

Because travel mode recognition requires accurate data, perturbation is the only viable method. For this reason, the method of Ruppel et al.[37] will be used for location obfuscation.

The method is based on a two-step obfuscation. First, rotate all coordinates a pre-set amount around a pre-set centerpoint, followed by applying a pre-set offset. Second, use a random oracle to generate a second offset to the latitude and longitude based on the timestamp of the location.

A random oracle rather than a random generator in the second step in order to both introduce an appreciable amount of fuzz and be able to reverse the step. A salted cryptographic hash will be used as this random oracle, with the intent that the salt will make it harder for an attacker to recreate the oracle.

5.2 Classifiers

Three different classifiers were tested for travel mode detection. These were a Bayes classifier, a decision tree based on Bayes classifiers, and a neural network. Testing proved that the neural network slightly outperformed a single-layer Bayes classifier.

5.2.1 Bayes classifier

The Bayes classifier is based on Bayes' theorem. The basic idea of the classifier is to find the likeliest class given the features. In other words,

$$Y(\bar{x}) = \arg\max_{Y} P(Y|\bar{x}) = \arg\max_{Y} \frac{P(\bar{x}|Y)P(Y)}{P(\bar{x})} \tag{5.1}$$

Where Y is the class and $\bar{x}$ is the feature vector.

In this thesis, the probability density function of a multivariate normal distribution is used as a proxy for the probabilities. For instance, $P(\bar{x}|Y)$ is calculated as

$$f_Y(\bar{x}) = \frac{e^{-\frac{1}{2}(\bar{x}-\bar{\mu}_Y)^T \Sigma_Y (\bar{x}-\bar{\mu}_Y)}}{\sqrt{(2\pi)^d |\Sigma_Y|}} \tag{5.2}$$

Where μ_Y is the mean vector, Σ_Y is the covariance matrix, and d is the dimension of the feature vector.

5.2.2 Continuous classification

The classifiers described above can be used to acquire the momentary likelihoods for the current trace via a moving window method. Relying on momentary outputs may cause aberrations in the output in case of momentary confusion by the classifier.

The classification will be bolstered against such confusions by using Bayesian inference to account for the confusion matrix of the classifier and the likelihoods of transferring from one mode of transport to another.

Let A^* denote the event of the classifier classifying the mode of transport as A, B_{pr} denote the event that the previous mode of transport was B, C_{AB} denote a classification of A as B, and T_{BA} denote the transfer from B to A. Let $\mathbb{M}$ denote the set of all modes of transport.

The probability of A given A^* is

$$P(A|A^*) = \sum_{B \in \mathbb{M}} P(A|A^* \cap B_{pr}) P(B_{pr}) \tag{5.3}$$

$$P(A|A^* \cap B_{pr}) = \frac{P(A^* \cap B_{pr} \cap A)}{P(A^* \cap B_{pr})} \tag{5.4}$$

$$P(A^* \cap B_{pr} \cap A) = P(A^* \cap A|B_{pr}) P(B_{pr}) \tag{5.5}$$

The event $A^* \cap B_{pr}$ above means "previous mode of transport was B and the classifier classifies the current mode as A".

In an ideal case, changing modes of transport and the classifier correctly classifying the current mode of transport would be unrelated. If we consider a correction of a past mis-classification as a transfer, we can write

$$P(A^* \cap A|B_{pr}) = (P(T_{BA}) + P(C_{AB}))P(C_{AA}) \tag{5.6}$$

In which case equation 5.4 becomes

$$P(A|A^* \cap B_{pr}) = \frac{(P(T_{BA}) + P(C_{AB}))P(C_{AA})P(B_{pr})}{P(A^* \cap B)} \tag{5.7}$$

Further,

$$P(A^* \cap B_{pr}) = P(B_{pr})P(A^*|B_{pr}) = P(B_{pr}) \sum_{C \in \mathbb{M}} P(T_{BC})P(C_{CA}) \tag{5.8}$$

Therefore, equation 5.3 becomes

$$\begin{aligned}
P(A|A^*) &= \sum_{B \in \mathbb{M}} \frac{(P(T_{BA}) + P(C_{AB}))P(C_{AA})P(B_{pr})}{P(B_{pr})P(A^*|B_{pr})} P(B_{pr}) && (5.9)\\
\frac{P(A^*|A)P(A)}{P(A^*)} &= \sum_{B \in \mathbb{M}} \frac{(P(T_{BA}) + P(C_{AB}))P(C_{AA})}{P(A^*|B_{pr})} P(B_{pr}) && (5.10)\\
P(C_{AA})P(A) &= P(A^*) \sum_{B \in \mathbb{M}} \frac{(P(T_{BA}) + P(C_{AB}))P(C_{AA})}{P(P(A^*|B_{pr})} P(B_{pr}) && (5.11)\\
P(A) &= P(A^*) \sum_{B \in \mathbb{M}} \frac{(P(T_{BA}) + P(C_{AB}))}{P(A^*|B_{pr})} P(B_{pr}) && (5.12)
\end{aligned}$$

Notice that $\frac{(P(T_{BA})+P(C_{AB}))}{P(A^*|B_{pr})}$ can be tabulated beforehand for more efficient computation.

Equation 5.12 lets the likelihood of each mode of transport be updated based on the previous modes' likelihoods, using the likelihood from the classifier's momentary output as $P(A^*)$ and the previously calculated value for $P(B_{pr})$. The confusion matrix can be calculated from the training dataset, and the transfer likelihoods can be calculated from travel survey data.

Transfer likelihoods calculated from the GeoLife data are in 5.1. The likelihoods were calculated by counting the amount of transfers from one mode to another, and the amount of minutes a each mode lasted.

	To			
From	Walking	Bike	Bus	Car
Walking	99%	0,056%	0,57%	0,24%
Bike	1,5%	98%	0,18%	0,034%
Bus	5,2%	0,055%	95%	0,020%
Car	2,7%	0,0036%	0,0072%	97%

Table 5.1 *The transfer likelihoods calculated from the GeoLife data*

5.3 Feature selection

5.3.1 Features considered

For each segment, the following statistics were calculated for speed and magnitude of acceleration:

- Minimum and maximum
- Median
- Average
- Variance
- Skewness
- Kurtosis

The minimum and maximum speed were ignored, because they had previously been used to filter out the data. Magnitude of acceleration was used because the sum of accelerations would be zero for a segment between two stops.

In the frequency domain, a spectrum was calculated up to the Nyquist frequency, $8mHz$, in four one-octave bins. Cross- and autocorrelations for speed and acceleration were calculated for up to three samples to the past and future.

The segments were classified by mode, and the means and standard deviations were calculated for the features inside each mode. Segments that had one feature more than three standard deviations from the mean were discarded as outliers for each mode.

After initial experimentation, it was found that the correlation, skewness and kurtosis features were of little worth. Therefore, a second experiment was conducted excluding these features from consideration.

Bus route dissimilarity, defined as the smallest DTW with a 3NN output of the MBR tree, was considered in the second experiment. For this reason, the traces from OpenStreetMap were filtered if they were outside a rectangle of specified latitudes. The parameters for this were latitude between 61.25 and 62.25, and longitude between 23.2 and25.2. These values were the approximate bounds of Tampere bus routes.

5.3.2 Dimensionality reduction

[41] used auto-encoders to reduce the dimensionality of a feature vector. In this thesis, however, the aim is to reduce the amount of features that need to be calculated.

Two methods of feature selection were based on two-sample statistical tests. For these, the data was first split to three pairs of classes. Because walking has been found to be easy to classify, the first sub-classification was Walking/Wheeled. Since bicycles were the only muscle-powered mode left Bicycle/Motorized was second, leaving the Car/Bus distinction last. Two statistical tests were then run for all features on each pair of classes.

The first test was Welch's t-test. Because of the large degrees of freedom produced, a comparison of p-values proved impractical. Therefore, the features were ranked on based on their t-values, a larger t-value being more significant.

Because some of the features, such as variances, were not normally distributed, Mann-Whitney U test for large samples was used as the second test.

In the Mann-Whitney U test, each feature was ranked by value, and the output was the smaller of $R - \frac{n(n+1)}{2}$, n being the class size and R being the sum of ranks. A lower output was considered more significant.

A third test, the F-test, could be used to test multiple groups of features at once. The F-test ranks features by the ratio of between-groups variability to within-group variability. A larger F-statistic was therefore more significant.

A number of features for each pair of classes were selected with the procedure in algorithm 5.1.

Since experimentation proved that auto- and cross-correlations were weak distinguishing features (as shown in figure 7.2), the amount of good features returned by each statistical test were estimated by the number of features more significant than the first correlation feature. These amounts are in table 5.2. For the t- and U-tests,

1. Select the next most significant feature not already considered. Call this feature f_0.
2. Select all other features for which correlation with f_0, $|C_{corr}| > C_t$.
3. Choose f_0 as an accepted feature.
4. Mark the other selected features as already considered.
5. Iterate until all features are considered or desired number of features are selected.

Algorithm 5.1 The algorithm for selecting features

C_t	t-test	U-test	F-test
0.1	1	1	1
0.2	1	1	1
0.3	1	1	1
0.4	2	1	2
0.5	2	2	2
0.6	3	2	3
0.7	3	2	4
0.8	4	4	5
0.9	6	6	8
1.0	6	6	17

Table 5.2 *Number of good features for each correlation threshold*

the smallest value for any split (typically the car/bus split) was used.

Correlation threshold of 0.7 was selected, which corresponded to three features per split. The selections from the U- and t-tests were combined. Because this increased the amount of features used, for the F-test, seven indices were selected by that with the correlation threshold 0.9.

5.4 MBR-tree

The method used by Vlachos et al.[40], the storing of a fixed amount of minimum bounding rectangles' coordinates in an R-tree, proved impractical for the purposes of indexing bus routes to measure bus route similarity. If a query trace started in the middle of a scheduled trace, the sequence of bounding boxes translated to a high-dimensional vector a considerable distance away from the scheduled trace's sequence. However, the fact that the method is based on sequences of bounding rectangles suggests a modification to the R-tree.

The MBR-tree is a generalization of the R-tree to use multiple bounding rectangles. For this thesis, a simple generalization of the R*-tree was created by summing costs over a collection of bounding rectangles.

5.4.1 Insertion

Upon insertion, the trace's minimum bounding rectangles are found, and this collection of bounding rectangles is then used as basis for all calculations of insert cost. The insertion cost is defined as the expansion of total bounding rectangle volume required to fit the inserted bounding rectangles, unless the nodes being considered for insertion are leaf nodes. In the case of leaf nodes, the additional overlap with sister nodes is used as the insertion cost, instead.

If the insertion cost ends up being more than a fraction p of the total volume of the node's sequence of bounds, an entirely new node is created with bounds equal to the inserted child node.

If the insertion brings the node's child count over a set number, the node is split.

5.4.2 Split

There are two kinds of split used by the MBR tree: forced reinsertion and regular split. In the first instance of split being invoked for a given level during a "top-level" insertion, a forced reinsertion takes place.

Forced reinsert

In a forced reinsert, the node's children are sorted according to their bounds' average center point's distance from the node's average center point. Then, a fraction q of the outermost nodes are removed from the tree, in ascending order of distance, and inserted back into the tree. The purpose of this procedure is to optimize the tree and offset any non-optimalities introduced by the order of insertion.

If a node split occurs during forced reinsertion, if it is not on a level already undergoing forced reinsertion, another forced reinsertion takes place. Otherwise a regular split is carried out to prevent infinite recursions.

Regular split

In a regular split, the dimension in which the total of all children's bounding boxes' widths is smallest is found, and the children are sorted according to their bounding boxes' extreme in that dimension. The children are then split into two at the index where the inter-group overlap is smallest. Two nodes are then filled with the two groups of children, and are inserted under the split node's parent. The split node is then removed from the parent.

If the split node is the root node of the tree, a new root node is created and the two new nodes inserted under it.

5.4.3 Querying

Two sorts of queries are used in this thesis. For purposes of bus route similarity measurement, a kNN query was carried out. The similarity measurement used is LCSS, with ties broken by DTW. A contains-query was used during the construction of the index to avoid inserting duplicate traces.

For the kNN query, a lower bound for the LCSS is established by counting the amount of points in the query trace that are contained by each node's bounding boxes. Nodes containing at least two thirds of the query's points are considered for return. The traces contained in the leaf nodes are also filtered according to their bounding boxes, and the remainder are sorted according to their LCSS and DTW with the query. During experimentation, filtering by the actual LCSS proved too strict to return any results.

The contains-query was implemented in a similar fashion, except that the bounds were required to contain all the query's points, and instead of sorting, the candidate traces were tested for equality.

5.4.4 Finding minimum bounding rectangles

Two methods were considered for generating a sequence of bounding rectangles: a method to find the genuine minimum bounding rectangles, and splitting the trace into five-minute segments.

The method for genuine bounding rectangles was lined out in [40]. The trace is first split into two-sample overlapping boxes, and then the two boxes that can be

combined with the least increase in total volume are combined, repeating until the amount of boxes is the desired amount. For this thesis, the desired amount was set at the square root of the trace's length in samples.

The bounding rectangles were in four dimensions: latitude, longitude, time of day and day of week. Since Tampere bus schedules apply to either single days or continuous stretches of days, it was trivial to generate the "day of week" dimension of the bounding rectangle. If the bus routes had run on non-continuous days, it would have been necessary to create duplicates of the route for each block of days.

5.5 Compass directional decomposition

The decomposition discussed here makes use of a method of approximating deviation from a straight line. The approximation is calculated by dividing the area inside the polygon defined by track points by the direct distance between the start and end of the track. This was done to further reduce the effect of road curvature on the perceived direction of movement. A threshold of 50 meters was used

Li et al. introduced the idea of comparing trajectories by representing each trajectory as a series of compass directions[25]. In this paper, this method is expanded by including the length and duration of the path segment in question to the comparison.

The scheduled trace is decomposed into a list of segments consisting of a principal wind, distance and duration. The distance metric $f(r, s)$ is defined as a weighted sum of the difference in durations, difference in distances and the amount of compass-point intervals between the directions. A gap segment is defined as having zero distance and duration, and zero difference in direction to any other.

The decomposition is carried out by algorithm 5.2.

5.5.1 Parameters of flexible measures

The following weights were used for the flexible distance measurements are presented in table 5.3. If multiple measures used a parameter with the same name, the same value was also used.

5.6 Incentive management

One challenge in incentive management is the delivery of an electronic coupon with a pair of cryptographically large numbers representing a point in a cryptographic

1: *segmentStart*:=first point of trace
2: *previousPoint*:=*segmentStart*
3: *currentDir*:=none
4: **for all** *trackpoint* in trace **do**
5: calculate *deviation* between *trackpoint* and *segmentStart*
6: **if** direction from *previousPoint* to *trackpoint* $\neq$ *currentDir* **and** *deviation* $\geq$ *threshold* **then**
7: Add segment with direction *currentDir* and distance and duration between *segmentStart* and *previousPoint* to decomposition
8: *segmentStart*:=*previousPoint*
9: *currentDir*:=direction from *previousPoint* to *trackpoint*
10: **end if**
11: previousPoint:=trackPoint
12: **end for**
13: Add segment with direction, distance and duration between *segmentStart* and *previousPoint* to decomposition

Algorithm 5.2 Algorithm for the compass-directional decomposition.

Parameter	Value	Explanation
Distance weight	1/km	Base weight
Time weight	$0.0139/s$	Corresponds to velocity of 50 km/h
Compass point weight	$\sin(\frac{\pi}{4})$	Distance between two lines 1 compass point apart at the 1 km mark
Distance stretch	50m	Allow for location queries happening when the bus is between stops.
Time stretch	5 min	Allow some deviation from bus schedule.

Table 5.3 *The parameters used for the metrics*

elliptic curve. For example, the signature of Enzmann et al. was calculated for a mockup payload, and was encoded to the QR code in figure 5.1.

It appears that a single elliptic curve point is close to the maximum for a QR code that can be shown on a smartphone screen.

While there are Java libraries for pairing-based cryptograpy, such as JPBC[15], for this thesis tools for elliptic-curve arithmetic were implemented as a learning exercise. Elaborating these tools to a coupon system will be left to future work.

Figure 5.1 *A QR code encoding a mockup payload and an Elliptic-curve based signature*

The decision whether or not to award green points will be based on a slightly modified DTW between the stated and observed trace, segmented by mode. The distance between and observed and stated segment will be the a weighted sum of the DTW between the corresponding GPS traces and the complement of the likelihood given to the stated segment's mode by classifying the observed trace.

6. RESULTS

6.1 Selected features

As mentioned in 5.3, two experiments were run on the same data. The main difference between the two experiments was that the confounding effect of buses was eliminated by introducing a bus route similarity metric.

Because non-bus traces did not match a single bus route, they were assigned a random DTW evenly distributed in the range $[58, 65]$ to prevent divisions-by-zero in the statistical tests. The range was selected because the maximum DTW of the bus traces was approximately 61.

6.1.1 Initial experiment

Ranking by the t-test produced the features, eight in total, in table 6.1. Ranking by the U-test produced the features, six in total, listed in table 6.2. Ranking by the F-test produced the features listed in table 6.3.

6.1.2 Second experiment

The second feature selection yielded results similar to the first, except for the absence of the eliminated features, and the prominent presence of bus route dissimilarity. The correlations between all features are in appendix 8.1. The F-statistics for each feature are in appendix 8.1

All the feature selections were affected by to the eliminated variables, and the prominent presence of the bus route dissimilarity. The outputs are in tables 6.4 and 6.5, respectively. The F-test results are in table 6.6.

#	Walk/Wheeled	Bicycle/Motorized	Car/Bus
1	Speed. spectrum $1-2mHz$ (119)	Speed spectrum $4-8mHz$ (88.4)	Average speed (50.9)
2	Accel. spectrum $2-4mHz$ (85.7)	Speed spectrum $2-4mHz$ (57.3)	Speed skewness (16.0)
3	Minimum acceleration (56.1)	Minimum acceleration (37.2)	Speed variance (14.3)

Table 6.1 *Features selected with Welch's t-test, t-values in parentheses*

#	Walk/wheeled	Bicycle/Motorized	Car/Bus
1	Average acceleration (247)	Average acceleration (156)	Average speed (112)
2	Average speed (272)	Speed spectrum $2-4mHz$ (837)	Speed skewness (515)
3	Minimum acceleration (937)	Minimum acceleration (1323)	Speed variance (550)

Table 6.2 *Features selected with U-test, u-values/1000 in parentheses*

#	Feature
1	Speed. spectrum $1-2mHz$ (8069)
2	Speed. spectrum $4-8mHz$ (7336)
3	Acceleration Spectrum $0-1mHz$ (7075)
4	Average acceleration (6944)
5	Speed spectrum $2-4mHz$
6	Minimum acceleration (1265)
7	Speed skewness (1187)

Table 6.3 *Features selected with F-test, f-statistic in parentheses*

#	Walk/Wheeled	Bicycle/Motorized	Car/Bus
1	Speed. spectrum $4-8mHz$ (24.0)	Speed spectrum $1-2mHz$ (13.8)	Bus route dissimilarity (46.1)
2	Bus route dissimilarity (15.2)	Average acceleration (13.5)	Speed spectrum $2-4mHz$ (11.6)
3	Median speed (13.7)	Speed spectrum $4-8mHz$ (7.34)	Minimum acceleration (4.41)

Table 6.4 *Features selected with Welch's t-test (2nd experiment), t-values in parentheses*

#	Walk/wheeled	Bicycle/Motorized	Car/Bus
1	Speed spectrum $1-2mHz$ (194)	Median speed (362)	Bus route dissimilarity (0)
2	Acceleration spectrum $0-1mHz$ (309)	Average acceleration(370)	Speed spectrum $2-4mHz$ (153)
3	Median acceleration (384)	Speed spectrum $4-8mHz$ (1236)	Minimum acceleration (829)

Table 6.5 *Features selected with U-test (2nd experiment), u-values in parentheses*

#	Feature
1	Speed spectrum $1-2mHz$ (382)
2	Acceleration spectrum $1-2mHz$ (207)
3	Speed spectrum $4-8mHz$ (181)
4	Speed spectrum $0-1mHz$ (159)
5	Bus route dissimilarity (145)
6	Average acceleration (118)
7	Acceleration variance (68.1)

Table 6.6 *Features selected with F-test (2nd experiment), f-statistic in parentheses*

6.2 Travel mode detection

6.2.1 Initial experiment

As explained in section 3.2, 70 traces from each class were set aside as a validation set. The rest of the data was used to train each classifier with 5-fold cross-validation.

The average F1 scores, weighed with the inverse of class size, over the four classes are in table 6.7. Only one value is shown for the autoencoder because the feature selection step was bypassed for that method.

Ranking criterion	BC	NN1	NN2	RF	AE
Welch's t	0.865	0.856	0.862	0.850	
U-test	0.856	0.849	0.845	0.846	0.651
F-test	0.841	0.859	0.848	0.813	

Table 6.7 *Average F1 scores, for each classifier and feature selection method*

Ranking the results by t-test produced the classifier with the highest F1 score for each case except single-layer neural network. An additional layer in the neural network provided a small benefit for one test and a marginally worse result for the other two.

The confusion matrices of the best-performing instance of each classifier will be presented next. Table 6.8 presents the confusion matrix and recall rates for a Bayes classifier.

Correct label	Classified as			
	Walk	Bicycle	Bus	Car
Walk	97.7%	2.31%	0.0%	0.0%
Bicycle	1.06%	92.4%	6.21%	0.367%
Bus	0.0%	17.3%	77.3%	5.32%
Car	0.0%	1.37%	23.1%	75.5%
Precision	98.8%	87.6%	73.7%	90.5%

Table 6.8 *Confusion matrix of the Bayes classifier*

The way buses confound this classifier is obvious from the 73.7% precision and 77.3% accuracy. A Similar confounding can be seen in tables 6.9, 6.10, which present the confusion matrices of the one- and two layer neural networks. As can be seen, the difference between one and two layers is a few percent points one way or the other in accuracy for each class, although there was a 5.6% increase in precision for bicycles and 6.9% for cars.

Correct label	Classified as			
	Walk	Bicycle	Bus	Car
Walk	100%	0.0%	0.0%	0.0%
Bicycle	2.90%	89.9%	5.80%	1.45%
Bus	0.0%	14.3%	70.0%	15.7%
Car	0.0%	0.0%	15.7%	84.3%
Precision	97.2%	86.1%	76.6%	83.1%

Table 6.9 *Confusion matrix of the neural network (1 hidden layer)*

Correct label	Classified as			
	Walk	Bicycle	Bus	Car
Walk	97.1%	2.86%	0.0%	0.0%
Bicycle	2.9%	87.0%	10.1%	0.00%
Bus	0.0%	14.3%	71.4%	14.3%
Car	0.0%	1.43%	15.7%	82.9%
Precision	97.7%	91.7%	78.8%	90.0%

Table 6.10 *Confusion matrix of the neural network (2 hidden layers)*

Table 6.11 presents the confusion matrix for the autoencoder, which performed all-around abysmally with an F1-score of 0.651. This is a considerably poorer performance than in Zhu et al.'s 93.58% precision and 93.32% recall[41].

Correct label	Classified as			
	Walk	Bicycle	Bus	Car
Walk	88.6%	11.4%	0.0%	0.0%
Bicycle	31.9%	62.3%	5.8%	0.0%
Bus	4.29%	17.1%	51.4%	27.1%
Car	0.0%	0.0%	21.4%	78.6%
Precision	71.3%	68.3%	65.5%	74.3%

Table 6.11 *Confusion matrix of the autoencoder neural network*

Table 6.12 presents the confusion matrix for the random forest. The classifier had particular problems within the motorized and non-motorized vehicle classes, as it classified 31.9% of bicycles as walking and over 20% of buses as cars and vice versa.

The percentages in the confusion matrices are the ratio of the known class classified as each class, meaning the diagonal values constitute the recall rates. The precision, or percentage of traces classified into a mode belonging to that mode, is given below each confusion matrix.

Correct label	Classified as			
	Walk	Bicycle	Bus	Car
Walk	98.6%	1.43%	0.0%	0.0%
Bicycle	1.45%	89.9%	8.7%	0.0%
Bus	0.0%	12.9%	75.7%	11.4%
Car	0.0%	1.43%	22.9%	75.7%
Precision	98.2%	94.7%	75.1%	91.7%

Table 6.12 *Confusion matrix of the random forest*

6.2.2 Second experiment

As explained in section 3.2, the entire data set was used as the validation set.

Isolating buses from the other modes of transport by the bus route similarity metric produced higher accuracy and precision in all four classes, except for the still-addled autoencoder. The F1-scores are in table 6.13. In fact, the classification results are fairly similar to Zhu et al.[41].

Ranking criterion	BC	NN1	NN2	RF	AE
Welch's t	0.978	0.974	0.980	0.953	
U-test	0.969	0.988	0.978	0.926	0.599
F-test	0.975	0.968	0.980	0.859	

Table 6.13 *Average F1 scores (2nd experiment), for each classifier and feature selection method*

The confusion matrices are presented next, as with the first experiment.

Correct label	Classified as			
	Walk	Bicycle	Bus	Car
Walk	98.4%	1.59%	0.0%	0.0%
Bicycle	0.0%	100%	0.0%	0.0%
Bus	0.0%	8.57%	91.4%	0.0%
Car	0.0%	0.0%	0.0%	100%
Precision	100%	90.1%	100%	100%

Table 6.14 *Confusion matrix of the Bayes classifier (2nd experiment)*

As can be seen from table 6.14, trimming the useless features and incorporating bus route similarity reduced most misclassifications to or from the bus class to near zero.

Tables 6.15 and 6.16 show the one- and two-layer neural networks' confusion matrices, with the two-layer network performing slightly worse. The difference is, however, only a few percent points. Indeed, the difference in F1 scores is only 0.008.

Correct label	Classified as			
	Walk	Bicycle	Bus	Car
Walk	100%	0.0%	0.0%	0.0%
Bicycle	0.0%	96.9%	3.13%	0.00%
Bus	0.0%	2.86%	97.1%	0.0%
Car	0.0%	0.00%	0.0%	100%
Precision	100%	96.9%	97.1%	100%

Table 6.15 *Confusion matrix of the neural network (1 hidden layer, 2nd experiment)*

Correct label	Classified as			
	Walk	Bicycle	Bus	Car
Walk	96.8%	3.17%	0.0%	0.0%
Bicycle	1.56%	93.8%	4.69%	0.0%
Bus	0.0%	0.0%	100%	0.0%
Car	0.0%	0.0%	0.0%	100%
Precision	98.4%	96.8%	95.9%	100%

Table 6.16 *Confusion matrix of the neural network (2 hidden layers, 2nd experiment)*

The bus/bicycle class division was the only cause of confusion for the one-layer network, with 3.13% being misclassified as bicycles, and 2.28% of bicycles as buses. The confusion shifted to the bicycle class in the two-layer network, with all three cases of misclassification being to or from the bicycle class.

Correct label	Classified as			
	Walk	Bicycle	Bus	Car
Walk	95.2%	4.76%	0.0%	0.0%
Bicycle	37.5%	43.8%	10.9%	7.81%
Bus	0.0%	22.9%	25.7%	51.4%
Car	0.0%	0.0%	0.0%	100%
Precision	71.4%	59.6.5%	72.6	47.4%

Table 6.17 *Confusion matrix of the autoencoder neural network (2nd experiment)*

As can be seen from table 6.17, the autoencoder was even more addled than in the first experiment.

The random forest's confusion matrix is in table 6.18. This classifier, too, had hardly any confusion. It would appear that bicycles and buses confounded the classifier at approximately equal degrees. The 88.6% accuracy in particular stands out.

Correct label	Classified as			
	Walk	Bicycle	Bus	Car
Walk	100%	0.0%	0.0%	0.0%
Bicycle	4.69%	92.2%	3.13%	0.0%
Bus	0.0%	11.4%	88.6%	0.0%
Car	0.0%	0.0%	2.70%	97.3%
Precision	95.5%	88.1%	95.4%	100%

Table 6.18 *Confusion matrix of the random forest (2nd experiment)*

6.3 MBR-tree

The performances of the indexing methods are in table 6.19. Standard deviations are also shown. The indexes did not return a single trace for queries with certain traces, the number of these is shown in the "missed traces" column. A one-sided Welch's t-test was performed to gauge the significance of the differences, the results are in table 6.20.

Index	time (ms)	DTW	LCSS/length	AED	missed traces
MBR-tree (actual MBR)	7.99 ± 17.4	21.4 ± 17.4	0.20 ± 0.14	1.79 ± 0.13	9
MBR-tree (time split)	10.7 ± 33.2	28.9 ± 68.5	0.19 ± 0.14	9.27 ± 64.8	9
M-tree	20.3 ± 80.2	350 ± 149	0.0 ± 0.002	345 ± 148	0
Linear search	29.3 ± 23.0	21.4 ± 17.4	0.20 ± 0.14	1.79 ± 0.13	9

Table 6.19 *Performance of the indices*

	P-value			
Metric	MBRt-MBRm	MBRm-M	MBRm-L	M-L
Query time	0.005	0.0	0.0	0.0
DTW	$\ll 0.001$	0.0	0.5	0.0
LCSS	0.05	0.0	0.5	0.0
AED	$\ll 0.001$	0.0	0.5	0.0

Table 6.20 *P-values between M-tree (M), MBR-tree with time split (MBRt) and minimum bounding(MBRm), and Linear search (L)*

Some of the missed traces may have been caused by buses still reporting their location while returning to the garage for refueling etc.. In any case, this amounted to 0.08% misses, an acceptable rate for most cases.

Examples of the 5NN outputs of the indexes are in figure 6.1 for the M-tree, and figures 6.2 and 6.3 for the MBR-tree. The query trace is marked in red, and the

other five traces are the 5NN query outputs in order of proximity. As could be expected, linear search and the corresponding MBR-tree performed identically in terms of accuracy, and its outputs are excluded.

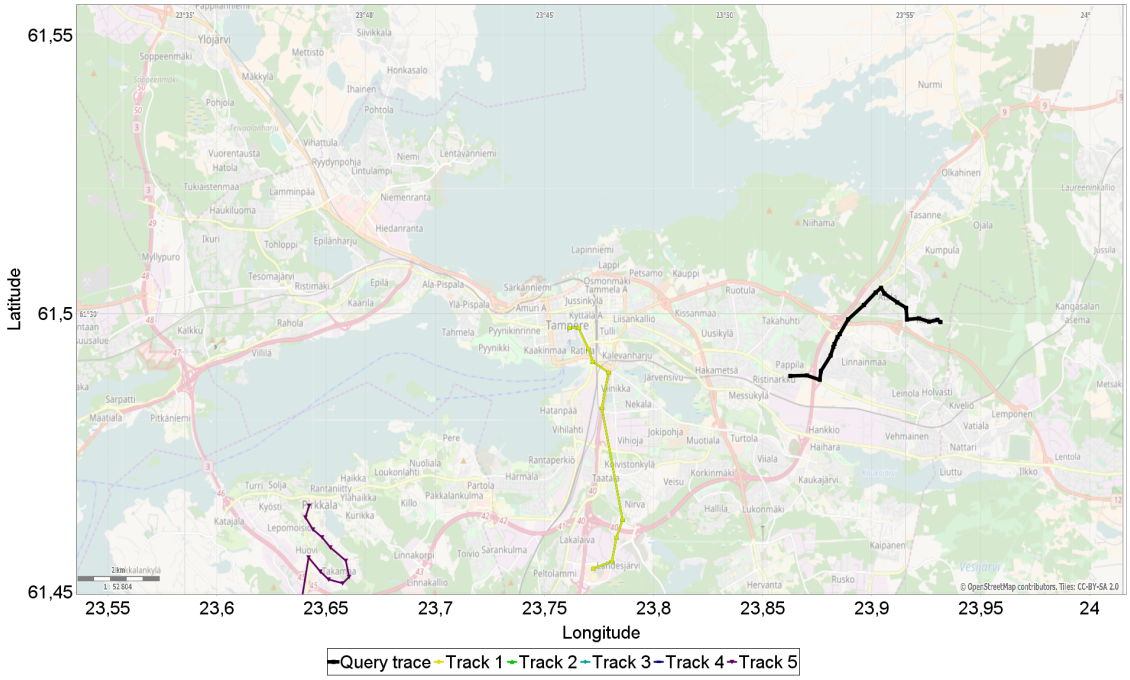

Figure 6.1 *5NN output from the M-tree*

All the traces returned by a query to the M-tree seem to be of approximately the same length as the query trace. This is shown in 6.1

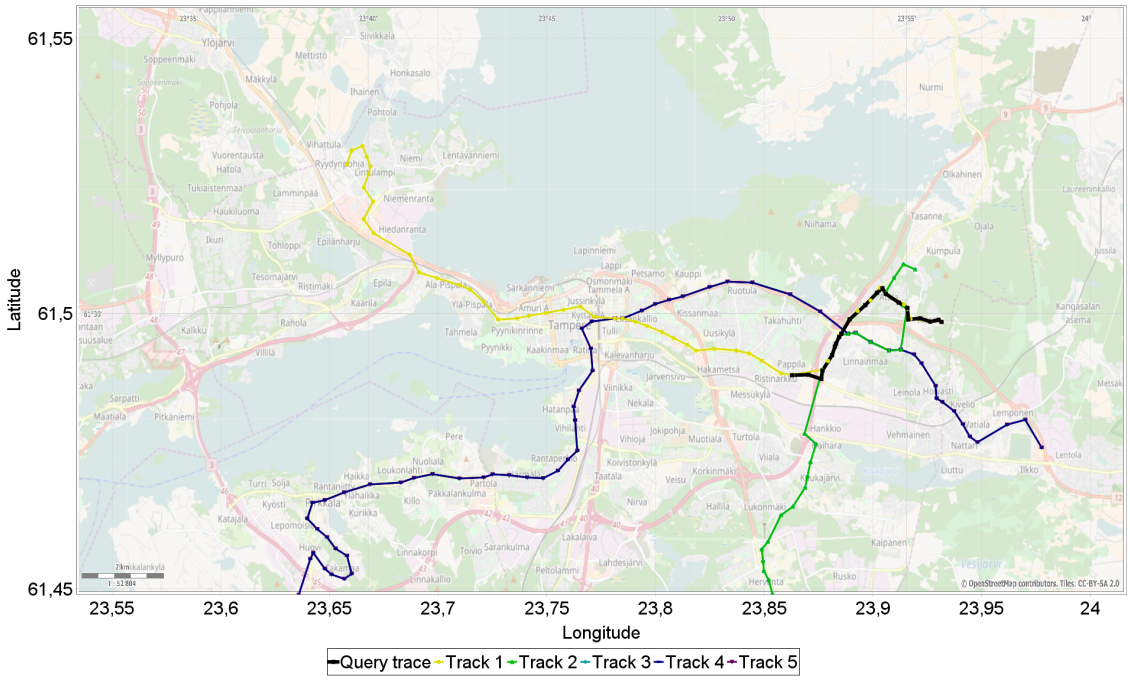

Figure 6.2 *5NN output from the MBR-tree (actual mbr)*

Using genuine MBRs, the tree returned one exactly matching bus route, one near miss and two that appear to only intersect the query, as can be seen from figure 6.2.

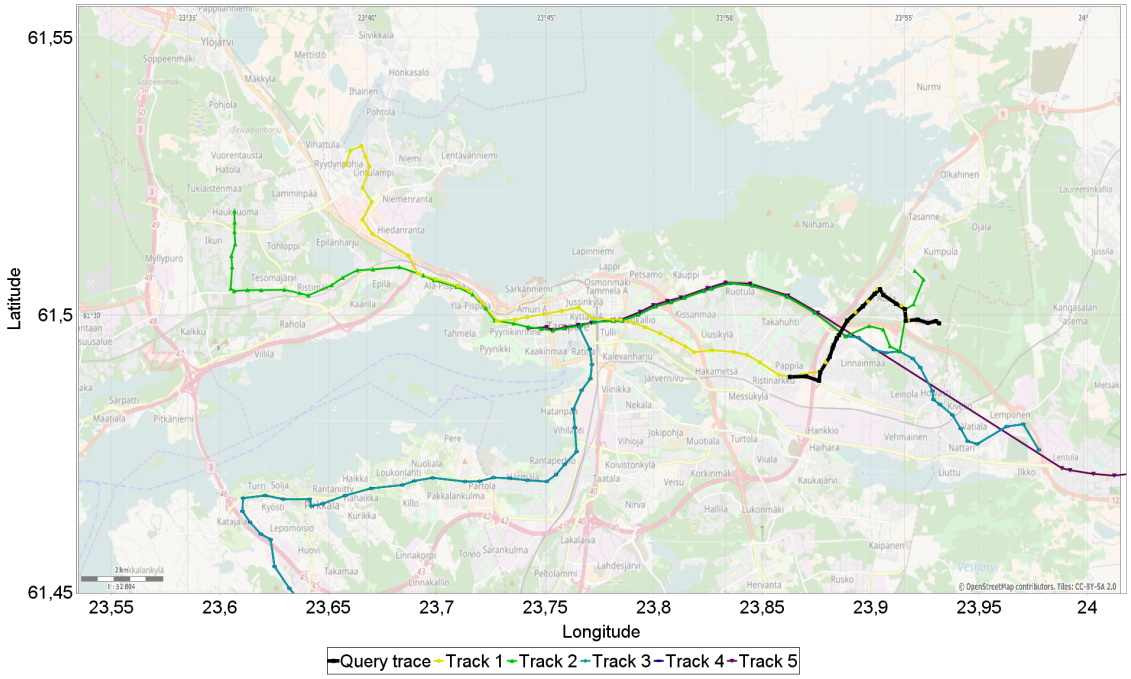

Figure 6.3 *5NN output from the MBR-tree (time split)*

Figure 6.3 shows that the first trace returned is the same, but the other traces seem to only intersect the query trace.

A visualization of the spatial dimension bounding rectangles of every trace inserted is in figure 6.4. For comparison, the MBR tree's first three levels' bounds are in figure 6.5 for the genuine minimum bounding rectangles and figure 6.6.

From figure 6.4, it can be seen that the Tampere bus routes cover an approximately triangular area some two hundred kilometers east-to-west, and approximately a hundred kilometers north-to-south. The closer one goes to the center of Tampere, the more densely packed the bus routes are.

The two large bounding boxes in figure 6.5 split the region into, roughly, northern and southern sections, with an overlap around Tampere proper. What the figure doesn't show is that the large northern box is duplicated over the first three levels, which results in the apparent emptiness of the bounding box.

As can be seen from figure 6.6, there are a few very tight envelopes in each level of the time-split tree, combined with several larger bounding boxes. A north-south divide similar to the genuine-MBR tree is evident on the third level. Once again,

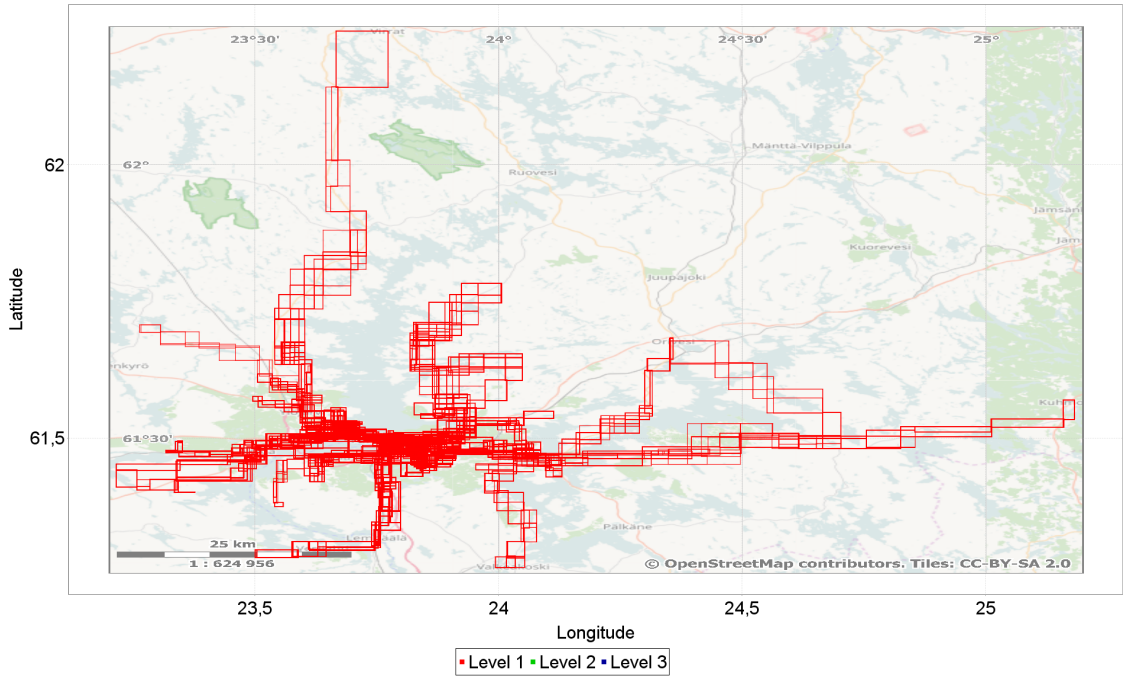

Figure 6.4 *The bounding rectangles of all scheduled traces*

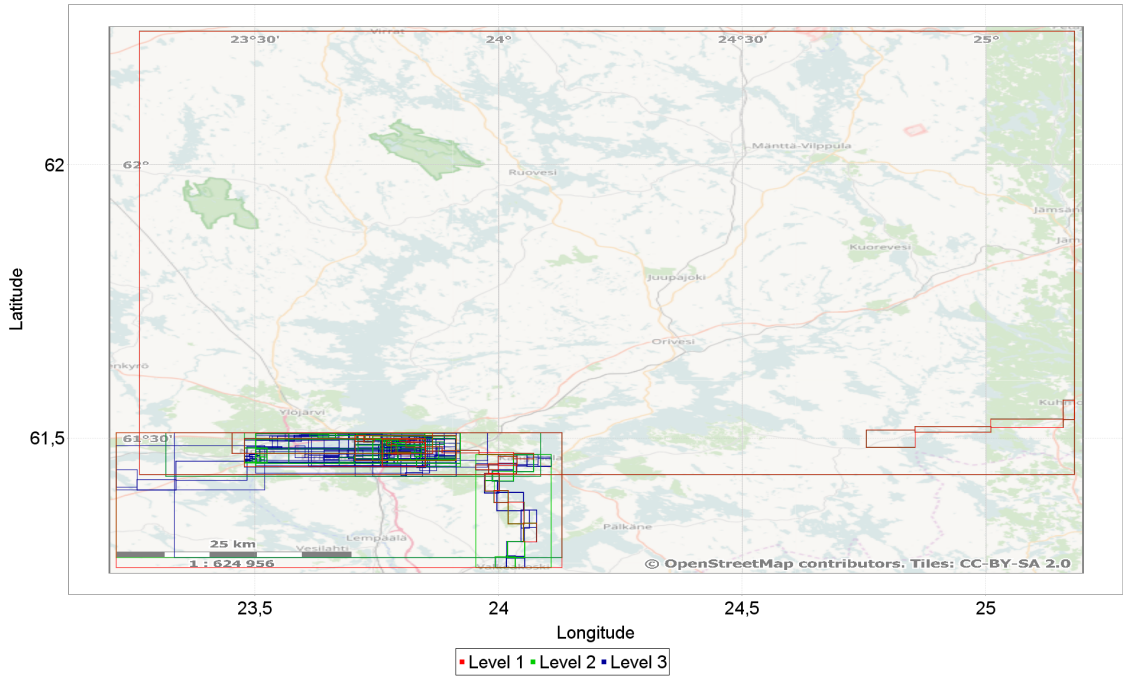

Figure 6.5 *The first three levels of the MBR-tree (genuine MBR)*

the largest bounding box has been duplicated on level 2.

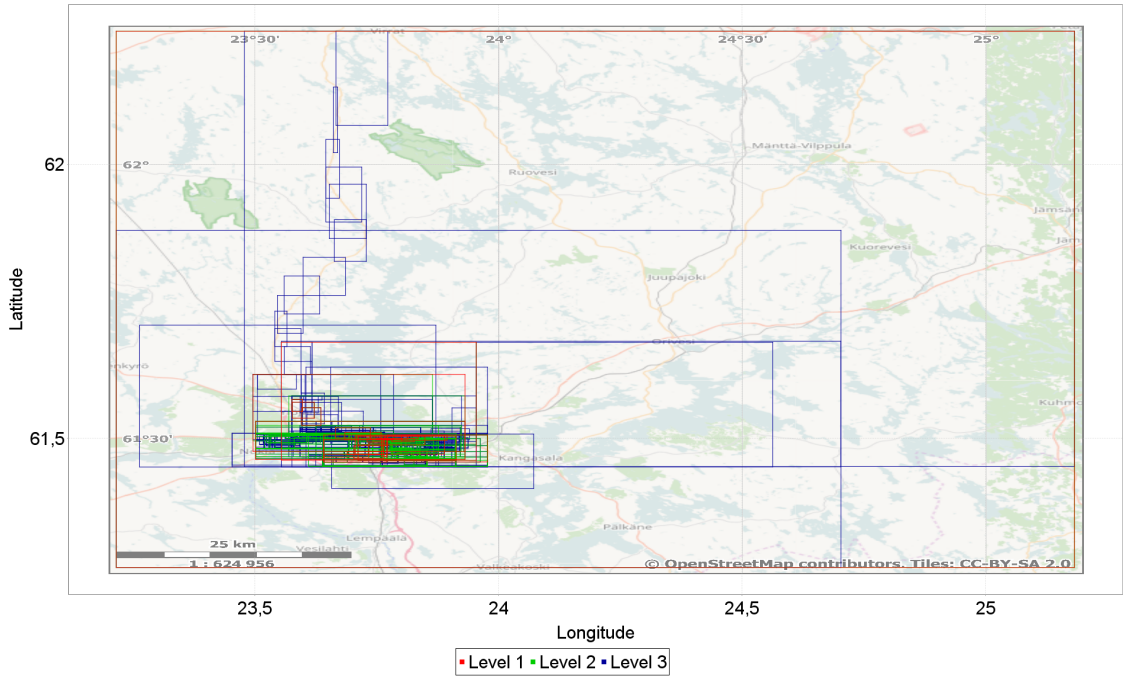

Figure 6.6 *The first three levels of the MBR-tree (time split)*

7. DISCUSSION

7.1 Travel mode recognition

7.1.1 Hypothesis testing

For all three hypotheses outlined in 1.4, the F-test produced p-values lower than floating point precision on at least one feature.

Spectral components for acceleration and speed were selected for each sub-classification. The F-test ranked spectral features at ranks 2-7. Most of the frequency bins selected by U- and t-tests did not contain zero, suggesting that there was more to this significance than the average. Therefore, hypothesis 1 can be accepted.

As a further illustration of the spectral components' suitability for the classification task, a scatterplot of two speed spectral components is shown in figure 7.1.

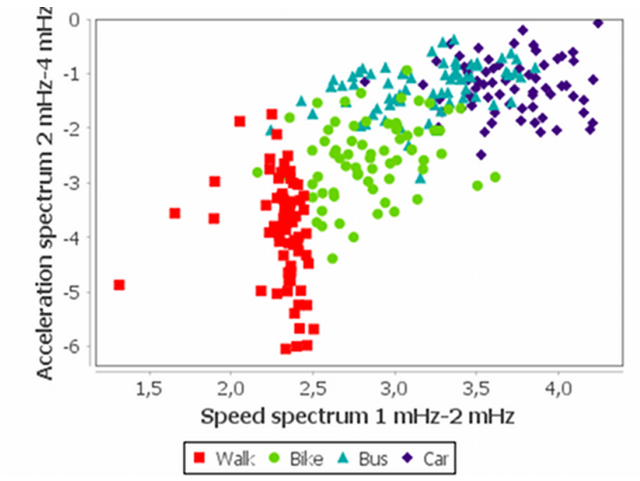

Figure 7.1 *Scatter plot two spectral components of speed and acceleration*

Auto- and cross-correlations of speed and acceleration were selected to differentiate between walking and wheeled modes of transport. Therefore, hypothesis 2 might be accepted.

However, looking at the scatterplot of autocorrelations in figure 7.2 reveals that the classes are heavily overlapped. Therefore the hypothesis can be discarded.

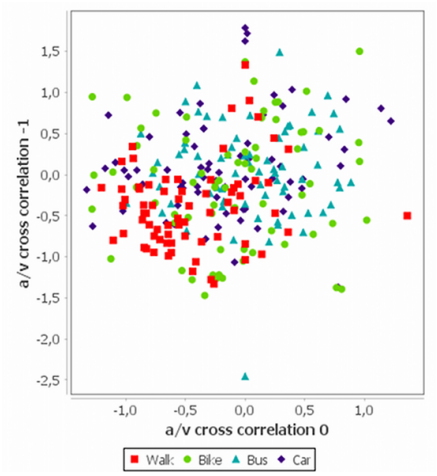

Figure 7.2 *Scatter plot of Speed and acceleration autocorrelations*

Skewness of speed was selected as one of the best features to differentiate cars and buses, and the seventh most discriminating feature. Therefore, hypothesis 3 might be accepted. However, as with the correlations, the scatterplot in figure 7.3 tells a different story. The hypothesis can also be discarded.

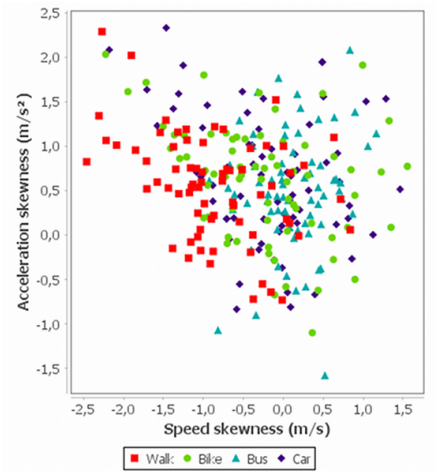

Figure 7.3 *Scatter plot of Speed and acceleration skewnesses*

Bus route similarity ended up having slightly worse than median F-statistic, possibly because only bus- and bike traces really had any similarity to bus routes in the training data. For the same reason, it ended up being the most significant feature for the car/bus split.

A scatter-plot of bus-route dissimilarity is shown in figure 7.4. The band of data-points to the right are the ones assigned a random maximum dissimilarity. Looking at the scatter plot, it would appear that hypothesis 4 would be valid.

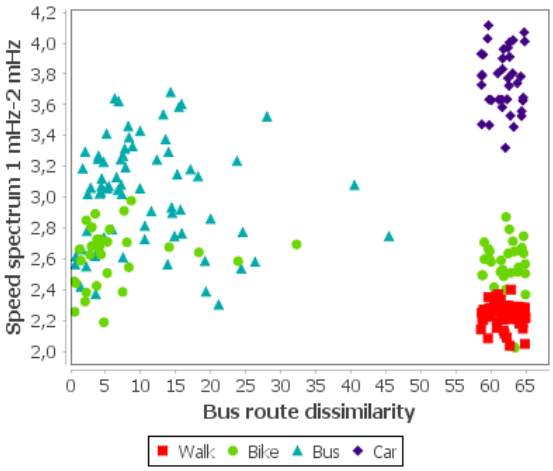

Figure 7.4 *Scatter plot of Speed and acceleration autocorrelations*

7.1.2 Classification

Buses stood out as a confounding class in the initial experiment, as can be seen from figure 7.1. In particular, the class had a precision of under 80% where the rest of the classes typically had at least 83% precision.

In the initial experiment, the Bayes classifier performed best, with a slight margin of 0.003 over the two-layer neural network. In the second experiment, the two-layer neural network performed best with all three ranking criteria.

The autoencoder neural network performed abysmally in both experiments. One possible cause is that the autoencoder was trained with regular backpropagation rather than an autoencoder- or deep neural network-specific algorithm. It is also possible that the autoencoder or classifier were not deep enough to truly leverage the benefits.

Visualizations were made for the highest F1 score classifier of each type, by assigning points at $(0, \pm 1)$ and $(\pm 1, 0)$ to the four classes counter-clockwise from $(1, 0)$, and assigning each datapoint a location as a sum of these points weighted by the likelihood, normalized to between 0 and 1, given by each classifier.

As can be seen from figure 7.5, the Bayes classifier occasionally assigned high likelihoods to two or more classes.

Figure 7.6 shows the visualization for the neural network. The classifier did not, as a rule, output high likelihoods for more than one class, but did on occasion output middling likelihoods for two classes. The intermingling of buses with cars and bicycles is clearly visible.

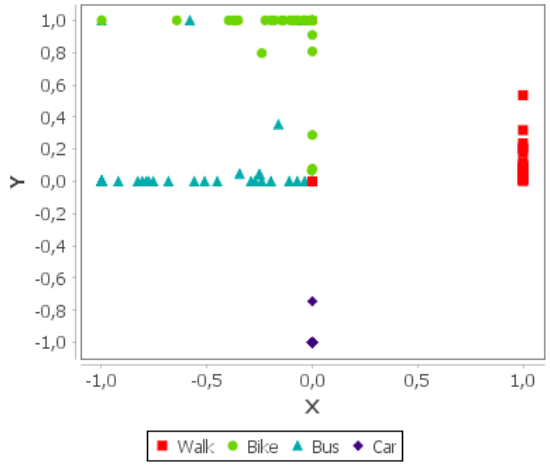

Figure 7.5 *Visualization of the Bayes classifier*

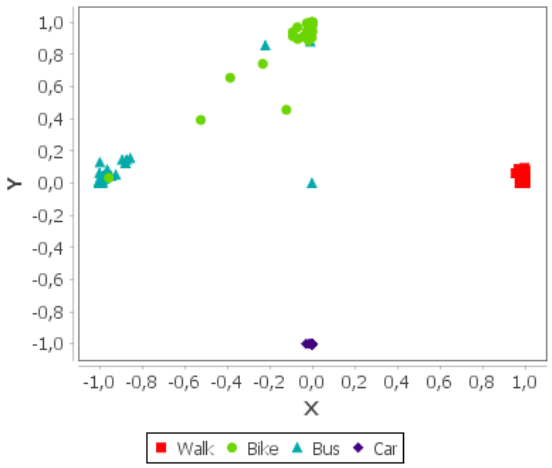

Figure 7.6 *Visualization of the Neural network*

Figure 7.7 shows the visualization of the random forest. The forest's likelihoods were discrete due to the implementation, and appear fairly widely spread.

From all three visualizations, it would appear that the classifiers would have a near-perfect accuracy if it were not for the bus class.

7.2 Results of the MBR-tree

The genuine minimum bounding rectangles method significantly outperformed the time-split method on all metrics. The slight increase in computational complexity ($O(nlogn)$ vs $O(n)$) is most likely worth the increased precision in most applications. In either case, the lower bound for LCSS was not very tight, and consistently produced scheduled traces of less than the minimum LCSS. This may be due to the scheduled traces' granularity being dictated by the time between bus stops, whereas

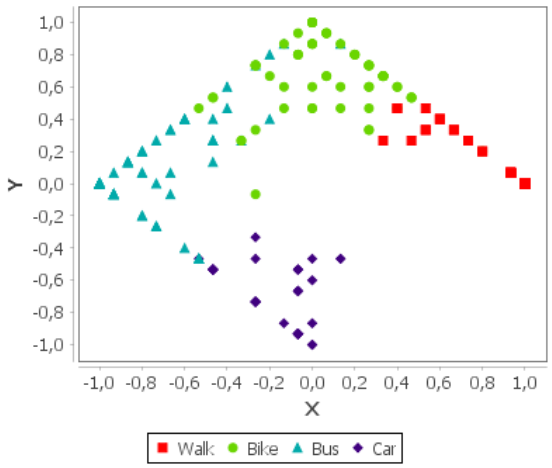

Figure 7.7 *Visualization of the random forest*

the testing data had an approximately 30 second granularity.

As can be seen both from the metrics, and the figures, the modified R-tree produced more spatially similar traces than the M-tree. A most likely cause is that ERP, due to triangle inequality, is offset by the difference in lengths of the trace. This is also the likely cause of the M-tree returning mostly shorter scheduled traces.

The average query time was also significantly faster with the MBR-tree than either the M-tree or a linear search. This may be in part due to differences in implementation. Another contributing factor is that ERP's computational complexity is quadratic in the worst case, whereas the bounding rectangle method used to upper-bound LCSS was linear.

All query times varied greatly. One explaining factor is that a large number of bus routes pass through certain nexuses, such as Keskustori in the center of Tampere. Therefore, a trace passing through one of these nexuses would resemble a large amount of scheduled traces, whereas a trace of an out-of-the-way bus route might only resemble it's own schedule.

7.3 Travel survey

The travel survey produced no car traces, a total of four bus traces, 64 bicycle traces and 12 walking traces, after splitting by time but before filtering outliers. Compared to the 441 walking-, 55 bicycle-, and 40 car traces from the Openstreetmap data, only bicycle traces were numerous enough to make a difference in the data.

8. CONCLUSION

This thesis has presented a means of classifying sparse GPS data into four modes of transport.

For this purpose, algorithm 5.1 was presented for selecting features. Spectral components were found to be among the most significant features to be considered. Auto- and cross correlations, skewnesses and kurtoses of speed and acceleration were found to be of little value for this classification task.

In order to account for bus-route similarity, an index structure for spatio-temporal traces was introduced in section 5.4, and shown to be more efficient than linear search. ERP on sequences of triples of compass-point, duration and distance, constructed by algorithm 5.2, was found to be unsuitable for comparing route fragments. The MBR-tree performed significantly faster, both in the statistical and absolute sense, than a linear search, but the query times had considerable variance.

A proof-of-concept comparison of machine-leaning approaches to travel mode recognition with and without a bus route similarity measure, based on the MBR-tree, was provided in chapter 6, and the bus route similarity was shown to improve the results.

Methods of obfuscating the user's location were discussed, and only reversible perturbation was compatible with the travel mode recognition described above.

Tools for elliptic curve cryptography were implemented, up to providing a mockup of an electronic coupon system. If electronic coupons are to be delivered as QR codes on a phone screen, a single elliptic curve point is fairly close to the limit of what can be delivered as a signature.

8.1 Future work

A wider travel survey is needed to gather proper training data. Possible causes of poor attendance were the lack of incentives for participation, and the poor usability of the travel survey application.

The MBR-tree had an issue where the parent tree's bounds were copied exactly by the child tree. There was also some imbalance to the size of the bounds of nodes in a given level. Both of these should be addressed to develop the MBR-tree as an indexing algorithm.

The possibility of using non-reversible location obfuscation while still inferring the user's mode of transport was not studied. Literature on the subject of travel mode recognition does not appear to consider intentionally obfuscated data. Therefore, combining location obfuscation and travel mode recognition is a possible avenue of future research.

This thesis did not touch on the subject of identifying location spoofing. If the incentives are to be worth money, this consideration must be made. For instance, the system of Pham et al. [34] could be either used as-is or modified for the needs of Tampere Smart Mobility Engine. As mentioned in section 5.6, implementing the incentive management has also been left for future work.

BIBLIOGRAPHY

[1] "Data sources in its factory."

[2] "Geolife: Building social networks using human location history."

[3] "OpenStreetMap project," http://www.openstreetmap.org.

[4] M. Andrews and L. Zhang, "Tracking mobile users via standard routing engines," in *2016 Annual Conference on Information Science and Systems (CISS)*, March 2016, pp. 258–261.

[5] S. Atev, G. Miller, and N. P. Papanikolopoulos, "Clustering of vehicle trajectories," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 647–657, Sept 2010.

[6] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The r*-tree: An efficient and robust access method for points and rectangles," in *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '90. New York, NY, USA: ACM, 1990, pp. 322–331. [Online]. Available: http://doi.acm.org/10.1145/93597.98741

[7] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication." Springer-Verlag, 1996, pp. 1–15.

[8] A. Bobol and T. Cheng, "Gps data collection setting for pedestrian activity modelling," in *GISRUK 2010: Proceedings of Geographical Information Science Research UK Conference 2010*, Sept 2010.

[9] A. Bolbol, T. Cheng, I. Tsapakis, and J. Haworth, "Inferring hybrid transportation modes from sparse gps data using a moving window svm classification," *Computers, Environment and Urban Systems*, vol. 36, no. 6, pp. 526 – 537, 2012, special Issue: Advances in Geocomputation. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0198971512000543

[10] L. Chen and R. Ng, "On the marriage of lp-norms and edit distance," in *Proceedings 2004 {VLDB} Conference*, M. A. Nascimento, M. T. Özsu, D. Kossmann, R. J. Miller, J. A. Blakeley, and B. Schiefer, Eds. St Louis: Morgan Kaufmann, 2004, pp. 792 – 803. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B978012088469850070X

[11] L. Chen, M. Enzmann, A.-R. Sadeghi, M. Schneider, and M. Steiner, *A Privacy-Protecting Coupon System.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 93–108. [Online]. Available: http://dx.doi.org/10.1007/11507840_12

[12] D.-O. Cho, "The incentive program for fishermen to collect marine debris in korea," *Marine Pollution Bulletin*, vol. 58, no. 3, pp. 415 – 417, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0025326X08004918

[13] P. Ciaccia, M. Patella, F. Rabitti, and P. Zezula, "Indexing metric spaces with m-tree." in *SEBD*, vol. 97, 1997, pp. 67–86.

[14] C. Costello, *Pairings for Beginners*, 2012.

[15] A. De Caro and V. Iovino, "jpbc: Java pairing based cryptography," in *Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011*, Kerkyra, Corfu, Greece, June 28 - July 1, 2011, pp. 850–855.

[16] V. M. Durand, "Behavioral ecology of a staff incentive program: Effects on absenteeism and resident disruptive behavior," *Behavior Modification*, vol. 7, no. 2, pp. 165–181, 1983. [Online]. Available: http://bmo.sagepub.com/content/7/2/165.abstract

[17] M. Enzmann, M. Fischlin, and M. Schneider, *A Privacy-Friendly Loyalty System Based on Discrete Logarithms over Elliptic Curves.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 24–38. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-27809-2_4

[18] "Directive 2010/40/eu of the european parliament and the council, on the framework for the deployment of intelligent transport systems in the field of road transport and for interfaces with other modes of transport," European Parliament, Jul 2010.

[19] M. Gašparović, P. Nicolau, A. Marques, C. Silva, and L. Marcelino, "On privacy in user tracking mobile applications," in *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, June 2016, pp. 1–6.

[20] H. Gong, C. Chen, E. Bialostozky, and C. T. Lawson, "A gps/gis method for travel mode detection in new york city," *Computers, Environment and Urban Systems*, vol. 36, no. 2, pp. 131 – 139, 2012, special Issue: Geoinformatics 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0198971511000536

[21] L. Gregorios-Pippas, P. N. Tobler, and W. Schultz, "Short-term temporal discounting of reward value in human ventral striatum," *Journal of Neurophysiology*, vol. 101, no. 3, pp. 1507–1523, 2009.

[22] K. Kasori and F. Sato, "Location privacy protection considering the location safety," in *Network-Based Information Systems (NBiS), 2015 18th International Conference on*, Sept 2015, pp. 140–145.

[23] E. Keogh and A. C. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358–386, 2005. [Online]. Available: http://dx.doi.org/10.1007/s10115-004-0154-9

[24] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explor. Newsl.*, vol. 12, no. 2, pp. 74–82, Mar. 2011. [Online]. Available: http://doi.acm.org/10.1145/1964897.1964918

[25] J. Z. Li, M. T. Ozsu, and D. Szafron, "Modeling of moving objects in a video database," in *Multimedia Computing and Systems '97. Proceedings., IEEE International Conference on*, Jun 1997, pp. 336–343.

[26] D. B. Lindenmayer, C. Zammit, S. J. Attwood, E. Burns, C. L. Shepherd, G. Kay, and J. Wood, "A novel and cost-effective monitoring approach for outcomes in an australian biodiversity conservation incentive program," *PLoS ONE*, vol. 7, no. 12, pp. 1–11, 12 2012. [Online]. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0050872

[27] N. Magdy, M. A. Sakr, T. Mostafa, and K. El-Bahnasy, "Review on trajectory similarity measures," in *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, Dec 2015, pp. 613–619.

[28] P. Marchetta, A. Salvi, E. Natale, A. Tirri, M. Tufo, and D. D. Pasquale, "S2-move: Smart and social move," in *2012 Global Information Infrastructure and Networking Symposium (GIIS)*, Dec 2012, pp. 1–6.

[29] P. F. Marteau, "Time warp edit distance with stiffness adjustment for time series matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 306–318, Feb 2009.

[30] A. Mayoue, Q. Barthélemy, S. Onis, and A. Larue, "Preprocessing for classification of sparse data: Application to trajectory recognition," in *2012 IEEE Statistical Signal Processing Workshop (SSP)*, Aug 2012, pp. 37–40.

[31] H. Mäenpää, A. Lobov, and J. Martinez Lastra, "Travel mode estimation for multimodal journey planner," *Transportation Research Part C: Emerging technologies, Submitted Jan. 27 2017*, Jan 2017.

[32] L. Nguyen, *Privacy-Protecting Coupon System Revisited.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 266–280. [Online]. Available: http://dx.doi.org/10.1007/11889663_22

[33] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 119–128, Jan 2006.

[34] A. Pham, K. Huguenin, I. Bilogrevic, I. Dacosta, and J. P. Hubaux, "Securerun: Cheat-proof and private summaries for location-based activities," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 2109–2123, Aug 2016.

[35] S. Reddy, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Determining transportation mode on mobile phones," in *2008 12th IEEE International Symposium on Wearable Computers*, Sept 2008, pp. 25–28.

[36] S. Reddy, K. Shilton, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, *Using Context Annotated Mobility Profiles to Recruit Data Collectors in Participatory Sensing.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 52–69. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01721-6_4

[37] P. Ruppel, G. Treu, A. Küpper, and C. Linnhoff-Popien, "Anonymous user tracking for location-based community services," in *International Symposium on Location-and Context-Awareness.* Springer, 2006, pp. 116–133.

[38] X. Su, H. Tong, and P. Ji, "Activity recognition with smartphone sensors," *Tsinghua Science and Technology*, vol. 19, no. 3, pp. 235–249, June 2014.

[39] Z. Sun and X. J. Ban, "Vehicle classification using {GPS} data," *Transportation Research Part C: Emerging Technologies*, vol. 37, pp. 102 – 117, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X13002040

[40] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh, "Indexing multi-dimensional time-series with support for multiple distance measures," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '03. New York, NY, USA: ACM, 2003, pp. 216–225. [Online]. Available: http://doi.acm.org/10.1145/956750.956777

[41] X. Zhu, J. Li, Z. Liu, S. Wang, and F. Yang, "Learning transportation annotated mobility profiles from gps data for context-aware mobile services," in *2016 IEEE International Conference on Services Computing (SCC)*, June 2016, pp. 475–482.

ALL FEATURES' CORRELATIONS

This table shows the correlations between all the considered features. To save space, the columns have been left unlabeled. However, the order from left to right is the same as the order from top to bottom.

Feature																																		
Minimum Speed (m/s)	1.0	0.70	0.86	0.81	0.10	0.28	-0.12	0.032	0.076	0.097	0.15	0.086	-0.055	-0.086	0.27	0.60	0.74	0.69	0.26	0.14	0.17	0.14	0.14	0.12	0.0084	-0.29	-0.082	0.16	-0.19	-0.19	-0.051	0.067	0.14	0.056	0.10	0.017
Maximum Speed (m/s)	0.70	1.0	0.95	0.93	0.77	0.33	0.040	0.42	0.71	0.68	0.67	0.67	0.029	0.077	-0.18	0.87	0.95	0.91	0.77	0.65	0.67	0.65	0.65	0.16	0.035	-0.26	0.048	0.15	-0.22	-0.10	-0.11	0.093	0.061	0.029	0.056	-0.024
Average Speed (m/s)	0.86	0.95	1.0	0.98	0.58	0.23	-0.018	0.30	0.51	0.51	0.52	0.49	-0.0046	0.0065	0.0068	0.83	0.94	0.91	0.63	0.48	0.51	0.49	0.49	0.16	0.011	-0.29	-0.012	0.16	-0.21	-0.12	-0.062	0.056	0.057	0.052	0.091	0.016
Median speed (m/s)	0.81	0.93	0.98	1.0	0.59	0.15	-0.026	0.32	0.53	0.53	0.52	0.51	-0.0050	-0.0029	-0.032	0.80	0.92	0.90	0.65	0.50	0.53	0.51	0.51	0.14	0.00018	-0.29	-0.018	0.14	-0.16	-0.11	-0.016	0.037	0.012	0.055	0.095	0.018
Speed σ (m/s)	0.10	0.77	0.58	0.59	1.0	0.21	0.061	0.59	0.92	0.90	0.82	0.86	0.040	0.10	-0.49	0.63	0.67	0.66	0.83	0.77	0.78	0.78	0.78	0.097	0.028	-0.12	0.11	0.070	-0.12	0.024	-0.11	0.061	-0.021	0.014	0.0093	-0.036
Speed skewness (m/s)	0.28	0.33	0.23	0.15	0.21	1.0	-0.12	0.14	0.21	0.20	0.21	0.19	0.025	0.097	-0.27	0.31	0.30	0.23	0.32	0.31	0.31	0.29	0.27	0.045	0.11	-0.037	0.037	0.12	-0.23	-0.14	-0.38	0.36	0.32	-0.081	-0.065	-0.15
Speed kurtosis (m/s)	-0.12	0.040	-0.018	-0.026	0.061	-0.12	1.0	-0.20	0.15	-0.039	-0.054	0.19	0.32	0.38	-0.052	0.16	0.024	0.010	0.085	0.071	0.048	0.063	0.091	0.11	0.00079	0.022	0.30	-0.19	-0.10	0.014	0.10	-0.042	-0.14	-0.24	-0.21	-0.17
Minimum Acceleration (m/s^2)	0.032	0.42	0.30	0.32	0.59	0.14	-0.20	1.0	0.55	0.78	0.66	0.33	-0.11	-0.037	-0.43	0.25	0.34	0.33	0.52	0.50	0.53	0.52	0.42	-0.27	0.22	-0.026	0.25	-0.072	-0.097	-0.10	-0.042	-0.022	0.084	0.040	0.078	0.013
Maximum Acceleration (m/s^2)	0.076	0.71	0.51	0.53	0.92	0.21	0.15	0.55	1.0	0.91	0.80	0.95	0.15	0.20	-0.48	0.61	0.62	0.61	0.81	0.76	0.77	0.78	0.77	-0.094	0.12	0.0016	0.18	-0.0078	-0.16	-0.00053	-0.082	0.010	-0.0068	-0.046	-0.022	-0.049
Average Acceleration (m/s^2)	0.097	0.68	0.51	0.53	0.90	0.20	-0.039	0.78	0.91	1.0	0.92	0.81	-0.13	0.0068	-0.50	0.52	0.59	0.58	0.79	0.74	0.77	0.76	0.72	-0.20	0.15	0.022	0.22	-0.035	-0.16	-0.022	-0.091	-0.00033	0.043	0.022	0.059	0.0094
Median acceleration (m/s^2)	0.15	0.67	0.52	0.52	0.82	0.21	-0.054	0.66	0.80	0.92	1.0	0.72	-0.23	-0.060	-0.40	0.51	0.59	0.57	0.72	0.68	0.71	0.69	0.66	-0.17	0.13	0.040	0.17	0.031	-0.24	-0.022	-0.088	0.030	0.053	0.026	0.068	0.019
Acceleration σ (m/s^2)	0.086	0.67	0.49	0.51	0.86	0.19	0.19	0.33	0.95	0.81	0.72	1.0	0.14	0.097	-0.39	0.55	0.59	0.59	0.73	0.66	0.68	0.69	0.72	-0.050	0.049	-0.0046	0.099	0.013	-0.13	0.032	-0.098	0.014	-0.050	-0.027	-0.0071	-0.035
Acceleration skewness (m/s^2)	-0.055	0.029	-0.0046	-0.0050	0.040	0.025	0.32	-0.11	0.15	-0.13	-0.23	0.14	1.0	0.46	-0.13	0.16	0.044	0.070	0.092	0.10	0.076	0.11	0.13	0.24	0.0082	-0.16	0.087	0.0017	-0.022	-0.012	-0.011	0.067	-0.044	-0.094	-0.20	-0.16
Acceleration kurtosis (m/s^2)	-0.086	0.077	0.0065	-0.0029	0.10	0.097	0.38	-0.037	0.20	0.0068	-0.060	0.097	0.46	1.0	-0.22	0.35	0.067	0.043	0.27	0.28	0.22	0.25	0.23	0.16	0.14	0.061	0.22	0.0075	-0.051	0.038	-0.0026	0.064	0.042	-0.23	-0.35	-0.34
Bus route dissimilarity	0.27	-0.18	0.0068	-0.032	-0.49	-0.27	-0.052	-0.43	-0.48	-0.50	-0.40	-0.39	-0.13	-0.22	1.0	-0.28	-0.18	-0.16	-0.57	-0.63	-0.61	-0.61	-0.59	0.064	-0.17	-0.066	-0.24	0.041	0.065	-0.041	0.14	-0.14	-0.015	0.084	0.13	0.0079
Speed spectrum 0 mHz-1 mHz	0.60	0.87	0.83	0.80	0.63	0.31	0.16	0.25	0.61	0.52	0.51	0.55	0.16	0.35	-0.28	1.0	0.91	0.85	0.80	0.71	0.68	0.67	0.68	0.27	0.10	-0.18	0.17	0.18	-0.27	-0.092	-0.13	0.14	0.051	-0.065	-0.046	0.0041
Speed spectrum 1 mHz-2 mHz	0.74	0.95	0.94	0.92	0.67	0.30	0.024	0.34	0.62	0.59	0.59	0.59	0.044	0.067	-0.18	0.91	1.0	0.96	0.74	0.63	0.65	0.63	0.62	0.18	0.054	-0.27	0.070	0.15	-0.29	-0.11	-0.11	0.13	0.054	0.012	0.048	-0.0072
Speed spectrum 2 mHz-4 mHz	0.69	0.91	0.91	0.90	0.66	0.23	0.010	0.33	0.61	0.58	0.57	0.59	0.070	0.043	-0.16	0.85	0.96	1.0	0.71	0.58	0.61	0.60	0.59	0.16	0.021	-0.28	0.033	0.15	-0.23	-0.11	-0.091	0.14	0.020	0.0080	0.052	-0.0010
Speed spectrum 4 mHz-8 mHz	0.26	0.77	0.63	0.65	0.83	0.32	0.085	0.52	0.81	0.79	0.72	0.73	0.092	0.27	-0.57	0.80	0.74	0.71	1.0	0.89	0.88	0.88	0.87	0.052	0.12	-0.017	0.22	0.018	-0.14	-0.0096	-0.18	0.16	0.031	-0.090	-0.039	-0.029
Acceleration spectrum 0 mHz-1 mHz	0.14	0.65	0.48	0.50	0.77	0.31	0.071	0.50	0.76	0.74	0.68	0.66	0.10	0.28	-0.63	0.71	0.63	0.58	0.89	1.0	0.99	0.99	0.97	0.060	0.14	-0.0050	0.24	0.061	-0.19	-0.020	-0.12	0.094	0.076	-0.069	-0.071	-0.018
Acceleration spectrum 1 mHz-2 mHz	0.17	0.67	0.51	0.53	0.78	0.31	0.048	0.53	0.77	0.77	0.71	0.68	0.076	0.22	-0.61	0.68	0.65	0.61	0.88	0.99	1.0	0.99	0.97	0.029	0.13	-0.025	0.22	0.049	-0.20	-0.028	-0.12	0.090	0.078	-0.071	-0.052	-0.023
Acceleration spectrum 2 mHz-4 mHz	0.14	0.65	0.49	0.51	0.78	0.29	0.063	0.52	0.78	0.76	0.69	0.69	0.11	0.25	-0.61	0.67	0.63	0.60	0.88	0.99	0.99	1.0	0.97	0.029	0.13	-0.025	0.23	0.037	-0.19	-0.032	-0.10	0.084	0.077	-0.064	-0.056	-0.036
Acceleration spectrum 4 mHz-8 mHz	0.14	0.65	0.49	0.51	0.78	0.27	0.091	0.42	0.77	0.72	0.66	0.72	0.13	0.23	-0.59	0.68	0.62	0.59	0.87	0.97	0.97	0.97	1.0	0.11	0.067	-0.031	0.13	0.10	-0.11	0.0089	-0.12	0.094	0.054	-0.059	-0.068	-0.030
Speed autocorrelation 1	0.12	0.16	0.16	0.14	0.097	0.045	0.11	-0.27	-0.094	-0.20	-0.17	-0.050	0.24	0.16	0.064	0.27	0.18	0.16	0.052	0.060	0.029	0.029	0.11	1.0	-0.60	-0.35	-0.28	0.34	0.0049	0.064	-0.064	0.16	-0.034	-0.0075	-0.092	-0.075
Speed autocorrelation 2	0.0084	0.035	0.011	0.00018	0.028	0.11	0.00079	0.22	0.12	0.15	0.13	0.049	0.0082	0.14	-0.17	0.10	0.054	0.021	0.12	0.14	0.13	0.13	0.067	-0.60	1.0	-0.22	0.41	-0.20	-0.14	-0.089	-0.021	-0.087	0.072	0.0057	-0.017	-0.044
Speed autocorrelation 3	-0.29	-0.26	-0.29	-0.29	-0.12	-0.037	0.022	-0.026	0.0016	0.022	0.040	-0.0046	-0.16	0.061	-0.066	-0.18	-0.27	-0.28	-0.017	-0.0050	-0.025	-0.025	-0.031	-0.35	-0.22	1.0	0.037	-0.098	0.077	0.24	-0.052	-0.067	-0.00017	-0.032	-0.049	0.024
Acceleration autocorrelation 1	-0.082	0.048	-0.012	-0.018	0.11	0.037	0.30	0.25	0.18	0.22	0.17	0.099	0.087	0.22	-0.24	0.17	0.070	0.033	0.22	0.24	0.22	0.23	0.13	-0.28	0.41	0.037	1.0	-0.66	-0.23	-0.055	-0.042	-0.0070	0.018	-0.11	-0.075	0.00083
Acceleration autocorrelation 2	0.16	0.15	0.16	0.14	0.070	0.12	-0.19	-0.072	-0.0078	-0.035	0.031	0.013	0.0017	0.0075	0.041	0.18	0.15	0.15	0.018	0.061	0.049	0.037	0.10	0.34	-0.20	-0.098	-0.66	1.0	-0.23	0.022	-0.038	0.038	0.037	-0.0065	-0.066	0.0027
Acceleration autocorrelation 3	-0.19	-0.22	-0.21	-0.16	-0.12	-0.23	-0.10	-0.097	-0.16	-0.16	-0.24	-0.13	-0.022	-0.051	0.065	-0.27	-0.29	-0.23	-0.14	-0.19	-0.20	-0.19	-0.11	0.0049	-0.14	0.077	-0.23	-0.23	1.0	0.10	0.11	-0.077	-0.14	0.063	0.074	0.037
a/v cross correlation -3	-0.19	-0.10	-0.12	-0.11	0.024	-0.14	0.014	-0.10	-0.00053	-0.022	-0.022	0.032	-0.012	0.038	-0.041	-0.092	-0.11	-0.11	-0.0096	-0.020	-0.028	-0.032	0.0089	0.064	-0.089	0.24	-0.055	0.022	0.10	1.0	-0.26	-0.042	-0.29	0.020	0.0045	-0.0089
a/v cross correlation -2	-0.051	-0.11	-0.062	-0.016	-0.11	-0.38	0.10	-0.042	-0.082	-0.091	-0.088	-0.098	-0.011	-0.0026	0.14	-0.13	-0.11	-0.091	-0.18	-0.12	-0.12	-0.10	-0.12	-0.064	-0.021	-0.052	-0.042	-0.038	0.11	-0.26	1.0	-0.49	-0.15	0.022	-0.021	-0.038
a/v cross correlation -1	0.067	0.093	0.056	0.037	0.061	0.36	-0.042	-0.022	0.010	-0.00033	0.030	0.014	0.067	0.064	-0.14	0.14	0.13	0.14	0.16	0.094	0.090	0.084	0.094	0.16	-0.087	-0.067	-0.0070	0.038	-0.077	-0.042	-0.49	1.0	-0.14	-0.071	-0.039	-0.015
a/v cross correlation 0	0.14	0.061	0.057	0.012	-0.021	0.32	-0.14	0.084	-0.0068	0.043	0.053	-0.050	-0.044	0.042	-0.015	0.051	0.054	0.020	0.031	0.076	0.078	0.077	0.054	-0.034	0.072	-0.00017	0.018	0.037	-0.14	-0.29	-0.15	-0.14	1.0	0.071	0.0024	-0.058
a/v cross correlation 1	0.056	0.029	0.052	0.055	0.014	-0.081	-0.24	0.040	-0.046	0.022	0.026	-0.027	-0.094	-0.23	0.084	-0.065	0.012	0.0080	-0.090	-0.069	-0.071	-0.064	-0.059	-0.0075	0.0057	-0.032	-0.11	-0.0065	0.063	0.020	0.022	-0.071	0.071	1.0	0.24	-0.097
a/v cross correlation 2	0.10	0.056	0.091	0.095	0.0093	-0.065	-0.21	0.078	-0.022	0.059	0.068	-0.0071	-0.20	-0.35	0.13	-0.046	0.048	0.052	-0.039	-0.071	-0.052	-0.056	-0.068	-0.092	-0.017	-0.049	-0.075	-0.066	0.074	0.0045	-0.021	-0.039	0.0024	0.24	1.0	0.26
a/v cross correlation 3	0.017	-0.024	0.016	0.018	-0.036	-0.15	-0.17	0.013	-0.049	0.0094	0.019	-0.035	-0.16	-0.34	0.0079	0.0041	-0.0072	-0.0010	-0.029	-0.018	-0.023	-0.036	-0.030	-0.075	-0.044	0.024	0.00083	0.0027	0.037	-0.0089	-0.038	-0.015	-0.058	-0.097	0.26	1.0

F-STATISTICS IN ORDER OF SIGNIFICANCE

#	Feature	f-statistic
1	Speed spectrum $1-2mHz$	382
2	Average speed	308
3	Median speed	286
4	Speed spectrum $2-4mHz$	265
5	Acceleration spectrum $1-2mHz$	207
6	Speed spectrum $4-8mHz$	181
7	Acceleration spectrum $0-1mHz$	180
8	Acceleration spectrum $2-4mHz$	175
9	Acceleration spectrum $4-8mHz$	169
10	Speed spectrum $0mHz-1mHz$	159
11	Bus route dissimilarity	145
12	Average Acceleration	118
13	Speed variance	114
14	Median Acceleration	88.7
15	Maximum Acceleration	84.3
16	Acceleration variance	68.1
17	Minimum Acceleration	50.6