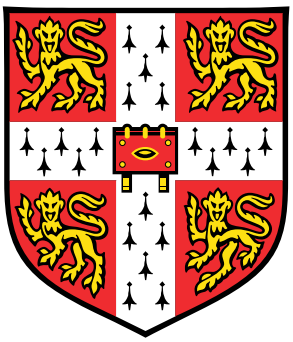
Understanding travel mode choice: A new approach for city scale simulation



Timothy Michael Hillel

Department of Engineering

University of Cambridge

This dissertation is submitted for the degree of

Doctor of Philosophy

Downing College January 2019

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

Timothy Michael Hillel
January 2019

Acknowledgements

First and foremost, I would like to express my gratitude to Dr Ying Jin for his continued help and support over the course of my PhD. His enthusiasm, patience, knowledge, and generosity with his time played a huge role in this process, and I look forward to collaborating again in future. I am also grateful to Dr Mohammed Elshafie, for managing to keep me on track for the last three and a half years, and whose critique was valuable throughout this research. I would also like to thank my examiners, Prof. Michael Batty and Dr Damon Wischik for their valuable feedback.

Thanks are also due to Prof. Peter Guthrie, who helped me to develop my scribbled thoughts into a viable research idea prior to starting my PhD, and who has offered me continued counsel throughout the process. I would also like to thank Gerry Casey for introducing me to the LTDS data and the Google directions API.

I was very fortunate in having the opportunity to spend three months during my PhD working with Prof. Michel Bierlaire within the Transport and Mobility Laboratory at EPFL, for which I am hugely grateful. Michel's advice on the theoretical framework, as well as the outcomes of our numerous discussions, helped to formalise the presentation throughout. Thanks must also go to Dr Virginie Lurkin and Gael Lederrey, for our impromptu conceptual discussions, which inspired much of the work on the assisted specification approach.

I would like to acknowledge Transport for London for their provision of the LTDS data and permission to make the processed dataset openly available, with particular thanks to David Wilby and Hannah Groot for their support.

I am grateful also for my funding from the UK Engineering and Physical Sciences Research Council via the Future Infrastructure and Built Environment Centre for Doctoral Training (EP/L016095/1).

As well as those who gave me practical support, there are also those who have helped make my life as a PhD student far more enjoyable than I could have anticipated. Many thanks to my FIBE cohort of Bryn Pickering, Christiana Smyrilli, Hannah Baker, Jason Pelekis, Marina Konstantatou, and Petia Tzokova, as well as the wider FIBE family. Long may we wave the flag of interdisciplinary civil engineering.

Also, to the students, staff, and fellows of Downing College that I have had the pleasure of getting to know over the past eight years, as well as my Ross Street housemates past and present, thank you for making Cambridge truly feel like home.

I would like to extend thanks to my high school physics teacher, John Hensman, who, by referring to me as Dr Hillel in our one-on-one conversations, first planted the seed of pursuing an academic career. I am still working on the Hillel theorem however.

My final thanks go to Alexandra Hetmanski, Deirdre Johnson, Laurence Hillel, and Rowena Hillel. Without their love, support, and, in the case of my parents, proofreading abilities, I would not have made it through my PhD. This thesis is dedicated to them.

Abstract

Understanding travel mode choice behaviour is key to effective management of transport networks, many of which are under increasing strain from rising travel demand. Conventional approaches to simulating mode choice typically make use of behavioural models either derived from stated preference choice experiments or calibrated to observed average mode shares. Whilst these models have played and continue to play a key role in economic, social, and environmental assessments of transport investments, there is growing need to gain a deeper understanding of how people interact with transport services, through exploiting available but fragmented data on passenger movements and transport networks.

This thesis contributes to this need through developing a novel approach for urban mode choice prediction and applying it to historical trip records in the Greater London area. The new approach consists of two parts: (i) a data generation framework which combines multiple data-sources to build trip datasets containing the likely mode-alternative options faced by a passenger at the time of travel, and (ii) a modelling framework which makes use of these datasets to fit, optimise, validate, and select mode choice classifiers. This approach is used to compare the relative predictive performance of a complete suite of Machine Learning (ML) classification algorithms, as well as traditional utility-based choice models. Furthermore, a new assisted specification approach, where a fitted ML classifier is used to inform the utility function structure in a utility-based choice model, is then explored.

The results identify three key findings. Firstly, the Gradient Boosting Decision Trees (GBDT) model is the highest performing classifier for this task. Secondly, the relative differences in predictive performance between classifiers are far smaller than has been suggested by previous research. In particular, there is a much smaller performance gap identified between Random Utility Models (RUMs) and ML classifiers. Finally, the assisted specification approach is successful in using the structure of a fitted ML classifier to improve the performance of a RUM. The resulting model achieves significantly better performance than all but the GBDT ML classifier, whilst maintaining a robust, interpretable behavioural model.

Table of contents

List of figures

List of tables

List of acronyms

AB AdaBoost. 31

ABM Agent Based Model. 68

AI Artificial Intelligence. 19

AIC Akaike Information Criterion. 89, 98, 135, 136, 165

AMPCA Arithmetic Mean Probability of Correct Assignment. 46, 82, 83, 85–87, 137, 139

ANN Artificial Neural Network. ix, 12, 13, 18, 19, 31, 32, 46, 49, 50, 100, 101, 139, 172

API Application Programming Interface. xi, 42, 107, 108, 110, 111, 113, 122, 176

ASC Alternative Specific Constant. 6, 9, 11, 81, 89, 90, 132, 135, 161, 162

BIC Bayesian Information Criterion. 47

BL Bayesian Learner. 31

BN Bayesian Network. 31

CART Classification and Regression Trees. 14

CEL Cross-Entropy Loss. xiii, xiv, xvi, 82, 84–86, 88, 92, 94, 97, 98, 101, 102, 140–143, 147–149, 156, 159, 165, 166

CLT Central Limit Theorem. 142

CNL Cross-Nested Logit. 5, 8, 10, 31, 33, 90, 91, 99, 131, 132, 134, 135

CNN Convolutional Neural Network. 139

CPU Central Processing Unit. 152

CV Cross-Validation. xvi, 147, 148, 150, 152–154

d.o.f. degrees of freedom. 95

DCA Discrete Classification Accuracy. 82, 83, 85, 86, 137, 139, 140, 148, 164

DCM Discrete Choice Model. 2, 32, 54, 174

DLR Docklands Light Railway. 121

DT Decision Tree. ix, 12, 14–16, 18, 20, 31, 32, 42, 46, 49, 104, 137, 139, 151, 157, 169, 173

EL Ensemble Learning. ix, 12, 15, 16, 18, 20, 31, 33, 34, 49, 54, 100, 104, 105, 137, 139, 145, 148, 151, 153, 154, 168–170, 172–174, 176

ET Extremely randomised Trees. xiii, xvi, xvii, 15, 16, 18, 100, 102, 137, 141–143, 145, 147, 154, 174, 182, 184, 187

FFNN Feed-Forward Neural Network. xvi, xvii, 13, 14, 31, 100, 101, 137, 139, 141, 147, 150–155, 181, 184, 186

FL Fuzzy Logic. 19, 23

GB Gradient Boosting. 31

GBDT Gradient Boosting Decision Trees. vii, xiv, xvi, xvii, 15, 16, 18, 33, 34, 54, 100, 101, 103, 132, 137, 139, 141, 143, 147, 148, 151, 152, 155, 157, 161–164, 166, 167, 169, 172–174, 182, 185, 187, 188

GMPCA Geometric Mean Probability of Correct Assignment. xvi, 82–87, 136, 137, 139–141, 145, 147, 148, 150, 154, 164, 165

GP Gaussian Process. 87

GPS Global Positioning System. 25, 36, 110

GPU Graphics Processing Unit. 152

HTTP Hypertext Transfer Protocol. 110

i.i.d. independent and identically distributed. 7, 69, 84

IIA Independence from Irrelevant Alternatives. 7, 8

IVTT In-Vehicle Travel Time. 40

KDE Kernel Density Estimation. xiv, 162, 163

LOS Level of Service. 33, 39, 44

LR Logistic Regression. ix, xvi, xvii, 9, 12–14, 18, 30, 32–34, 47, 49, 100, 101, 137, 139–141, 151, 153, 155, 172, 174, 181, 183, 186

LTDS London Travel Demand Survey. xi, xiii, xv, xvii, 4, 79, 98, 99, 103, 107–109, 111, 113, 115–119, 121–123, 125, 129, 133, 134, 155, 161, 167, 172, 176, 177, 190, 191

ML Machine Learning. vii, xi, xiii–xv, 2–5, 9, 12, 17–20, 22, 23, 29, 30, 32–34, 49, 51, 53, 54, 57, 63, 65, 67, 70, 75, 77, 80–82, 84, 87, 88, 91, 92, 95, 97, 99, 100, 103, 105, 130–132, 136–138, 140, 148–153, 155, 157, 159, 161, 165, 167–175

MLE Maximum Likelihood Estimation. 7, 10, 81, 84, 100, 101, 143

MLP Multi-Layer Perceptron. 13, 50

MNL Multinomial Logit. 5, 7, 9, 10, 13, 14, 31, 33, 81, 89–91, 98, 105, 131–135, 161, 162, 164–166, 170, 172–174

MSE Mean Squared Error. xvi, 46, 47, 49, 145, 147

NaPTAN National Public Transport Access Nodes. 108, 122

NB Naïve Bayes. 31

NL Nested Logit. xvii, 5, 8, 10, 31, 33, 90, 91, 99, 131, 132, 134, 135, 137, 153, 165, 172, 174, 192, 193

O-D Origin-Destination. xiii, 80, 112, 177

OOB Out-Of-Bootstrap. 46

OVTT Out-of-Vehicle Travel Time. 40

PDF Probability Density Function. 94, 95, 141, 142

PNN Probabilistic Neural Network. 31

PRISMA Preferred Reporting Items for Systematic Reviews and Meta-Analyses. 23, 25, 26, 54

PT Public Transport. 40, 41

RBF Radial Basis Function. 16, 155

RBFNN Radial Basis Function Neural Network. 31

RBML Rule-Based Machine Learning. 23, 31, 32

RF Random Forest. xiii, xvi, xvii, 15, 16, 18, 31, 46, 100, 102, 137, 141–143, 145, 147, 148, 150, 154, 174, 182, 184, 187

ROC Receiver Operating Characteristic. 47

RUM Random Utility Model. vii, xi, xiii, xv–xvii, 1–3, 5–12, 14, 17–19, 30, 32–34, 57, 63, 65, 67, 68, 75, 77, 80, 81, 84, 85, 87, 88, 91, 92, 97, 98, 100, 103–105, 131–133, 135–138, 140, 141, 143, 150–153, 159, 161, 162, 164–167, 169–176, 189–191

SMBO Sequential Model-Based Optimisation. xiv, 87, 91, 148, 149, 153, 168

SQL Structured Query Language. 113

SVM Support Vector Machine. ix, xvi, xvii, 12, 16–18, 20, 31, 49, 50, 100, 102, 137, 139–141, 143, 145, 147, 151–155, 172, 183, 185, 188

TfL Transport for London. 99, 108, 110, 113, 117, 118, 121–123, 130

TPE Tree-structure Parzen Estimator. 87, 88, 97, 148

VOC Vehicle Operating Cost. 40, 79, 80, 108, 123, 133

VoT Value of Time. 10, 33, 67, 89, 133, 135

WebTAG UK DfT Transport Analysis Guidance. 108, 123

Chapter 1

Introduction

1.1 Background & motivation

Predicting travel mode choice is one of the fundamental steps in understanding how people use transport networks and services. Multi-modal transport models, used to inform transport management and investment decisions, require reliable mode choice predictions in order to estimate flows of passengers through a transport system.

In many cities worldwide, urban transport infrastructure is under increasing pressure from rising travel demand. For many of these cities, it is no longer sustainable or even economically viable to cope with increased demand by continually adding capacity to transport networks. Instead, travel demand must be managed by encouraging passengers to adapt their travel behaviour. This approach necessitates a significantly deeper understanding of the diverse and seemingly random variations of passenger flows than is afforded by the current travel demand modelling techniques.

Current approaches to simulating mode choice typically make use of Random Utility Models (RUMs) either derived from *stated preference* choice experiments or calibrated to observed average mode shares. Whilst these models have played and continue to play a key role in economic, social, and environmental assessments of transport investments, the current implementations are of limited functional use when investigating individual behaviours at a high spatial and temporal resolution. There is therefore a growing need to develop techniques which provide a deeper understanding of how people interact with a transport system.

The adoption of several notable transportation-related technologies, such as live travel information feeds, mobile-phone-based location services, contactless smart cards, vehicle tracking cameras, and connected vehicles, has driven a step change in the availability of data on passenger movements of several orders of magnitude. This data provides the opportunity to build much richer models of passenger behaviour which directly infer the causal relationship

between transport and environment conditions and passenger travel decisions. These models could be used to simulate passenger flows with finer spatial, temporal, and behavioural granularity than is possible with current techniques. However, there have been limited attempts at integrating these disparate data sources to create cohesive models.

This thesis contributes to this need through developing a novel approach for passenger travel mode choice prediction and applying it to historical trip records in the Greater London area. The new approach consists of two parts: (1) a data generation framework which combines multiple data-sources to build trip datasets containing the likely mode-alternative options faced by a passenger at the time of travel, and (2) a modelling framework which makes use of these datasets to fit, optimise, validate, and select mode choice classifiers. This approach is used to compare the relative predictive performance of a complete suite of current Machine Learning (ML) classification algorithms, as well as traditional utility-based choice models. Furthermore, a new assisted specification approach, where a fitted ML classifier is used to inform the utility function structure in a utility-based choice model, is then explored.

1.2 Summary of contributions

This thesis is intended to serve two sets of purposes for two different audiences. For the transport modeller who typically estimates RUMs on stated preference data, this thesis intends to

1. present a data generation framework which can be used to build rich datasets from revealed preference data, on which choice-models can be trained (Chapter 6);
2. present ML approaches as an alternative to RUMs, and provide a fair evaluation of their potential and limitations when compared to RUMs (Chapters 5 and 7);
3. introduce a new assisted specification approach, which can provide valuable insights into how to structure a Discrete Choice Model (DCM), including high-order variable interactions and non-linear relationships between input variables and mode choice (Sections 5.6 and 7.4);
4. introduce best practices from the ML community which can be adopted for traditional RUM-based investigations (e.g. formal model validation schemes) (Chapters 4 and 5).

In contrast, for the ML practitioner who wishes to model choice behaviour using ML classification techniques, this thesis intends to

1. identify the pitfalls of using ML for choice prediction in the existing mode choice modelling literature (Chapter 3);

2. establish a new set of good-practices for machine-learning choice modelling, by addressing the pitfalls identified in the existing work (Chapter 5);

3. introduce assisted specification RUMs as an alternative to ML classifiers, which can achieve similar performance to the best ML classifiers whilst relying on a robust and interpretable underlying behavioural model (Sections 5.6 and 7.4).

1.3 Thesis overview

The material in this thesis is divided into eight chapters. This chapter introduces the thesis and summarises the intended contributions.

Next, Chapter 2 presents the background and motivation for mode choice prediction. The chapter aims to (a) present the theory and limitations of the state-of-practice solutions to transport simulation and mode choice prediction, (b) summarise the motivations for new approaches to transport simulation, and (c) introduce ML classification algorithms which can be applied to mode choice prediction.

Chapter 3 presents a systematic review of ML approaches to passenger mode choice modelling. The review focuses on the methodologies employed within each study in order to (a) establish the state-of-research modelling frameworks for ML mode choice prediction, (b) identify and quantify the prevalence of methodological limitations in previous studies, and (c) evaluate the research background to which this work contributes.

Chapter 4 introduces a new theoretical framework in order to formalise the requirements for structuring investigations into classification problems. The chapter aims to (a) establish a rigorous theoretical framework for supervised classification which applies to both ML and statistical RUMs, (b) present a uniform notation which covers the relevant tasks within the classification problem, and (c) assess the technical limitations of existing ML approaches identified by the systematic review (Chapter 3) within this theoretical framework.

Chapter 5 introduces the modelling methodology used within this thesis to investigate travel mode choice. The chapter aims to (a) establish a methodological approach within the context of the theoretical framework (Chapter 4) which addresses the technical limitations identified in the systematic review (Chapter 3), (b) specify a formal modelling framework for both statistical RUMs and ML classifiers, and (c) present plans for investigations into random utility, ML, and assisted specification models of passenger mode choice.

Chapters 3 to 5 are linked sequentially: Chapter 3 identifies the technical limitations in existing research and quantifies their prevalence; Chapter 4 then analyses the technical limitations within a theoretical framework, demonstrating how each technical limitation may

affect the results of the existing studies; and finally, Chapter 5 specifically addresses each methodological limitation, demonstrating how they are solved within this research.

Chapter 6 presents a new framework for recreating passenger mode choice-sets developed for this research. The framework is used to build a dataset combining individual records from the London Travel Demand Survey (LTDS) with closely matched trip trajectories alongside their corresponding mode-alternatives (i.e. the choice-set faced by the passenger at the time of travel), and precise estimates of public transport fares and car operating costs.

Chapter 7 presents the experimental results, following the methodology outlined in Chapter 5 and making use of the data created in Chapter 6. The results cover the investigations into (a) the random utility approach, (b) ML models, and (c) the assisted specification approach.

Finally, Chapter 8 presents the conclusions of the thesis, including a summary of the key findings and an evaluation of the limitations and further work.

Chapter 2

Existing and emerging approaches for mode choice prediction

2.1 Overview

This chapter presents the existing and emerging approaches for mode choice prediction. Firstly, Section 2.2 presents the theory and limitations of the Random Utility Model (RUM), which is the primary approach used in practice for mode choice prediction. Next, Section 2.3 introduces Machine Learning (ML) classification algorithms which are being explored as an alternative for mode choice prediction. Section 2.4 then provides a brief overview of equivalent and nearly equivalent terminology across RUMs and ML. Finally, Section 2.5 summarises the chapter and outlines the scope for the thesis.

2.2 Random utility models

Solutions used both in industry and academic research for modelling passenger mode choice rely almost exclusively on econometric Random Utility Models (RUMs) (McFadden 1981). In order to understand and evaluate the ubiquitous usage of RUMs for mode choice modelling, the following sections describe the theory, operation, and limitations of RUMs. First, Section 2.2.1 presents an overview of random utility theory. Next, Section 2.2.1.1 introduces three RUM structures: Multinomial Logit (MNL), Nested Logit (NL), and Cross-Nested Logit (CNL). Section 2.2.1.3 then discusses the types of data used to estimate RUMs. Finally, Section 2.2.2 summarises the limitations of the random utility approach as it is used in practice.

2.2.1 Random utility theory

Within a RUM, mode choice is treated as a discrete choice. An *agent* is assumed to make a decision by choosing from a finite *choice-set* of mutually exclusive and exhaustive options. In the case of mode choice, a passenger (agent) chooses which mode to take out of the set of possible transport modes (choice-set), for a trip with known origin and destination zones and trip purpose.

Selecting each mode-alternative represents a *utility* (U) to the passenger. The passenger is assumed to choose the mode-alternative with the highest utility. In reality, the true value of the utility is only known fully by the passenger and cannot be observed within the model. Therefore, the model uses a measure of the *observed utility* (V) which is known by the modeller.

An error term ε_{in} is defined by the difference between the utility U_{in} and observed utility V_{in}, such that

$$U_{in} = V_{in} + \varepsilon_{in}. \tag{2.1}$$

The error terms ε represent all unknown or unobserved effects in the model. As ε_i is unknown, it is treated as a random variable (Bhat, Eluru, and Copperman 2008). This introduces a stochastic process to the decision model, as it means there is a probability P_{in} of a decision maker n choosing alternative i from J alternatives, where

$$\begin{aligned} P_{in} &= \text{Prob}(U_{in} > U_{jn} \,\forall\, j \neq i), && (2.2)\\ &= \text{Prob}(V_{in} + \varepsilon_{in} > V_{jn} + \varepsilon_{jn} \forall\ j \neq i), && (2.3)\\ &= \text{Prob}(\varepsilon_{in} - \varepsilon_{jn} < V_{in} - V_{jn} \forall\ j \neq i). && (2.4) \end{aligned}$$

The observed utility V is defined as a linear function of the *variables* (or attributes) X_k of each mode-alternative, e.g. the trip cost, duration, etc. Each variable has an associated *parameter* β_k within the linear function. The parameters determine the impact of each variable on the observable utility V. One additional parameter β_0, known as the Alternative Specific Constant (ASC), sets the intercept (i.e. the default bias towards a particular mode). Thus, for mode i being considered for trip n

$$V_{in} = \beta_{0,i} + \sum_k \beta_{ki} X_{k,i,n} \tag{2.5}$$

This is known as the *utility function*.

The parameters in the utility function are unknown, and as such must be *estimated* from a dataset of mode choices and associated attributes. This is achieved using Maximum Likelihood Estimation (MLE) (see Section 5.2.4).

The model parameters ($\beta_{0,i}$ and β_{ki}) may also be dependent on a set of *covariates*. These are attributes of the passenger or trip (e.g. age, gender, journey purpose, departure time) which either affect the tendency (bias) towards a particular mode for a trip (β_0) or the impacts of variables in the trip (β_k). The covariates therefore represent the behavioural heterogeneity for different types of passengers and trip. Typically, the covariates are used to define a finite number of discrete socio-economic/trip groups s (this requires continuous covariates to be *binned* into discrete categories). Separate parameters are then estimated for each group s, so that

$$V_{in} = \beta_{0,i,s} + \sum_k \beta_{k,i,s} X_{k,i,n}. \tag{2.6}$$

A RUM therefore requires three elements to be specified: (i) the assumed distribution of the error terms ε_{in}, (ii) a set of utility specifications for the observed utilities V_i (one for each mode), and (iii) a dataset of mode choices from which to estimate the parameters $\beta_{k,i,s}$. These are presented respectively in Sections 2.2.1.1 to 2.2.1.3.

2.2.1.1 Model structures

The simplest model structure, Multinomial Logit (MNL), is obtained by assuming each error term ε_{in} is independent and identically distributed (i.i.d.) extreme value. Train (2009) shows that, by substituting this assumption into Eq. (2.4), it is possible to derive the choice probabilities for the MNL model

$$P_{in} = \frac{e^{V_{in}}}{\sum_{j=1}^{J} e^{V_{jn}}}. \tag{2.7}$$

This is also known as the *softmax* function.

A major limitation of the MNL model is the Independence from Irrelevant Alternatives (IIA) assumption on the error terms. This implies that substitution occurs proportionally from all modes - e.g. increasing the utility of cycling would result in the same proportional reduction in the probabilities of all the other modes. This does not hold true if the choices between two modes were correlated (e.g. if people tend to substitute cycling with walking more than the other modes). This limitation is typified by the red-bus/blue-bus paradox (Ben-Akiva and Lerman 1985), where a bus service is 'split' into two separate bus services by painting half of the existing red buses blue (the services are otherwise identical). Under the IIA assumption, trips to the new blue buses would be substituted equally from all existing

modes (e.g. red-bus and driving), therefore increasing the predicted overall mode share of bus trips.

The Nested Logit (NL) model addresses the IIA assumption by partitioning the choice set into nests, thus allowing for correlation among alternatives considered in the model (McFadden 1981). In the NL model each alternative mode belongs to exactly one nest (under this definition, a nest is able to contain a single mode). The choice is then effectively modelled in a two-stage process. In the first stage, the choice between nests is modelled. The choice between alternatives belonging to nests with more than one mode is then modelled in the second stage. Each nest containing more than one alternative has an associated scale parameter μ which must also be estimated during model fitting. This scale parameter determines the share of trips the nest represents in the first modelling stage and is bound between one (when the alternatives in the nest are completely independent), and infinity (when the modes in the nest are completely correlated).

The Cross-Nested Logit (CNL) model is a further generalisation of the NL model, where each alternative may belong to one or more nests. As well as the scale parameter μ estimated for each nest, additional α parameters must be estimated for all alternatives belonging to two or more nests. $n-1$ α-parameters are needed for an alternative belonging to n nests. The parameter α defines the proportion to which an alternative belongs to a particular nest and is bound between zero (alternative does not belong at all to that nest) and one (alternative belongs purely to that nest). As with the NL, the CNL also effectively models the choice in a two-stage process.

2.2.1.2 Utility specifications

As discussed, the utility specifications in a RUM define the interaction between attributes in the dataset (see Section 2.2.1.3) and the observed utilities, via the model parameters. For statistical modelling, the meaning and values of these parameters is crucial for *describing* passenger behaviour. As such, there is a strong focus on devising utility specifications which represent a theoretically robust behavioural model and which result in parameter values which conform to expected behavioural conventions.

The utility specifications in a statistical model are therefore typically explicitly specified for each mode, subject to a set of constraints. For example, typically the attributes of a mode are used *only* to determine the observable utility of that mode (i.e. the expected walking duration would not be included in the utility specification for cycling).

The utility specifications provide RUMs two primary advantages over other modelling approaches: *interpretability* and *robustness*. The parameter values in the utility specifications of a RUM present a complete explanation of the scale and direction (positive/negative) of

influence of every input variable on the utility for each relevant class. Whilst increasing model complexity by adding more parameters makes the join relationships more complicated when considering all features simultaneously, it is always possible to understand the impact changing each feature individually has on the class utilities, no matter how complex the model. This results in a high level of *interpretability*.

Additionally, it is possible to check that the utility specifications are consistent with established behavioural theory (through investigating the parameter signs/magnitudes and parametric significance tests). This ensures that the RUMs describe *robust* behavioural models which do not contradict established theory.

Note that it is possible to fit a MNL using all attributes in the data indiscriminately (i.e. without the constraints of the behavioural model). Within this thesis, a distinction is therefore made between statistical RUMs, which make use of a constrained behavioural model with explicitly specified utility functions, and Logistic Regression (LR) classifiers, which do not.

The discussion in this section (Section 2.2) relates to statistical RUMs with an explicit behavioural model, hereby simply referred to as RUMs. Logistic Regression (LR), which uses the same mathematical foundations without the behavioural constraints of statistical RUMs, are presented instead as a ML classifier, and introduced in Section 2.3.

2.2.1.3 Datasets

As discussed in Section 2.2.1, the parameters in a RUM are unknown, and must be estimated from a dataset. There are two broad categories of data used in practice: *stated preference* and *revealed preference*.

Stated preference data is collected from controlled questionnaires where an individual evaluates the choice they would make in several predetermined, hypothetical choice situations. The model parameters can then be estimated using this data, with the ASCs calibrated to mode-shares from observed passenger head counts. The primary advantage of stated preference data is that the modeller knows the exact details (attributes) of the choice-set faced by the decision maker. Additionally, as the choice situations are predetermined, it is possible to ensure in advance that there is sufficient coverage of each choice situation over the desired range of each input variable to estimate the model. However, stated preference data has a fundamental limitation in that it describes only hypothetical choices, and there is no guarantee that these hypothetical choices are representative of real-world behaviour.

Conversely, revealed preference data is collected by recording details of historic trips or travel behaviour. This has the opposite benefits and drawbacks to stated preference data. Revealed preference data describes real world behaviour and not hypothetical choices. However, it typically only contains attributes describing the selected choice and thus contains

no details of the choice-set faced by the decision maker. For practical applications using stated preference data, average zone-to-zone travel times and costs for each transport mode are typically used to define the attributes of the choice-set.

2.2.2 Limitations of random utility approach

RUMs have a number of desirable qualities which help explain their ubiquitous usage. Five are given here:

1. RUMs are highly interpretable. By constraining the utility functions in a RUM to be linear combinations of input variables, it means it is possible to easily understand how each variable affects the choice probabilities for each mode.

2. RUMs are based on established economic behavioural theory. By investigating the signs and magnitudes of the parameters in the utility function, it is possible to determine if the predictions made by a RUM are consistent with expected behaviour.

3. RUMs are parametric models. It is therefore straightforward to conduct significance tests on RUM specifications and parameter values, e.g. to investigate if the Value of Time (VoT) of walking is different in one population compared to another.

4. MNL, NL, and CNL parameter estimates are consistent and have a convex solution. This means that gradient descent is guaranteed to find a globally optimal solution.

5. RUMs output well calibrated choice probabilities. RUM parameters are estimated using MLE. This enables well calibrated choice probabilities to be outputted (see Chapter 4).

Despite the advantages of RUMs, they have a number of limitations. The predominant limitation of the random utility approach is the complexity of manually specifying the model utility functions. As discussed above, the utility functions for each mode must be specified in advance of fitting a RUM. Any interaction which is tested in the model needs to be added manually to the utility function before estimating the parameters.

This becomes particularly problematic when interacting socio-economic covariates with the parameters in the model. Fully interacting $i = 1,..,N$ categorical covariates each with n_i classes results in $\prod_i n_i$ discrete socio-economic/trip groups s. As can be seen from Eq. (2.6), a full set of β-parameters has to be estimated for each group s. This can easily result in a very large number of parameters in the utility specifications. However, there can be issues with convergence of the parameters for highly multidimensional specifications, and there

needs to be sufficient data available for each group s to draw valid conclusions about the significance of the parameters. As such, for data with more than a few simple covariates, it is not normally possible to simply interact all covariates within the model, and instead the modeller must determine which specific interactions to test.

This makes optimising the utility functions using manual search highly complex (see Section 5.2.5.2), as the number of possible utility specifications tends quickly to infinity as the data becomes more complex. This is even more pertinent for continuous covariates in the data, as the threshold values for each discrete bin must also be specified manually for each continuous covariate.

Another limitation of RUMs is that they do not have the flexibility to model non-linear relationships in the data unless these relationships are explicitly specified in the utility function.

In practice, these limitations have typically restricted modellers to using coarse categorical covariates, and only investigating first order interactions of the covariates with choice probabilities, via the ASCs (β_0).

The limitations discussed above are inherent to statistical RUMs. The utilities in a RUM are constrained by a behavioural model. The behavioural model enhances both model interpretability and robustness (via checking for consistency of the parameter values with expected behaviour). However, it requires any relationships in the model to be explicitly specified in the utility functions, prior to estimating the parameters.

As well as the fundamental limitations of RUMs, there are a number of limitations of the current implementations of RUMs as used in practice. These limitations predominantly relate to the datasets typically used to estimate model parameters. As discussed, RUMs are often estimated using stated preference In this case, there is no guarantee that the model represents real-world behaviour.

Where revealed preference data is used, the mode-alternative attributes for the choice set are typically given as static measures of average zonal travel times and costs. Models trained using this data therefore only provide the precision to model behaviour at low spatial and temporal granularity.

Additionally, whilst the coefficients of a RUM are used to check if the model is consistent with behavioural theory, the estimated models are seldom applied to new revealed preference data. This means it is not possible to assess how accurate individual model predictions are likely to be in practice. This is particularly important for the models trained using stated-preference data.

2.3 Machine learning classifiers

As discussed in Section 1.1, the adoption of new transportation-related technologies has driven a step change in the availability of data on passenger movements of several orders of magnitude. This data presents the opportunity for a new approach for mode choice prediction, through using Machine Learning (ML) classifiers to directly infer the relationship between transport and environment conditions and passenger mode choice decisions.

As with RUMs, ML classifiers can be used to predict the probability of a trip being made by a certain mode, given a set of attributes (*features*) describing the trip. This is known as *supervised probabilistic classification*. The aim of the classifier is to accurately predict the mode actually taken (*class label*) for an unseen set of trips.

ML classifiers do not have the same behavioural constraints as RUMs, and as such require no utility functions to be defined. In addition, they have a higher degree of flexibility, and most can easily model non-linear relationships in the data. This gives them the potential to address many of the limitations associated with RUMs raised in Section 2.2.2.

As ML classifiers do not have a behavioural model, they are evaluated using only their output predictions. This requires the models to be validated (tested) on out-of-sample data (i.e. data separate from the that used to fit the model) (see Section 4.3.2). The model validation ensures that the model has successfully *generalised* to valid relationships in the data, without fitting to noise in the data. There is therefore a balance between *underfitting* and *overfitting*, known as the *bias-variance trade-off* (Hastie, Friedman, and Tibshirani 2008). Models with high bias underfit to the training data and fail to account for relevant correlations between input features and mode choice that are present in the real-world test data. Models with high variance overfit to noise in the test data and as such will introduce correlations in the model which are not present in other data samples. ML classifiers have a set of *hyper-parameters* which control how the algorithm fits to the data (see Section 4.2.5). These hyper-parameters are used to *regularise* the model and control the model's propensity to under or overfit.

There is a growing body of existing research into ML classification of mode choice. This research is presented in detail in Chapter 3. In order to provide an understanding of the techniques used, the following sections give an overview of five classes of supervised classification algorithm: Logistic Regression (LR), Artificial Neural Networks (ANNs), Decision Trees (DTs), Ensemble Learning (EL), and Support Vector Machines (SVMs). Each section gives a brief description of the algorithm and introduces the key hyper-parameters.

2.3.1 Logistic Regression

The Logistic Regression (LR) classifier is very similar to the MNL model described in Section 2.2.1.1, with linear functions of the input features passed through the softmax function to generate class probabilities. This thesis distinguishes between the two models on the basis that the LR model does not include an explicit behavioural model or utility specifications. Instead, with LR all features are included uniformly for all modes, with a single weight (equivalent to *parameters* in MNL) trained for each feature for each mode.

As the LR classifier has fewer constraints than the MNL model, it is more flexible, and as such has a higher propensity for overfitting. This can be addressed by using *regularisation*, which penalises the model during fitting on the basis of the values of the weights. *L1* regularisation (also known as *lasso* regularisation) penalises the model for the sum of absolute values of the weights. Conversely, *L2* regularisation (also known as *ridge* regularisation) penalises the model for the sum of squares of the weights. The amount of regularisation is controlled using the C hyper-parameter, with a larger value of C indicating more regularisation (higher penalty for the values of the weights).

2.3.2 Artificial Neural Networks

Artificial Neural Network (ANN) is a term used to cover a family of classifiers which mimic the network structure of the brain. Whilst there are a huge variety of possible Artificial Neural Network (ANN) structures for dealing with different input data types (e.g. images, time-series, natural language etc), this thesis focuses on the *general-purpose* algorithm: the Feed-Forward Neural Network (FFNN) (also known as the Multi-Layer Perceptron (MLP)) (Svozil, Kvasnicka, and Pospichal 1997).

A FFNN consists multiple *layers* of *nodes* (neurons), including (i) an input layer, which passes the feature values to the network; (ii) an output layer, which outputs the predicted values from the network; and (iii) any number of hidden layers. For probabilistic classification, the number of nodes in the input and output layers is fixed by the number of features and classes in the data respectively. The hidden layers can each contain any number of nodes. In a fully connected network, every node in one layer is linked to every node in the next layer.

Each node has an activation function, which determines the output of that node from the weighted sum of its inputs. There are many possible activation functions used in practice. This thesis considers nine activation functions: linear, sigmoid, hard sigmoid, tanh, softplus, softsign, ReLU (rectified linear unit), ELU (exponential linear unit), and SELU (scaled exponential linear unit).

FFNN are highly flexible in terms of relationships they can approximate. It can be shown via the universal approximation theorem (Hornik 1991) that a FFNN containing *one or more* hidden layers with a sufficient (finite) number of nodes and *any* non-linear, bounded, continuous activation function can approximate any continuous function.

As with the MNL and LR models, the output values of the network are passed through the softmax function to generate classification probabilities. Note that a LR can be thought of as a FFNN with no hidden layer and a linear activation function.

The weights (parameters) for each link in the network are fitted to the input data (equivalent to estimating a RUM). FFNNs are most commonly trained using *mini-batch gradient descent*. This algorithm splits the input data into small batches. The network weights are then updated iteratively on the individual batches. Each time the model sees all of the data once is termed an *epoch*. The number of epochs can then be controlled to limit overfitting. Further regularisation can be applied using *dropout*, where a proportion of the neurons are dropped randomly from the network for each mini-batch of data (Srivastava et al. 2014)

2.3.3 Decision Trees

Decision Trees (DTs) (or Classification and Regression Trees (CART)) are classifiers which sort data into groups using a set of sequential splits in a tree-like structure (Breiman 2017). The most commonly used Decision Trees (DTs) are fitted using recursive binary splits, with each split chosen to result in the greatest reduction in the *randomness* of the data at that point (i.e. it is a *greedy* algorithm). Two metrics can be used to measure how shuffled the data are, *Gini impurity* and *entropy*.

To calculate each split, the data at the selected node are sorted according to each feature, and each possible binary split point (less/greater than a certain value) is tested for each feature. The split point which results in the greatest reduction in the impurity or entropy (across all features) of the data is then selected, resulting in two new child nodes. The same algorithm can then be applied recursively to the child nodes. This process is repeated until a stopping condition is met.

The stopping conditions can be set using a combination of different hyper-parameters in order to prevent overfitting. For example, the *maximum depth* specifies the maximum number of sequential splits which can be applied along a branch, the *minimum leaf size* specifies the minimum size *both* nodes of a split must have in order for a split to take place, and the *minimum split size* specifies the minimum number of samples in a node for a split to be considered at that node.

Decision trees can only generate discrete predictions, and so are not suitable for probabilistic mode choice prediction when used independently. However, they can be combined in ensembles to generate probabilistic predictions (see Section 2.3.4).

As the split points in a DT represent a simple binary inequality, DTs are not sensitive to data scaling. Any monotonic transformation of a feature (where the ranks stay the same) has no impact on the model (provided the split-points are also transformed).

2.3.4 Ensemble Learning

Ensemble Learning (EL) algorithms combine several *weak learners* in an ensemble to improve the quality of predictions. Provided the weak learners make errors *independently* (i.e. the learners are uncorrelated), and are more likely to be right than wrong, then combining them in an ensemble reduces their individual uncertainty.

DTs make good candidate weak learners for Ensemble Learning (EL). DTs have high variance, making them highly unstable (small changes in the input result in large differences between classifiers). As such, it is relatively easy to train uncorrelated DTs compared to more stable classifiers (e.g. logistic regression). In addition, DTs are algorithmically simple to fit and obtain predictions from. This means that large ensembles of DTs can fit and predict in reasonable time.

Three tree-based EL algorithms are used in this thesis: Random Forest (RF) (Breiman 2001); Extremely randomised Trees (ET) (Geurts, Ernst, and Wehenkel 2006); and Gradient Boosting Decision Trees (GBDT) (Friedman 2001).

The Random Forest (RF) algorithm works by training weak learners in parallel on different *bootstrapped* samples of the data. The bootstrap samples are the same size as the original dataset, sampled with replacement from the original data. A separate DT is then trained on each sample. This process reduces the correlations between the DTs in the ensemble, therefore reducing the uncertainty of the ensemble. The votes of the DTs can then be combined to obtain a prediction for the ensemble. As well as the hyper-parameters of the DTs in the ensemble (discussed in Section 2.3.3), the number of separate DTs in the ensemble must be specified. Further regularisation can be applied by considering only a random sample of the features for each individual tree in the ensemble.

The Extremely randomised Trees (ET) algorithm is very similar to the RF algorithm. The primary difference is that each split in each DT in the ET ensemble considers a random split point for each feature (instead of identifying the split point for that feature which reduces the entropy or impurity the most). As with bootstrapping in the RF algorithm, this process reduces the correlations between the individual DTs in the ensemble (each DT is typically trained on the full dataset without bootstrapping for the ET algorithm). The rest of the

hyper-parameters correspond with those for the RF algorithm. Note that it is possible to apply the ET using bootstrapped samples for each tree, and similarly the RF algorithm can be applied without bootstrapping.

Conversely to RF and ET, the Gradient Boosting Decision Trees (GBDT) algorithm trains DTs sequentially on the full dataset, with each DT predicting the total residual error of the previous DTs. As the trees are applied sequentially, it is possible to set the ensemble size dynamically, by stopping boosting once the improvement in score does not meet a threshold within a given number of iterations. Regularisation can be introduced in a GBDT model by specifying a learning rate multiplier for each tree. The result of each tree is multiplied by the learning rate before calculating the residual error. This results in having a greater number of weaker DTs in the ensemble (when setting the size dynamically), reducing the propensity of the ensemble to overfit. Friedman (2001) finds that small values of learning rate (≤ 0.1) dramatically improve model performance.

By combining the binary splits of a large number of DT in an ensemble, EL algorithms are able to approximate arbitrary complex, non-linear relationships (in particular non-continuous relationships). This makes EL algorithms highly flexible at generalising to relationships in the data. Provided appropriate hyper-parameters are used, EL algorithms can perform well on a wide range of classification tasks. GBDT in particular have been shown to have best in class performance in several supervised classification tasks (Zhang, Liu, et al. 2017; Brown and Mues 2012; Chapelle and Chang 2011; Caruana and Niculescu-Mizil 2006).

2.3.5 Support Vector Machines

The Support Vector Machine (SVM) algorithm makes use of a *kernel* to transform the data into a high-dimension space. The algorithm then finds the optimal linear decision surface (or *hyper-plane*) in the transformed space which divides the data into two classes (Cortes and Vapnik 1995).

There are multiple kernels which can be used to transform the data. This thesis considers four: linear (no transformation), polynomial, Radial Basis Function (RBF) (or *Gaussian*), and sigmoid.

For linearly-separable data (within the transformed space), the optimal hyperplane is the one that exactly divides the data without misclassification whilst maximising the possible *margin*. The margin is defined as the perpendicular distance between the hyperplane and the nearest data points (these distances are the *support vectors*). For complex, real-world examples, the input data are not normally linearly-separable, even within the transformed space. As such, there is a balance between the width of the hyperplane and the number of misclassifications of the training data. This is controlled using the regularisation parameter

(C). A higher value of C represents a higher importance of the misclassified points (higher variance), whilst a lower value of C will put a higher importance on the width of the hyperplane (higher bias).

Support Vector Machines (SVMs) are inherently binary classifiers. However, they can be used for multiclass classification using either a *one-vs-rest* or *one-vs-one* strategy. The one-vs-rest strategy trains a single binary classifier per class, with each classifier trained to predict whether an instance belongs to a class or not. Each binary classifier is trained on all of the data. The one-vs-one classification strategy trains a binary classifier on each unique pair of classes (i.e. $J(J-1)/2$ binary classifiers for a J-class problem) and predicts which class the instance belongs to. Each classifier is trained on all samples from the data belonging to the corresponding pair of classes (i.e. a walk-vs-cycle classifier would be trained on only the walking and cycling journeys). For both strategies, the confidence scores of the respective classifiers are used to determine the predicted class for unseen data.

SVMs output a continuous score for each prediction. This score can be interpreted as the confidence of the classification. However, these scores do not correspond well to class probabilities (Niculescu-Mizil and Caruana 2005). Methods to calibrate the scores as class probabilities are proposed by Wu, Lin, and Weng (2004) and Platt (1999).

SVM complexity scales at a minimum with $O(n^2)$, where n is the number of instances (rows) in the data (Bottou and Lin 2007). This rises to $O(n^3)$ for high C (regularisation) values. This can cause computational issues and/or considerable fit-times when using SVMs with large datasets.

2.4 Random utility and machine learning differences and terminology

The work in this thesis crosses two research fields of statistical modelling (RUMs) and ML. Whilst there is a lot of overlap in the theory and practice in the two fields (see Chapters 4 and 5), there are fundamental differences between the two. These differences primarily relate to the respective motivations for the two fields.

Utility-based choice models were primarily developed in order to *describe* the behaviour of a population. As such, the primary focus of RUMs tends to be on the model structure, including the parameter values and elasticities, and there tends to be less focus on validating the model. Conversely, ML supervised classification methods were primarily developed to automate the process of predicting a class for unseen data. Within the context of mode choice prediction, a ML classifier is developed in order to *predict* the behaviour of the population.

As such, there a strong focus on model validation with ML classification, and a much lower focus on model structure.

The separate development of the two fields has resulted in a substantial number of equivalent or nearly-equivalent terms between them. Within this thesis, an attempt is made, wherever possible, to use the same theory, analysis, and methodologies for the two approaches (ML and RUMs). Therefore, for clarity of the associated material, Table 2.1 summarises some of the equivalent and nearly equivalent terms used in this thesis.

Table 2.1 Equivalent and nearly-equivalent terms between random utility and ML models.

Random utility	**Machine learning**	**Notes**
Variable	Feature	Both describe *attributes* of the trip.
Covariate	Feature	No distinction is made between variables and covariates in ML classifiers.
Parameter	Weights	Weights are used only in parametric ML models (LR and ANNs).
Estimate	Train	Both are often referred to as *fitting* the model.

2.5 Summary

This chapter presents the existing and emerging approaches for mode choice prediction. In order to understand and evaluate the ubiquitous usage of RUMs for mode choice modelling, an overview of random utility theory, model structures, and typical datasets is given. The limitations of RUMs are presented, establishing the need for a new approach to mode choice prediction. ML classifiers are proposed as an alternative to the random utility approach, with the potential to address many of the limitations associated with RUMs. Five classes of ML classification algorithms which could be used for mode choice prediction are introduced: Logistic Regression (LR), Artificial Neural Networks (ANNs), Decision Trees (DTs), Ensemble Learning (EL) (including RFs, ET, and GBDT), and Support Vector Machine (SVM). Finally, a brief overview of the differences between random utility and ML is given, as well as a summary of the equivalent and nearly equivalent terminology between them.

Whilst systematic reviews on RUMs for mode choice prediction have long existed, and the methods have been well scrutinised, the same is not at all true for ML models. To address this, Chapter 3 conducts a systematic review of ML methodologies for modelling passenger mode choice.

Chapter 3

Systematic review of machine learning methodologies for modelling passenger mode choice

3.1 Overview

This chapter presents a systematic review of Machine Learning (ML) approaches to passenger mode choice modelling. There exist several review papers in the literature focusing on mode choice modelling, including those by Barff, Mackay, and Olshavsky (1982), Hensher and Johnson (1983), Kruger (1991), Nerhagen (2000), Meixell and Norbis (2008), Ratrout, Gazder, and Al-Madani (2014), Jing et al. (2018), and Minal and Sekhar (2014). However, all but two of these reviews focus exclusively on statistical Random Utility Model (RUM) techniques. Ratrout, Gazder, and Al-Madani (2014) and Minal and Sekhar (2014) explicitly review ML and Artificial Intelligence (AI) approaches within the literature, including Artificial Neural Network (ANN) approaches to mode choice modelling alongside RUM based studies. The studies conclude that ANN have been successfully used for mode choice modelling, in particular due to their flexibility when dealing with multidimensional non-linear data. Ratrout, Gazder, and Al-Madani (2014) further state that whilst the vast majority of existing studies are based on logit models, it can be expected that the trend of using ML methods will continue in future.

Whilst the studies by Ratrout, Gazder, and Al-Madani (2014) and Minal and Sekhar (2014) evaluate some of the existing ML mode choice research, they have a number of limitations within the context of this research. Primarily, they focus only on ANN (and Fuzzy Logic (FL)) approaches, and as such do not cover any contributions using other ML techniques,

including Decision Trees (DTs), Support Vector Machines (SVMs), and Ensemble Learning (EL). Secondly, these reviews are intended to be exploratory as opposed to systematic, and do not represent comprehensive coverage of all relevant studies. Additionally, the reviews are intended to be general, and do not focus on specific aspects of the methodologies used in each study, of which a detailed understanding is required for this research. Finally, there have been a substantial number of new studies published since these reviews were carried out.

To address the limitations of the previous reviews, this chapter conducts a systematic review of ML approaches to passenger mode choice modelling. The review focuses on the methodologies employed within each study in order to (a) establish the state-of-research frameworks for ML mode choice modelling, (b) identify and quantify the prevalence of methodological limitations in previous studies, and (c) evaluate the research background to which this work contributes.

Firstly, Section 3.2 outlines the methodology for the review, including the research questions, review protocol, and study selection. Section 3.3 then presents the results of the review, first giving an overview of the selected studies, before exploring each research question in turn. The limitations identified are categorised into technical limitations, which represent technical issues within the methodologies of specific studies that are likely to have an impact on their results; and general limitations, which represent trends across multiple studies or areas that require further investigative work. Finally, Section 3.4 summarises the findings, identifies potential limitations, and presents the conclusions of the review.

This chapter identifies the technical limitations in existing research and quantifies their prevalence. The limitations are then analysed within a theoretical framework in Chapter 4, where a unified notation is used to demonstrate how each technical limitation may affect the results of the existing studies. Finally, the methodology presented in Chapter 5 specifically addresses each methodological limitation, demonstrating how they are solved within this research.

It is important to note that there are two different types of conclusions that could be drawn from the results of a comparison of classifiers for mode choice prediction. The first claim relates to the best performing classifier on a given dataset, i.e. out of a set of models tested with specific hyper-parameters, one particular model specification works best on a specific dataset. The second claim relating to the general performance of the classifiers used, i.e. that one type of classifier should be preferred over another for a general modelling task.

To make firm conclusions on the typical performance of a family of classifiers for a general task (i.e. the second claim detailed above), several benchmarks on multiple datasets are needed, e.g. by considering the results of several studies. However, in order to draw meaningful conclusions from the results of a classifier on a given dataset in a single study, it

is still important to ensure a fair comparison of the classifiers, e.g. by ensuring a thorough and unbiased approach to specifying the model hyper-parameters is used. As such, each study in the review is assessed on whether a fair comparison between the classifiers is possible, even if the paper does not attempt to investigate this type of claim directly.

To address this, each technical limitation identified in the study is further classified into *bad-practices*, i.e. incorrect modelling decisions which are likely to impact the results of an investigation, and *areas for improvement*, i.e. modelling decisions which are not incorrect but could be addressed in order to improve the reliability of the results for comparing the classifiers and/or the predictive performance of the models.

3.2 Method

The procedure for this systematic review is adapted from that given by Kitchenham and Charters (2007). The suggested procedure suggested has 10 stages broken down into three phases:

- *Planning the review*
    1. Identification of the need for a review
    2. Specifying the research questions
    3. Developing a review protocol
- *Conducting the review*
    4. Identification of research
    5. Selection of primary studies
    6. Study quality assessment
    7. Data extraction and monitoring
    8. Data synthesis
- *Reporting the review*
    9. Specifying dissemination mechanisms
    10. Formatting the main report

This review is focused on summarising the methodologies used in each study. The implications of any limitations and assumptions highlighted by the review are discussed in detail in Chapter 4 and investigated experimentally throughout the thesis.

The meta-analysis in this review focuses on the methodologies used in each study, and as such, no attempt is made to draw conclusions from the aggregate results or combined findings of the studies. Consequently, no assessment of the quality of each study is made (step 6 in the framework). The review presented in this thesis therefore consists of the nine remaining stages presented above.

3.2.1 Research questions

The focus of this review is the methodologies used in ML approaches to modelling passenger mode choice. In particular, the review serves to investigate the following research questions:

1. Which classification techniques have been used to investigate mode choice?
2. What is the nature of datasets used to investigate mode choice?
3. How is model performance determined?
4. How are optimal model hyper-parameters selected?
5. How is the best model selected?

3.2.2 Review protocol

This section outlines the protocol for the search strategy, selection criteria, and data extraction strategy. This review protocol was developed in discussion with the PhD supervisors and fellow PhD students.

3.2.2.1 Search strategy

The search strategy is used to identify relevant papers to the review. In order to ensure full coverage of relevant papers, papers are collated from three databases: the two major online curated publication databases, Web of Science and Scopus; and the Google Scholar search engine. The same search is repeated for each database.

This review focuses on papers with a core focus of mode choice modelling. As such, only papers with the *title* directly relating to mode choice are included. The following initial phrases are tested: *mode choice*, *mode selection*, *travel mode*, *transport mode*, *transportation mode*, and *mode of travel*.

In order to only select papers that discuss ML techniques, only papers with one or more selected phrases relating to ML across all relevant fields are selected. The following initial phrases are tested: *machine learning*, *neural network*, *decision tree*, *ensemble method*, *random forest*, *boosting*, and *support vector*.

Papers from any period up until the search date are included in the search.

The initial search phrases are tested in different combinations across the three databases. The terms *mode of travel* and *mode selection* are emitted from the title search, as they return no relevant papers when used alongside the ML search terms.

Additionally, a number of papers using Fuzzy Logic (FL) (within Rule-Based Machine Learning (RBML)) were found in the initial search results. To reflect this, the phrase *fuzzy logic* is added to the search across all relevant fields.

3.2.2.2 Selection criteria

The following eligibility criteria are determined for the papers found in the search to be included in the study:

- Studies in peer-reviewed journals or conference proceedings written in English
- Studies which investigate passenger mode choice at disaggregate (individual) level.
- Studies which employ one or more ML technique(s) for predictive modelling.

Paper selection is carried out using the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines (Moher et al. 2009). Firstly, duplicates are removed from the search records. Secondly, the record titles and abstracts are screened against the eligibility criteria. Finally, the remaining full-text articles are assessed for eligibility. All stages of the selection criteria are carried out independently by the author.

Where a paper contains more than one relevant modelling scenario (defined as having separate input datasets and different methodologies), each modelling scenario is treated as a separate study for the analysis.

Published work by the author which form part of this thesis are omitted from the review (Hillel, Elshafie, and Jin 2018).

3.2.2.3 Data extraction strategy

In order to extract the necessary data from each study without bias, a list of attributes is collected from each study. The attributes, shown in Table 3.1, are intended to be specific, objective, and quantifiable/categorical, in order to limit subjectivity in the data extraction process. Together the attributes provide the evidence for the research questions presented in Section 3.2.1.

Table 3.1 Research questions and corresponding attributes of studies for data extraction.

No.	Description
Q1	**Which classification techniques have been used to investigate mode choice?**
Q1a	Classification algorithms used in study
Q1b	Logit model implementation
Q2	**What is the nature of datasets used to investigate mode choice?**
Q2a	Nature of dataset
Q2b	Unit of analysis
Q2c	Dataset availability
Q2d	Modes in choice-set
Q2e	Modelling of mode-alternatives
Q2f	Input features dependent on output choice
Q2g	Hierarchical data
Q3	**How is model performance determined?**
Q3a	Validation method
Q3b	Sampling method
Q3c	Performance metrics used
Q4	**How are optimal model hyper-parameters selected?**
Q4a	Hyper-parameter search method
Q4b	Hyper-parameter validation method
Q4c	Hyper-parameter validation data
Q5	**How is the best model selected?**
Q5a	Model selection technique

Data extraction is carried out independently by the author. Each paper is reviewed in detail, with each attribute for each study determined and tabulated in a spreadsheet. Separate entries are entered into the spreadsheet for papers containing multiple studies (modelling scenarios).

3.2.3 Study selection

The following search terms are used to carry out the search strategy outlined in Section 3.2.2.

- **Web of Science**: *TITLE: ("mode choice" OR "travel mode" OR "transport mode" OR "transportation mode") AND TOPIC: ("machine learning" OR "neural network" OR "decision tree" OR "ensemble method" OR "random forest" OR "boosting" OR "support vector" OR "fuzzy logic")*

- **Scopus**: *(TITLE ("mode choice" OR travel mode" OR "transport mode" OR "transportation mode") AND TITLE-ABS-KEY ("machine learning" OR "neural network" OR "decision tree" OR "ensemble method" OR "random forest" OR "boosting" OR "support vector" OR "fuzzy logic"))*

- **Google Scholar**: *(intitle:"mode choice" OR intitle:"travel mode" OR intitle:"transport mode" OR intitle:"transportation mode") AND ("machine learning" OR "neural network" OR "decision tree" OR "ensemble method" OR "random forest" OR "boosting" OR "support vector" OR "fuzzy logic").*
  Due to the restriction on search length in Google Scholar, this search is divided into two separate searches, with the results combined.

The search was carried out on 25/05/2018 on all three databases. Figure 3.1 shows a PRISMA flowchart of the study selection process.

There were 78 records returned from the Web of Science search, 220 records from Scopus, and 442 records from Google Scholar, for a total of 740 records. Duplicates are then removed, leaving 468 records to be screened. The total number of records after removing duplicates is more than were obtained from any one database, showing that there were results from Web of Science/Scopus which were not returned with the Google Scholar search.

The 468 remaining records are then screened as to whether they meet the eligibility criteria outlined in Section 3.2.2. During screening, 327 papers are excluded for relevance on the basis of their title and abstract. The majority of these records relate to transportation mode detection from Global Positioning System (GPS) data. Of the records which are deemed relevant, a further 45 are excluded as they are not published in peer reviewed publications, (e.g. Thesis/dissertation, unpublished paper, book section), or are not written in English (only having a title and abstract in English).

The full text is obtained for the remaining 96 articles for further review. Of these, a further 36 are excluded on the basis of the selection criteria, as detailed in Fig. 3.1. This leaves 60 selected articles for data-extraction.

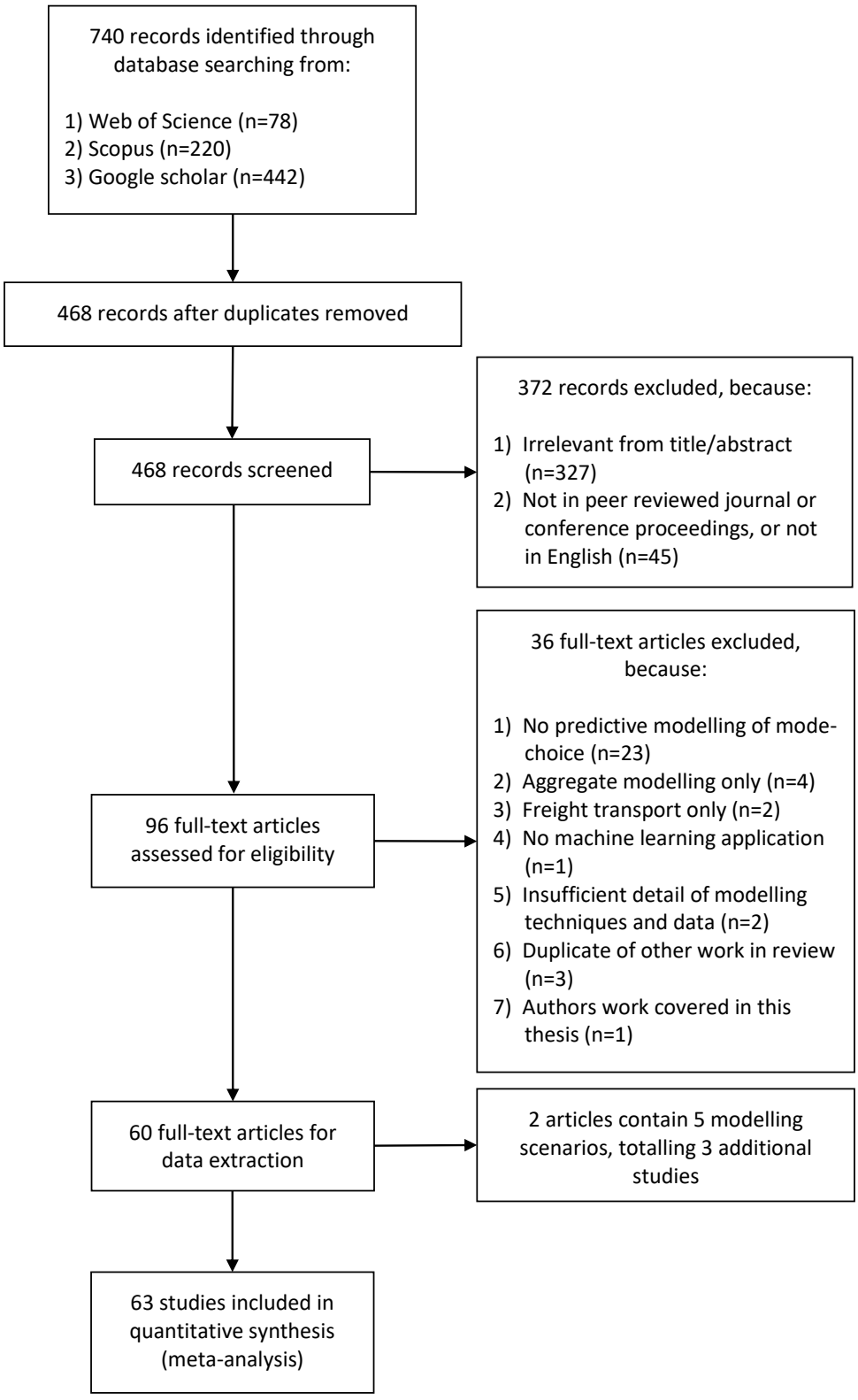


Fig. 3.1 PRISMA flowchart of study selection process.

Two articles contain multiple modelling scenarios, for a total of 63 separate studies for meta-analysis.

3.3 Results and discussion

This section presents the results obtained from the systematic review process. Firstly, Section 3.3.1 provides an overview of the 60 articles used for data extraction, including the publication sources and years. The articles with multiple studies are identified, and each of the 63 studies are given a unique identifier. Sections 3.3.2 to 3.3.5 then use evidence from the 63 studies to explore each of the five research questions in turn.

3.3.1 Articles for data extraction

This section provides an overview of the 60 articles used for data extraction. Table 3.2 provides a unique identifier for each article, alongside its individual reference.

Two papers [S5; S21] contain multiple modelling scenarios, using separate datasets and a different methodology for each one. A separate identifier is assigned to each modelling scenario in each of these papers, and they are treated as separate studies for the meta-analysis. Table 3.3 provides the label for the additional studies, alongside a description of each modelling scenario. The two papers have a total of five modelling scenarios. This results in a total of 63 studies for meta-analysis.

Four further papers have multiple modelling phases but are deemed not to be separate studies for the purpose of this review.

S6 includes input datasets for two cities: Visakhapatnam and Nagpur. The datasets are collected as part of the same study and the modelling methodology used for each is identical. As such, both are treated as the same study. The dataset for Visakhapatnam is analysed in Section 3.3.3, with the only difference between the two for the purpose of the review being that the Nagpur city dataset has 27 fewer records (1045 vs 1018).

S16 also contains multiple datasets, one for Sydney and one for Melbourne. However, both datasets are collected using the same methodology, and are used in combination in the same model the paper. The combined dataset is used for the analysis in Section 3.3.3.

S13 includes three separate modelling phases, which all model slightly different choice situations. However, each phase uses different subsets of the same dataset, and all use the same methodology. *Phase I*, which models the revealed preference choice between car and plane, is used for the analysis in the review.

Table 3.2 Selected primary articles for review.

No.	Paper	No.	Paper
S1	Andrade, Uchida, and Kagaya (2006)	S31	Moons, Wets, and Aerts (2007)
S2	Assi et al. (2018)	S32	Nam et al. (2017)
S3	Biagioni et al. (2009)	S33	Omrani (2015)
S4	Cantarella and de Luca (2003)	S34	Omrani et al. (2013)
S5	Cantarella and de Luca (2005)	S35	Papaioannou and Martinez (2015)
S6	Chalumuri et al. (2009)	S36	Pirra and Diana (2017)
S7	Cheng et al. (2014)	S37	Pitombo et al. (2015)
S8	Dell'Orco and Ottomanelli (2012)	S38	Pulugurta, Arun, and Errampalli (2013)
S9	Edara, Teodorović, and Baik (2007)	S39	Raju, Sikdar, and Dhingra (1996)
S10	Ermagun, Rashidi, and Lari (2015)	S40	Ramanuj and Gundaliya (2013)
S11	Errampalli, Okushima, and Akiyama (2007)	S41	Rasouli and Timmermans (2014)
S12	Gao et al. (2013)	S42	Seetharaman et al. (2009)
S13	Gazder and Ratrout (2015)	S43	Sekhar, Minal, and Madhu (2016)
S14	Golshani et al. (2018)	S44	Semanjski, Lopez, and Gautama (2016)
S15	Hagenauer and Helbich (2017)	S45	Shafahi and Nazari (2006)
S16	Hensher and Ton (2000)	S46	Shukla et al. (2013)
S17	Hossein Rashidi and Hasegawa (2014)	S47	Subba Rao et al. (1998)
S18	Hussain et al. (2017)	S48	Tang, Yang, and Zhang (2012)
S19	Jia, Cao, and Yang (2015)	S49	Tang, Xiong, and Zhang (2015)
S20	Juremalani (2017)	S50	Van Middelkoop, Borgers, and Timmermans (2003)
S21	Karlaftis (2004)	S51	Wang and Ross (2018)
S22	Kedia, Saw, and Katti (2015)	S52	Wang and Namgung (2007)
S23	Kumar, Sarkar, and Madhu (2013)	S53	Xian-Yu (2011)
S24	Lee, Derrible, and Pereira (2018)	S54	Xie, Lu, and Parkany (2003)
S25	Li et al. (2016)	S55	Yin and Guan (2011)
S26	Liang et al. (2018)	S56	Zenina and Borisov (2011)
S27	Lindner, Pitombo, and Cunha (2017)	S57	Zhang and Xie (2008)
S28	Lu and Kawamura (2010)	S58	Zhao et al. (2010)
S29	Ma (2015)	S59	Zhou and Lu (2011)
S30	Ma, Chow, and Xu (2017)	S60	Zhu et al. (2017)

Finally, S17 also involves three modelling stages which use different subsets of the same dataset: *Model 1* predicts total number of trips in a day, *Model 2-1* predicts attributes of the first trip in a day made by an individual, and *Model 2-2* predicts attributes of subsequent trips made by using attributes of the previous trip. As *Model 2-2* uses details of the previous trip taken from the dataset (and not as predicted by a model), it is not relevant as a predictive model within the scope of this study. As such only *Model 2-1* is analysed within this review.

3.3.1.1 Publication source

Table 3.4 provides details of all journals and conferences/proceedings from which more than one article was selected. The articles come from a wide spread of publications, with a total of 28 different journals and 16 different conferences featured. The majority of the

Table 3.3 Primary studies with multiple modelling scenarios in review.

No.	Paper	No.	Scenario
S5	Cantarella and de Luca (2005)	S5.1	VENETO dataset
		S5.2	UNISA dataset
S21	Karlaftis (2004)	S21.1	Interurban mode choice in Australia
		S21.2	Commuter mode choice in Athens, Greece
		S21.3	Commuter mode choice in Las Condes-CBD corridor, Chile

papers (39/60) are published in journals, making up 65 % of the articles, with the remaining 21 papers (35 %) published in conference proceedings.

Table 3.4 Summary of publication sources contributing more than one paper to review. Multi-conference proceedings are shown in bold, with the individual conferences in italics below.

Publication	Type	No.
Transportation Research Record	Journal	7
Transportation Research Board Annual Meeting	Conference	6
Transportation Research Procedia:	**Proceedings**	**4**
Euro Working Group on Transportation	*Conference*	*3*
Transportation Planning and Implementation Methodologies for Developing Countries	*Conference*	*1*
Travel Behaviour and Society	Journal	2
East Asia Society for Transportation Studies	Conference	2
International Conference of Chinese Transportation Professionals	Conference	2
Totals (all papers)	**Journal**	39
	Conference	21

The top two sources for articles are the Transportation Research Record Journal and the Transportation Research Board Annual Meeting conference, both of which are published by the Transportation Research Board. Together, they make up 22 % (13/60) of the articles.

3.3.1.2 Publication year

Figure 3.2 shows the distribution of article publication dates from 1995 to 2018.

There is a clear upwards trend of increasing number of publications regarding ML applications to mode choice per year. Over half of the selected articles (31/60) were published from 2012 onwards. Conversely, only 10 relevant papers were published prior to 2007. Data

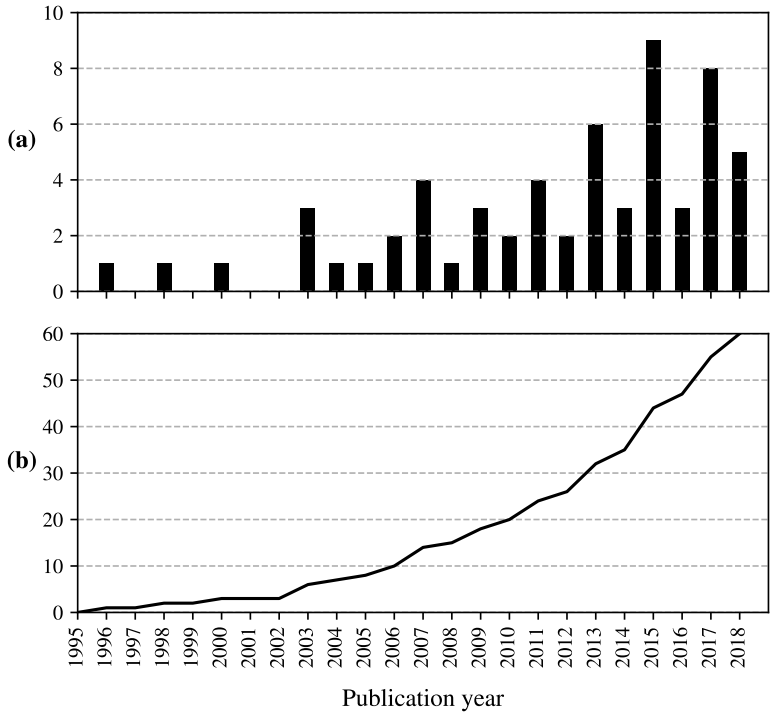


Fig. 3.2 Publication distribution of articles in systematic review (a) per year and (b) cumulative.

for 2018 is incomplete as the search was carried out in May 2018, but there are still a considerable number of articles published in this year (5/60).

3.3.2 Which classification techniques have been used to investigate mode choice?

The following sections present an overview of the classification techniques used in the 63 studies in the review.

Q1a: Classification algorithms used in study

Based on the responses to Q1a, the classification techniques are grouped into nine categories, as shown in Table 3.5. An overview of the classification techniques used in this thesis is given in Sections 2.2.1 and 2.3. For each algorithm, an example paper from the systematic review which makes use of that algorithm is provided.

Table 3.6 shows which classification techniques are used in each study. The majority of studies (40/63) compare ML techniques with statistical RUMs and Logistic Regression (LR),

Table 3.5 Classification techniques used in studies in review.

Classification algorithm	**Example reference**
1. Logit models (Log)	
Multinomial Logit (MNL)	Cantarella and de Luca (2005)
Nested Logit (NL)	Hensher and Ton (2000)
Cross-Nested Logit (CNL)	Nam et al. (2017)
Binary Logit	Lindner, Pitombo, and Cunha (2017)
2. Artificial Neural Networks (ANNs)	
Feed-Forward Neural Network (FFNN)	Lee, Derrible, and Pereira (2018)
Radial Basis Function Neural Network (RBFNN)	Omrani (2015)
Probabilistic Neural Network (PNN)	Zhou and Lu (2011)
Other neural network structures	Cantarella and de Luca (2003)
3. Decision Trees (DTs)	Karlaftis (2004)
4. Ensemble Learning (EL)	
Random Forests (RFs)	Hossein Rashidi and Hasegawa (2014)
Gradient Boosting (GB)	Wang and Ross (2018)
AdaBoost (AB)	Biagioni et al. (2009)
Bagging	Hagenauer and Helbich (2017)
5. Support Vector Machines (SVMs)	Xian-Yu (2011)
6. Bayesian Learners (BLs)	
Naïve Bayes (NB)	Hagenauer and Helbich (2017)
Bayesian Network (BN)	Ma (2015)
Tree Augmented Naïve Bayes	Tang, Yang, and Zhang (2012)
7. Rule-Based Machine Learning (RBML)	
Fuzzy Inference System	Dell'Orco and Ottomanelli (2012)
Rough Sets Model	Cheng et al. (2014)
Class Association Rules	Lu and Kawamura (2010)
8. Hybrid methods (HM)	
Clustered Logistic Regression	Li et al. (2016)
Boosted logit	Biagioni et al. (2009)
Logit-ANN	Gazder and Ratrout (2015)
9. Miscellaneous (Msc)	
Multivariable Fractional Polynomials	Nam et al. (2017)
Discriminant Analysis	Karlaftis (2004)
Structural Equation Modelling	Papaioannou and Martinez (2015)
Linear regression	Ramanuj and Gundaliya (2013)

making logit models the most commonly used classification technique in the studies. The most commonly used ML algorithms are ANNs (30 studies), followed by DTs (16 studies), and RBML (11 studies). The remaining classes of algorithms have been used in 10 or less studies each.

Table 3.6 ML techniques used in each study in review.

No.	Log	ANN	DT	EL	SVM	BL	RBML	HM	Msc	No.	Log	ANN	DT	EL	SVM	BL	RBML	HM	Msc
S1	✓						✓			**S30**	✓					✓			
S2	✓	✓								**S31**	✓		✓		✓				✓
S3	✓		✓	✓	✓	✓				**S32**	✓	✓							
S4	✓	✓								**S33**	✓	✓			✓				
S5.1	✓	✓								**S34**	✓	✓	✓		✓	✓			✓
S5.2	✓	✓								**S35**									✓
S6	✓	✓								**S36**					✓				
S7	✓						✓			**S37**	✓		✓						
S8	✓						✓			**S38**	✓						✓		
S9		✓								**S39**		✓							
S10	✓			✓						**S40**		✓							✓
S11	✓						✓			**S41**				✓					
S12		✓								**S42**	✓						✓		
S13	✓	✓						✓		**S43**			✓	✓					
S14	✓	✓								**S44**					✓				
S15	✓	✓		✓	✓	✓				**S45**							✓		
S16	✓	✓								**S46**		✓	✓						
S17			✓	✓						**S47**	✓	✓							
S18	✓	✓								**S48**		✓				✓			
S19		✓			✓					**S49**	✓		✓						
S20	✓			✓	✓					**S50**			✓						
S21.1			✓							**S51**	✓			✓					
S21.2	✓	✓	✓						✓	**S52**							✓		
S21.3			✓							**S53**	✓	✓							
S22							✓			**S54**	✓	✓	✓						
S23							✓			**S55**		✓							
S24	✓	✓								**S56**	✓		✓						✓
S25	✓							✓		**S57**	✓	✓			✓				
S26	✓			✓						**S58**		✓							
S27	✓	✓	✓							**S59**	✓	✓							
S28							✓			**S60**	✓		✓			✓			
S29	✓					✓				**Sum**	40	30	16	9	10	7	11	2	6

Q1b: Logit model implementation

Whilst the overall focus of this review is the ML methodologies used in the studies, Q1a identifies 40 studies which compare ML approaches with logit models (statistical RUMs and LR). As such, this section gives a brief overview of the logit models used in these studies.

As discussed in Section 2.2.2, a distinction is made between RUMs, which use logistic regression with explicit utility functions for each mode within the random utility framework; and LR classification, where all input features are included uniformly for all classes in the model, and no utility functions or behavioural assumptions are specified. The distinction is not necessarily clear as to which approach is used in each study, due to the overlapping terms used to describe them. Many studies describe LR classification as Discrete Choice

Models (DCMs) or RUMs, and both approaches are frequently referred to as Multinomial Logit (MNL) models.

For the purpose of this review, a model is deemed to be a RUM only if it uses different utility specifications for the different modes in the model. This can be through including the relevant Level of Service (LOS) variables, e.g. expected journey duration, for each mode (see Q2e); by determining and removing irrelevant features through significance testing and behavioural constraints (e.g. correct sign of parameters); through testing multiple utility specifications to identify one which fits the data best; or a combination of these approaches. Any model where all features are included uniformly across all modes is deemed to be a LR classifier.

Of the 40 studies which use logit models, 19 make use of utility-based RUMs. This includes binary logit [S6; S42], MNL [S1; S3; S4; S5.1; S5.2; S14; S24; S25; S28; S37; S47; S51; S53; S57], Nested Logit (NL) [S5.1; S10; S16; S32; S53], and Cross-Nested Logit (CNL) models [S5.2; S32].

S25 additionally makes use of a clustered logit structure, where a decision tree is used to segment the population into three clusters on the basis of their socio-economic variables, and separate MNL models are trained for each cluster on the remaining variables. None of the other studies model input feature interactions - e.g. effect of car ownership on Value of Time (VoT).

Nine studies provide no details of the model structure [S8; S11; S13; S20; S21.2 S34; S49; S56; S59], so that it cannot be determined which approach is used.

The remaining 12 studies [S2; S7; S15; S18; S26; S27; S30; S29; S31; S33; S54; S60] use LR classification, and include all input variables uniformly for each mode, with no LOS (alternative-specific) variables.

3.3.2.1 Techniques - Limitations

Three general limitations are identified regarding the ML techniques used to investigate mode choice: (i) the limited number of studies which systematically compare several classification algorithms on the same task; (ii) the relatively low number of investigations into EL algorithms, in particular Gradient Boosting Decision Trees (GBDT); and (iii) the inconsistent representation of RUMs in ML studies.

There are very few studies which systematically compare the performance of a range of classification techniques on the same task. Furthermore, the extensive differences in the datasets and methodologies used in each study make it impossible to make meaningful comparisons of model performances across individual studies. Table 3.6 shows that the vast majority of studies (52/63) use only one or two types of classification algorithm. Of these, 17

studies use only one type of classification algorithm. In total, only three studies [S3; S15; S34] have attempted to compare more than five types of classification algorithm. All of these studies suffer from various methodological limitations discussed in this review, as shown in Table 3.12. As such, there is a need for a comparative study of classification techniques for mode choice prediction, using a rigorous methodology.

Q1a shows that EL algorithms are used in nine out of the 63 studies. In particular, GBDT models are only used in three studies [S15; S20; S51], despite GBDT models consistently showing best in class performance in a number of similar tasks (see Section 2.3.4). All of the studies which investigate GBDT show various methodological limitations (see Table 3.12). As such, further investigation into the suitability of GBDT and other EL techniques is required.

Finally, Q1b highlights the inconsistent representation of RUMs in the studies in the review. Twenty-one studies either use uniform LR classification, or do not provide any information on the logit model structure. In the majority of these studies, these models are stated as representing RUMs when compared with other ML classification algorithms. The distinction is important as the utility function allows the modeller to add structural information based on solid behavioural foundations to the model. This structure assists the model in generalising to relationships between the input variables and mode choice, and can help prevent overfitting to training data, thereby improving model performance. Only one of the 40 studies which make use of logit models includes any interaction of input features. This is despite this frequently having a significant impact on modelling results and being common practice in RUM applications.

3.3.3 What is the nature of datasets used to investigate mode choice?

The following sections discuss the datasets used in the 63 studies in the review, focusing in turn on the nature of the dataset (trip diary/single-trip questionnaire/stated preference survey, etc); the unit of analysis (trip/tour/commute pattern/mobility); the size of the dataset; the dataset availability; the modes in the choice-set; the modelling of mode-alternatives; input features dependent on output choice; and hierarchical data.

Q2a: Nature of dataset

Table 3.7 shows the description and size of each dataset.

Only three studies [S1; S16; S32] use stated preference data. One study [S42] uses synthetic choice data, where the choice for a hypothetical metro service is synthesised based

Table 3.7 Nature and size of dataset used in each study in review.

No.	Type	N
S1	Stated preference - individual panel survey (160 people, 6 trips per person)	960
S2	Individual single-trip questionnaire (education commute)	597
S3	Trip diaries from household survey (1-2 day, 116666 trips, 19118 tours)	116666
S4	Individual single-trip questionnaire (mixed purpose urban)	2350
S5.1	Individual single-trip questionnaire (student extra-urban trips)	1116
S5.2	Individual single-trip questionnaire (mixed purpose urban)	2350
S6	Unclear survey	1045
S7	Trip diaries from household survey (5721 outbound trips only, 4831 individuals, 1809 households)	5721
S8	Individual single-trip questionnaire (outbound home-work trip)	361
S9	Trip diaries from household survey (1 year, >100 mile, business trips only)	118000
S10	Individual single-trip questionnaire (outbound home-school trip)	4700
S11	Unclear individual questionnaire	2868
S12	Trip diaries from unclear survey (650 trips sampled from larger survey, 130 for each mode)	650
S13	Individual single-trip questionnaire (cross-border)	516
S14	Trip diaries from household survey (1-2 day, outbound shopping trips only)	9450
S15	Trip diaries from household survey (6 day, 230608 trips, 69918 individuals)	230608
S16	Stated preference - individual panel survey (3 trips per person)	801
S17	Trip diaries from household survey (1 day, only first trip, 24807 individuals, 12568 households)	24807
S18	Individual single-trip questionnaire (mixed purpose urban)	620
S19	Unclear trip survey (4500 trips sampled from 17539)	4500
S20	Individual single-trip questionnaire (work commute)	224
S21.1	Individual single-trip questionnaire (mixed-purpose)	210
S21.2	Individual single-trip questionnaire (mixed purpose urban)	7100
S21.3	Individual single-trip questionnaire (work commute)	617
S22	Trip diaries from household survey (education trips only)	409
S23	Individual single-trip questionnaire (work commute)	606
S24	Trip diaries from household survey (home-based trips, sampled to over-represent transit)	4764
S25	Individual single-trip questionnaire (Holiday travel)	731
S26	Trip diaries from household survey (mode choice analysed at household mobility level)	101053
S27	Trip diaries from household survey (mode choice analysed at household mobility level)	18733
S28	Trip diaries from household survey (1-day, morning-peak home-work trips only)	9210
S29	Trip diaries from individual survey (1-day, 11993 trips made by 7235 people)	11993
S30	Trip diaries from individual survey (1-day, commute patterns extracted)	5040
S31	Activity diaries from individual survey (commute patterns extracted)	1025
S32	Stated preference - individual panel survey	6768
S33	Commute patterns in household economic survey (9500 individuals, 3670 households)	3670
S34	Commute patterns in household economic survey (9500 individuals, 3670 households)	3673
S35	Trip diaries from individual survey (530 trips, <382 individuals)	530
S36	Trip diaries from household survey (Grouped into tours, 39167 home-based tours, 24396 individuals)	39167
S37	Unclear household survey (1 day, mobility of household head only)	1216
S38	Trip diaries from household survey (Unknown trips, 5822 individuals, 2627 households)	?
S39	Unclear household survey (work trips only, 535 trips sampled randomly from 3500)	535
S40	Unclear household survey	1348
S41	Trip diaries from household survey (1 day, unknown trips, 1446 individuals)	?
S42	Individual single-trip questionnaire (synthetic choice)	229
S43	Unclear survey	5843
S44	Trip diaries from individual GPS survey (4 months, 17040 trips, 292 individuals)	17040
S45	Unclear household survey	4147
S46	Trip diaries from household survey (1-day)	100000
S47	Individual single-trip questionnaire (access to rail on work trip)	4335
S48	Unclear daily travel survey (2000 trips sampled from larger survey, 500 for each mode)	2000
S49	Trip diaries from household survey (2-day, 72536 trips, 31000 individuals, 14000 households)	72536
S50	Activity diaries from individual survey (1 year, >2-day vacations only, 7121 vacations, 2791 individuals)	7121
S51	Trip diaries from household survey (1 day)	51910
S52	Individual single-trip questionnaire (fixed O-D, mixed-purpose)	366
S53	Travel diaries from unknown survey (work travel mode choice)	4725
S54	Trip diaries from household survey (2-day, 4746 outbound work trips)	4746
S55	Unclear survey	1007
S56	Individual single-trip questionnaire (mixed-purpose)	498
S57	Trip diaries from unknown survey (outbound work trip only)	5029
S58	Individual single-trip questionnaire	100
S59	Trip diaries from unclear survey (500 trips sampled from larger survey, 125 for each mode)	500
S60	Trip diaries from household survey (1-day, home-based social activity)	5213

on a proposed fare structure and the respondent's willingness-to-pay (which is recorded during the interview).

The remaining 59 studies use revealed preference data, which can be largely grouped into two categories: trip diaries, or single-trip questionnaires.

Thirty-one studies make use of trip diary or activity-diary data, over periods ranging from one day to one year. These diaries are collected either from household surveys [S3; S7; S9; S14; S15; S17; S22; S24; S26; S27; S28; S36; S37; S38; S41; S46, S49; S51; S54; S60] or individual surveys [S29; S30; S31; S35; S44; S50]. Five studies which use trip diary data do not specify enough detail to determine if an individual or household survey is used [S12; S48; S53; S57; S59]. One study [S44] uses GPS tracking to log trips automatically, the rest use manually reported trip diaries.

In many studies, a subset of trips is selected from complete trip diaries, e.g. work trips only [S28; S30; S31; S39; S53; S54; S57], education trips only [S22], shopping/social trips only [S14; S60], outbound trips only [S7], trips from home only [S24], first trip of the day only [S17], or random sampling [S12; S48; S59].

Eighteen studies use individual single-trip questionnaires, where an individual is asked about a single trip they have made [S4; S5.1; S5.2; S8; S10; S13; S18; S21.1; S21.2; S21.3; S25; S47; S52; S56, S58] or a commute they make regularly [S2; S20; S23].

Two studies [S33; S34] make use of a household survey, in which each working member of the household details their work commute.

Eight studies [S6; S11; S19; S39; S40; S43; S45; S55] do not describe the data in enough detail to be able to determine the nature of the dataset.

The size of each dataset is also shown in Table 3.7. Twenty studies use small datasets, with between 100-1000 entries. Twenty-nine studies use medium datasets, with between 1000-10 000 entries. Seven studies use large datasets, with between 10 000-100 000 entries. Five studies use datasets larger than 100 000 entries.

Two studies [S38, S41] do not give the exact size of the dataset. They both use the trip diaries of individuals in a household survey (5822 individuals in S38, 1446 individuals in S41).

Q2b: Unit of analysis

Fifty-seven of the studies use a single independent choice as the unit of analysis. The choice can be for a single one-way trip per respondent, a return trip (by assuming each leg is made by the same mode), trip diary data where sequences of trips are treated as independent, a regular commute, or a stated preference. Fifty-five of these studies model the mode choice only, whilst two studies [S17; S60] jointly consider other trip attributes (see Q2e).

Six studies use a different unit of analysis. Four studies analyse *mobility*. S26 and S27 both analyse household mobility by predicting the predominant mode used by a household across all trips made on the survey day. S37 analyses individual mobility, by predicting the predominant mode used by an individual across all trips they make on the survey day. Finally, S9 analyses the mobility within clusters. Clusters of similar trips are generated using *k*-means clustering (Hartigan and Wong 1979). The proportions of trips made by each mode within these clusters is then predicted.

Two studies use a tour-based approach. S3 uses the predicted mode choice of the first trip in a tour (the *anchor mode*) as an input feature for subsequent trips. S36 groups trips into home-based tours across eight categories and predicts overall mode choice for each tour (including mixed mode tours).

Note that (as discussed in Section 3.3.1) S17 implements a tour-based analysis, but the subsequent trips in a tour are predicted on the basis of the attributes of the previous trip (including mode choice) as recorded in the dataset, and not as predicted by a model. As such, only the model which predicts attributes of the first trip of the day is analysed in the review (*Model 2-1* in the paper).

Q2c: Dataset availability

An attempt was made to identify and check the availability of the dataset used in each study. The following section discusses all datasets which were found to be openly available. Note that some studies which make use of open data may not have been identified, due to resource constraints when searching for datasets (see Section 3.4).

Eighteen studies are identified as using open or partially open data. The majority use open household travel survey data. Two studies make use of academic datasets made public by the authors: S21.1 makes use of the CLOGIT dataset, available with the Ecdat R library (Croissant 2016; Greene 2011); and S32 uses the SwissMetro dataset (Bierlaire, Axhausen, and Abay 2001; Bierlaire 2018). Four studies [S29; S30; S33; S34] make use of the partially open LISER PSELL data, which is available on registration (Luxembourg Institute of Socio-Economic Research 2018). Eleven studies use openly available household travel surveys:

- CMAP Travel Tracker Survey, 2007-2008 (Chicago Metropolitan Agency for Planning 2018b) - 3 studies [S3; S14; S24]
- CATS Household Travel Survey, 1990 (Chicago Metropolitan Agency for Planning 2018a) - [S28]
- San Francisco Bay Area Travel Survey, 2000 (Metropolitan Transportation Commission 2018b) - [S54]

- San Francisco Bay Area Travel Survey, 1990 (Metropolitan Transportation Commission 2018a) - [S57]
- Delaware Valley Household Travel Survey, 2012 (Delaware Valley Regional Planning Commission 2018) - [S51]
- National Household Travel Survey, 2009 (Federal Highway Administration 2018) - [S36]
- American Travel Survey, 1995 (Bureau of Transportation Statistics 2018) - [S9]
- Sydney Household Travel Survey (Transport for NSW 2018) - [S46]
- Victorian Integrated Survey of Travel and Activity, 2007-2008 (Transport for Victoria 2018) - [S17]

Only one study [S15] is identified as making the fully processed data openly available, in the format used for modelling within the paper.

Q2d: Modes in choice-set

Figure 3.3 shows a frequency plot of the number of modes considered in each study, which ranges from two to nine. The most common number of modes considered is four, which is used in 18/63 studies.

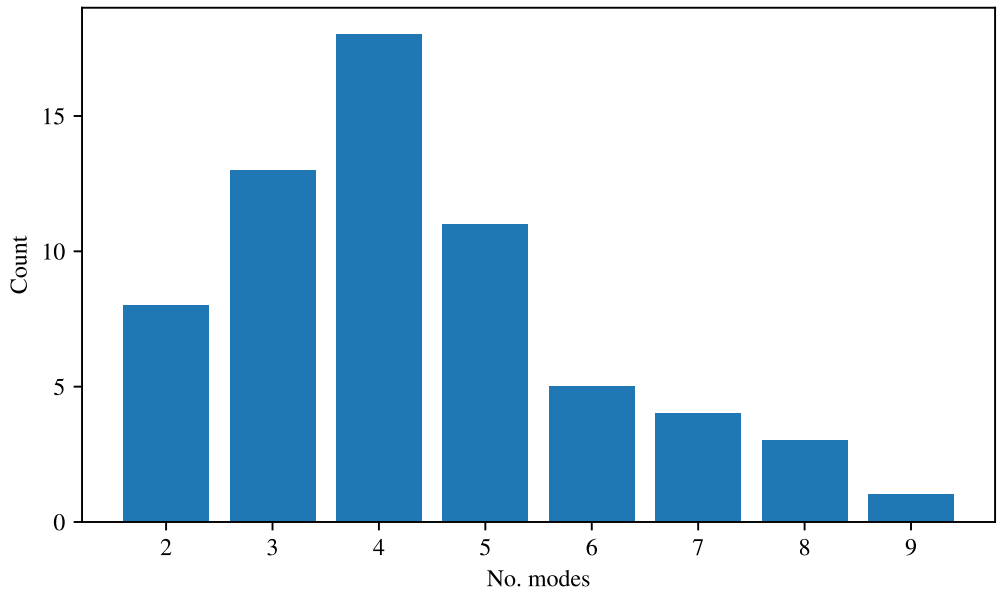


Fig. 3.3 Frequency bar chart of number of modes considered in each study in review.

Five papers have a different number of classes modelled in the classification problem from the number of modes considered. Three papers perform only one-vs-one or one-vs-rest

modelling. S9 and S31 both consider three modes, but in both studies the modelling is performed one-vs-rest across the three modes, so that each model considers two different classes. Unlike other studies which use one-vs-rest modelling, the individual models are not combined to create a multiclass classifier in either study. Similarly, S49 considers four modes, but the modelling is performed one-vs-one. As with S9 and S31, the individual models are not combined to create a single multiclass classifier.

Two models jointly model other variables alongside mode choice. S60 jointly considers four modes across two different time-periods (peak/off-peak), therefore modelling a total of eight classes. Similarly, S17 jointly models three modes, three trip purposes, three departure periods, and four distance categories, for a total of 108 classes, 102 of which are observed in the data.

A total of eleven studies use only binary classification. This includes the eight studies which model only two modes [S2; S6; S11; S13; S25; S27; S35; S42] and the three studies which use one-vs-rest/one-vs-one modelling without combining individual models to a single classifier [S9; S31; S49].

Figure 3.4 shows the frequency of each mode/grouping of modes considered in each study. The *car* mode is the most commonly modelled, appearing in 45 studies, followed by *walk* (28 studies) and *public transport* (27 studies). Certain modes either appear individually or grouped. For example, cycling is treated as an independent mode in 21 studies and grouped with walking in nine studies. The grouping of public transport modes cannot be immediately understood from Fig. 3.4, due to different combinations of groupings being possible. For example, for many studies, rail services are not a viable mode of transport, and so *bus* is the only mode considered. Twenty-seven studies consider all public transport modes under one combined *public transport* mode. Of the 25 studies which consider the independent *bus* mode, 14 include *bus* as the only public transport mode. A total of 15 studies consider two or more separate public transport modes.

Q2e: Modelling of mode-alternatives

In order to understand the impact that the transport network has on mode choice, it is necessary for the dataset to include attributes of the mode-alternatives, e.g. the expected duration and cost of travelling by each mode in the choice-set. These are commonly referred to as Level of Service (LOS) attributes in the literature. For revealed preference data, typically only details of the choice made by the passenger are recorded. As such, details of the mode-alternatives need to be synthesised and added to the dataset to be included in the modelling.

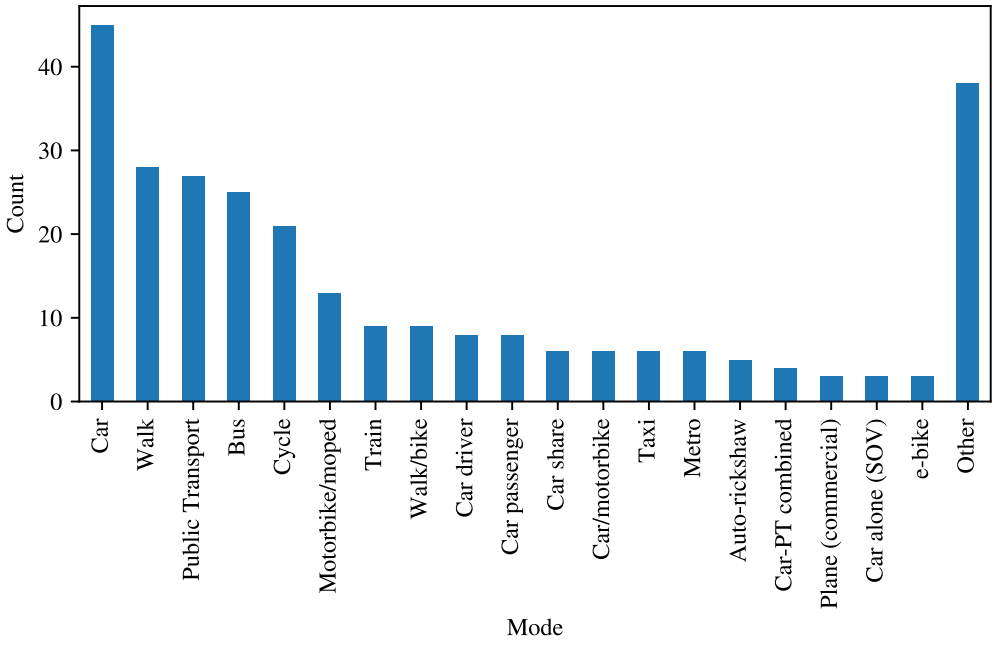


Fig. 3.4 Frequency bar chart of individual modes/grouping of modes in each study in review. The 'Other' category groups all modes/combinations of modes with less than three occurrences across all studies.

Of the 59 studies which use revealed preference data, 27 include no attributes of the mode-alternatives in the choice-set [S2; S7; S12; S13; S15; S17; S18; S19; S20; S22; S25; S26; S27; S29; S36; S37; S40, S41; S44; S46; S48; S50; S52; S53; S54; S56; S59]. A further two studies do not list the input features used in the model with enough clarity to deduce whether any attributes of the mode-alternatives are included [S43; S55].

Table 3.8 shows the relevant features used in the 30 papers which include attributes of the mode-alternatives. The definition of each term is given below:

- *Duration* - journey time from start-point to end-point (including access, transfers etc.)
- *Cost* - Out of pocket cost (e.g. transport fares, Vehicle Operating Costs)
- *Generalised costs* - Combined duration and cost as a single value of disutility, expressed in the unit of currency
- *Vehicle Operating Cost (VOC)* - the mileage dependent costs of operating a vehicle (e.g. fuel, tires, maintenance, repairs, depreciation)
- *In-Vehicle Travel Time (IVTT)* - the duration spent in vehicle/ on-board public transport services
- *Out-of-Vehicle Travel Time (OVTT)* - the combined access, egress, transfer, and waiting durations for Public Transport (PT)

Table 3.8 Attributes of mode-alternatives in selected studies in review. Unless stated otherwise, each attribute is a duration. *PT*=Public Transport, *IVTT*=In-Vehicle Travel Time, *OVTT*=Out-of-Vehicle Travel Time, *VOC*=Vehicle Operating Cost

No.	Duration	Cost	Other	No.	Duration	Cost	Other
S3	✓	✓		**S28**			Duration, VOC (Drive), IVTT (train)
S4	✓	✓	Access (Bus)	**S30**	✓	✓	
S5.1	✓	✓	Transfer, access/egress (PT)	**S31**			Duration ratios (Each mode)
S5.2	✓	✓		**S33**			Generalised costs (Each mode)
S6		✓	Access, egress, IVTT (PT)	**S34**			Generalised costs (Each mode)
S8			Generalised costs (Each mode)	**S35**			IVTT, transfer, speed, directness (PT)
S9	✓	✓		**S38**		✓	IVTT and OVTT (Each mode)
S10	✓		Access distance (PT)	**S39**			IVTT and route distance (Each mode)
S11	✓	✓		**S45**			OVTT (Bus)
S14	✓		Access & egress distance (PT)	**S47**	✓	✓	
S21.1	✓	✓	IVTT (PT)	**S49**	✓		
S21.2	✓	✓		**S51**	✓		
S21.3	✓	✓	IVTT (PT)	**S57**	✓	✓	
S23	✓	✓		**S58**	✓	✓	
S24	✓	✓		**S60**	✓	✓	

- *Access* - The walking duration/distance between the start-point and first public transport access stop
- *Egress* - The walking duration/distance between the last public transport stop and the end-point

Fourteen of the 30 studies which model mode-alternatives do not state the methods used to calculate these attributes [S4; S6; S8; S11; S21.2; S21.3; S23; S24; S31; S39; S45; S47; S57; S58]. Fourteen studies use zonal, time-independent (static) transport models to calculate durations and/or costs [S3; S5.1; S5.2; S9; S10; S21.1; S28; S30; S33; S34; S35; S39; S49; S60]. One study [S5.2] additionally makes use of a time-dependent public transport model to calculate transfer and combined access/egress durations for the PT route at the time of departure. Finally, two studies [S14; S51] make use of an online directions service to generate trip durations.

Q2f: Input features dependent on output choice

In order to be used as a valid predictive model, model input features must be independent of the output choice. Features which are dependent on the choice, e.g. the recorded trip duration (which is dependent on the mode taken) cannot be known until the trip is made, and so cannot be used for prediction.

A significant number of studies (17/63) include input features which are related to the output choice, either directly or indirectly.

Eight studies [S2; S25; S26; S29; S40; S49; S54; S56] include the recorded travel duration of the selected mode as an input feature. Four of these studies also include the trip

distance [S26; S29; S40; S49], which would allow the classifier to infer the speed of the mode-selected. A further two studies [S40; S54] additionally include the reported cost of the selected mode.

Two studies [S3; S46] implicitly include the reported duration by including both the reported departure time and arrival time in the feature vector.

Three studies [S21.1; S21.2; S21.3] implicitly include the selected mode in the input feature vector by labelling attributes of the *selected* mode and best *alternative* mode. For example, one node in the DT for S21.2 separates trips between those made by *Auto* and those made by *Metro* on the basis of whether the cost of the selected mode is greater than or equal to 1.6 euro.

Two studies use different definitions of duration in the mode-alternative attributes for the selected mode. [S30] uses the reported duration as the driving duration if the trip is made by car and uses the driving time predicted by a static zonal transport model otherwise. [S51] similarly uses the reported duration for the selected mode, and the duration as predicted by the Google Directions Application Programming Interface (API) for all other modes. In both cases, this may cause leakage of the selected mode into the input feature vector.

Finally, two studies [S13; S37] include survey questions on reasons for not taking a particular mode in the input feature vector.

As with the modelling of mode-alternatives, two studies [S43; S55] provide insufficient detail of the modelling process to determine whether input features are included which are dependent on the output choice.

Q2g: Hierarchical data

As shown by Q2a, 31 studies make use of trip diary data. Household trip survey data has an inherent hierarchical structure: households are made up of multiple people, each of whom make multiple tours, in which there are multiple legs or trips. Elements within the same groups in the hierarchical structure may show interdependency. This hierarchical structure arises from the specific nature of how trip diary data is collected, and introduces strong correlations which can be observed in the data. Formally, three levels of hierarchy can be considered (each with examples of how the structure could cause interdependency):

- *Household-Person (H-P)* - e.g. multiple members of a household travelling together therefore all travelling by the same mode, one person using the only vehicle in a household meaning that others cannot use that vehicle, all members of a household sharing a tendency to/not to travel by a particular mode, etc.

- *Person-Tour (P-T)* - e.g. individual showing a tendency to/not to travel by a particular mode, individual not being able drive/cycle for all tours due to a vehicle/bike not being available to them on the survey date, individual having a season ticket and therefore being more likely to travel by public transport, etc.

- *Tour-Trip (T-T)* - e.g. return trip being highly likely to be made by the same mode as the outbound trip, vehicle/bike not being available for onwards travel as it was not used for first leg (trip) in tour, vehicle/bike needing to be used for onwards travel as it was used for first leg (trip) in tour and cannot be left behind, etc.

Individual survey trip diaries do not have a household-person grouping, leaving person-tour and tour-trip groupings.

Many of the studies which make use of trip diary data sample the data in a way which removes all/part of the hierarchical structure, e.g. by sampling only outbound trips (removes tour-trip hierarchy), or by sampling only trips made by one member of a household (removes household-person hierarchy). This sampling is presented in Table 3.9.

Additionally, the commute patterns analysed in S33 and S34 are taken from a household survey, with multiple members in each household. As such, these datasets contain a household-person hierarchical structure.

Finally, there may be hierarchical structure in the studies with datasets of unknown nature [S6; S11; S19; S39; S40; S43; S45; S55].

Table 3.9 shows the levels present in the input dataset (after any sampling/processing) for all studies which make use of hierarchical data.

Whilst S3 uses a tour-based analysis, it still predicts mode choice for individual trips, and so the Tour-Trip hierarchy in the data is still present. In total, there are 35 studies which use hierarchical data, or data which may be hierarchical, after sampling/processing. This includes 10 studies which use complete, unsampled trip diaries [S3; S15; S29; S35; S38; S41; S44; S46; S49; S51; S60].

3.3.3.1 Model datasets - limitations

Three limitations are identified in relation to the datasets used to investigate mode choice. Two limitations are technical: (i) studies not including any attributes of the mode-alternatives, (ii) studies using input features dependent on output choice; and three limitations are general: (i) not describing the dataset and modelling process in sufficient detail, (ii) the shortage of studies using large datasets to investigate mode choice, (iii) the lack of relevant, openly available datasets including mode-alternative attributes.

Table 3.9 Details of hierarchies in datasets in relevant studies in review, after sampling/ processing. *H-P*=Household-Person, *P-T*=Person-Tour, *T-T*=Tour-Trip.

No.	H-P	P-T	T-T	Sampling	No.	H-P	P-T	T-T	Sampling
S3	✓	✓	✓	None (complete trip diary, household)	**S36**	✓	✓		Tours from household trip diary
S6	?	?	?	Unclear data	**S37**				Mobility of head of household only
S7	✓	✓		Outbound trips only	**S38**	✓	✓	✓	None (complete trip diary, household)
S9				Mobility of similar clusters	**S39**	?			Work trips only, sampled from larger survey
S11	?	?	?	Unclear data	**S40**	?	?	?	Unclear data
S12	?	?	?	Random sampling from larger survey	**S41**	✓	✓	✓	None (complete trip diary, household)
S14	✓	?		Outbound shopping trips only	**S43**	?	?	?	Unclear data
S15	✓	✓	✓	None (complete trip diary, household)	**S44**		✓	✓	None (complete trip diary, individual)
S17	✓			First trip in day only	**S45**	?	?	?	Unclear data
S19	?	?	?	Unclear data	**S46**	✓	✓	✓	None (complete trip diary, household)
S22	?	?	?	Education trips only	**S48**	?	?	?	Random sampling from trip diaries
S24	✓	✓		Home-based trips only	**S49**	✓	✓	✓	None (complete trip diary, household)
S26				Household mobility only	**S50**		✓		None (complete activity diary, individual)
S27				Household mobility only	**S51**	✓	✓	✓	None (complete trip diary, household)
S28	✓			Morning home-work trips only	**S53**	?			Outbound work trips only
S29		✓	✓	None (complete trip diary, individual)	**S54**	?	?		Outbound work trips only (2-day)
S30				Commute patterns from individual survey	**S55**	?	?	?	Unclear data
S31				Commute patterns from individual survey	**S57**	?			Outbound work trips only
S33	✓			None (Household survey)	**S59**	?	?	?	Random sampling from larger survey
S34	✓			None (Household survey)	**S60**	✓	?		Home-based social trips only
S35		✓	✓	None (complete trip diary, individual)					

Note that using hierarchical data is not an issue in itself, as long as appropriate sampling is used for validation. This is therefore discussed in Section 3.3.4.

Two technical limitations are identified related to datasets. Q2f identifies 27 studies which include no attributes of the mode-alternatives in the choice-set, and a further two studies which do not list the input features used in the model with enough detail to be able to determine whether any attributes of the mode-alternatives are included. These studies therefore do not allow for modelling the impacts of changes to the transport network on the mode choice decisions made by an individual. Of the studies which do model mode-alternatives, the majority generate LOS variables from static zonal graphs. This means that they do not capture the highly granular spatial and temporal variability of conditions on a transport network.

Q2g identifies 17 studies which include input features which are related to the output choice. These features cannot be known in advance of a trip being made, and so these models cannot be used for prediction. Furthermore, these variables allow data-leakage from the output to the input of the model, therefore overstating the achievable performance of model within these studies. Again, a further two studies provide insufficient detail of the modelling process to be able to determine whether any input features which are dependent on the output choice are included.

Of the two technical limitations related to datasets, using input features dependent on output choice is explicitly bad-practice, and is likely result in incorrect conclusions being drawn from the modelling results. Conversely, not including any attributes of the mode-

alternatives is an area for improvement, as doing so is likely to improve the performance of the model.

The discussion of the research question also highlights four general limitations. Firstly, the vast majority of studies (49/63) analysed make use of small to medium datasets, with less than 10 000 entries. Only 12 studies make use of datasets with more than 10 000 entries, and as such there is a need for further investigation into problems of this scale.

Secondly, multiple studies do not describe the dataset and modelling process in sufficient detail for the required information for the systematic review to be extracted. This is problematic for repeatability of the mode choice experiments implemented in these studies, particularly when there is such large variation in the methodologies used in each study. In order to ensure repeatability of the results, methodologies should be recorded in detail, and where possible, data and code should be made available.

Finally, there is a need for a relevant, openly available dataset including mode-alternative attributes. There exist several openly available, large datasets for investigating passenger mode choice. Of the 12 studies which use datasets with greater than 10 000 entries, eight make use of openly available datasets [S3; S9; S15; S17; S29; S36; S46; S52]. However, only two of these studies [S3; S9] add mode-alternative information to these datasets, and the processed dataset is not openly available for either study. As mentioned, only the processed dataset for S15 is openly available, and this dataset does not include any mode-alternative attributes.

3.3.4 How is model performance determined?

The following sections discuss the techniques used to determine model performance in the 63 studies in the review, focusing in turn on the validation method, the sampling method, and the performance metrics used.

Q3a: Validation method

The validation method most commonly used in the studies is holdout validation (non-repeated), which is used in 43 studies. Train-test splits range from 23:77 to 91:9, but the most commonly used splits are 70:30 (10 studies), and 80:20 (nine studies).

Seven studies use repeated holdout validation: S2 runs 3 repetitions of a 75:25 split, S13 runs 10 repetitions of a 75:25 split, S32 runs 10 repetitions of a 70:30 split, S48 runs 50 repetitions of a 70:30 split, S51 runs 100 repetitions of a 75:25 split, and finally S33 and S34 run 100 repetitions of a 60:40 split. Confusingly, S48 only shows the results for both the train and validation data combined, averaged over the 50 runs.

k-fold cross-validation is used in six studies. Four studies use 10-fold cross-validation [S15; S17; S24; S60], and two studies use 5-fold cross-validation [S30; S36]. As well as 10-fold cross-validation, S24 also performs holdout validation (60:40 split).

Three papers use different validation techniques for different models. Whilst they all use in-sample validation for the logit models, S1 uses 80:20 holdout validation for the neuro-fuzzy multinomial logit model; S21.2 uses 60:40 holdout validation for the ANN, DT, and discriminant analysis models; and S26 uses Out-Of-Bootstrap (OOB) error for the Random Forest (RF) model.

Four studies use in-sample validation for all models [S25; S50; S52; S56].

Finally, three studies do not state the validation method used [S3; S20; S35].

Q3b: Sampling method

Of the 35 studies which use hierarchical data, or data which may be hierarchical, none mention the use of grouped (by household or individual) sampling. This includes all 10 studies which make use of complete, unsampled trip diaries. Furthermore, two studies which make use of trip diaries [S3; S35] do not state which validation technique is used at all (see Q3a).

All studies which perform out-of-sample validation appear to use random sampling (either stated explicitly or assumed). Only one study [S16] tests models on data collected separately from, or after, the training data (*external validation*). Each city-specific model (Melbourne and Sydney) is additionally validated on the data from the other city, as well as the holdout test sample.

Q3c: Performance metrics

Table 3.10 shows the performance metrics used for validation in each study. Performance metrics used only during model estimation/fitting are not recorded, except for studies which use in-sample validation.

Four different discrete classification metrics are shown in the table: accuracy, recall, the confusion matrix, and the predicted mode shares. Precision and specificity are also used by S3 and S31 respectively, but not recorded in the table.

Metrics which evaluate probabilistic classification are grouped together in Table 3.10. Seven different probabilistic metrics are used: percent clearly right/wrong/unclear [S4; S5.1; S5.2], Arithmetic Mean Probability of Correct Assignment (AMPCA) (referred to as fitting factor in S4, S5.1, and S5.2; and average probability of correct assessment in S33), Mean Squared Error (MSE) [S4; S5.1; S5.2; S43], simulated mode shares [S4; S5.1; S5.2; S32],

Receiver Operating Characteristic (ROC) curves [S18; S19; S60], log-likelihood [S30; S32], Bayesian Information Criterion (BIC) [S30], and Expected Simulation Recall (ESR) [S33].

Table 3.10 does not show the metrics used in two studies. S9 performs regression on the total number of trips performed by each mode within a cluster, and so uses regression-based metrics (MSE and average relative variance of regression). S16 uses three metrics: *predicted share less observed share*, *weighted percent correct*, and *weighted success index*. However, no definitions for the performance metrics are provided in the paper, and so it cannot be determined if the metrics are discrete or probabilistic.

In total, 52 of the remaining 61 studies use only discrete classification metrics, whilst nine studies use a combination of probabilistic and discrete classification metrics. Of the studies which use only discrete classification metrics, 29 make use of LR models.

Twelve studies use only one performance metric: accuracy is used as the sole metric in 10 studies [S2; S6; S8; S17; S20; S26; S35; S41; S44; S45], recall per mode in S25, and the confusion matrix in S47.

3.3.4.1 Model performance estimation - limitations

Four technical limitations are identified in relation the model performance estimation techniques used in the studies: (i) studies using inappropriate validation schemes, (ii) studies using incorrect sampling methods for hierarchical data, (iii) studies using only discrete metrics, (iv) studies not performing external validation.

Q3a identifies 10 studies which make use of inappropriate validation schemes. This includes four studies which use in-sample validation [S25; S50; S52; S56], three studies which use different validation techniques for different models being tested [S1; S21.2; S26], and three which do not state the validation method being used [S3; S20; S35]. Formal validation of a model on data separate training data is essential to ensure models have generalised to the training data without overfitting. Additionally, in order to make any valid comparisons between models, it is essential the validation scheme is constant across models.

Q2g identifies 20 studies which make use of hierarchical data, and a further 15 which makes use of data which may be hierarchical. As identified by Q3b, none of these studies sample validation sets or folds grouped by individual or household. As such, trips from the same group (household/person/tour) will occur in both test and training data, allowing for data-leakage and overfitting. This is particularly problematic for the 10 studies which use complete trip diary data [S3; S15; S29; S35; S38; S41; S44; S46; S49; S51; S60]. Many of these trip diaries are multi-day, compounding the issue. Notably, S15 uses a six-day trip diary (average 3.3 trips per person), and S44 uses sets of GPS trips logged over four months (average 58.4 trips per person). This problem is not unique to mode choice modelling

Table 3.10 Summary of performance metrics used for validation in each study in review. **Acc**=Accuracy, **Rec**=Recall, **CM**=Confusion Matrix, **MS**=Mode Shares (Discrete), **Pro**=Probabilistic metric.

No.	Acc	Rec	CM	MS	Pro	No.	Acc	Rec	CM	MS	Pro
S1	✓		✓			**S30**	✓		✓		✓
S2	✓					**S31**	✓	✓		✓	
S3	✓	✓				**S32**	✓			✓	✓
S4	✓	✓			✓	**S33**			✓		✓
S5.1	✓	✓			✓	**S34**	✓	✓	✓		
S5.2	✓	✓			✓	**S35**	✓				
S6	✓					**S36**		✓	✓		
S7	✓			✓		**S37**	✓	✓			
S8	✓					**S38**	✓	✓		✓	
S9	-	-	-	-	-	**S39**	✓	✓			
S10	✓	✓				**S40**	✓	✓	✓		
S11	✓		✓			**S41**	✓				
S12	✓	✓	✓			**S42**	✓		✓		
S13	✓	✓				**S43**	✓				✓
S14	✓	✓				**S44**	✓				
S15	✓	✓	✓			**S45**	✓				
S16	-	-	-	-	-	**S46**	✓			✓	
S17	✓					**S47**			✓		
S18	✓		✓		✓	**S48**	✓	✓	✓		
S19	✓	✓	✓		✓	**S49**	✓		✓		
S20	✓					**S50**	✓	✓		✓	
S21.1		✓	✓			**S51**	✓	✓		✓	
S21.2	✓	✓	✓			**S52**	✓		✓		
S21.3		✓	✓			**S53**	✓		✓		
S22	✓		✓	✓		**S54**	✓	✓	✓	✓	
S23	✓		✓	✓		**S55**	✓	✓	✓		
S24	✓		✓			**S56**	✓	✓			
S25		✓				**S57**	✓	✓	✓		
S26	✓					**S58**		✓		✓	
S27	✓	✓	✓			**S59**	✓	✓		✓	
S28	✓	✓	✓			**S60**	✓	✓	✓		
S29	✓	✓				**Sum**	54	34	29	12	9

applications. Saeb et al. (2017) conduct a review of sampling methods in studies using ML to make clinical predictions from smartphone or wearable technology data. They review studies which use hierarchical data, where there are multiple *records* for each individual *subject*. They find that of the 62 of the studies included in the meta-analysis, 28 (45 %) use inappropriate *record-wise* sampling, instead of *subject-wise* sampling.

Q3b identifies that only one of the studies reviewed [S16] uses *external validation*, where the model is validated on data collected separately from, or after, the training data. External validation using future data is the only possible method of directly simulating the use case for a mode choice model, of predicting future, unknown trips. External validation can also identify issues with data-leakage, overfitting, and incorrect validation schemes, e.g. the incorrect sampling methods for hierarchical data, as highlighted by Q2g.

Finally, Q3c identifies that the vast majority of studies (51/63) use purely discrete metrics to assess model performance. This includes 29 studies which assess LR using only discrete classification metrics, despite LR being a statistical technique intended to generate probability distributions. In total, only six studies make use of *proper* continuous scoring metrics, log-likelihood [S30; S32] and MSE [S4; S5.1; S5.2; S43].

Of the four technical limitations related to model performance estimation, three explicitly represent bad practice (using inappropriate validation schemes, using incorrect sampling for hierarchical data, and using only discrete metrics), and one represents an area for improvement (not performing external validation).

3.3.5 How are optimal model hyper-parameters selected?

This section discusses the techniques used to optimise model specifications and hyper-parameters for conventional ML algorithms (ANNs, DTs, EL, SVMs). The 14 studies which do not use at least one these algorithms are therefore omitted from this section of the review [S1; S7; S8; S11; S22; S23; S28; S29; S30; S35; S38; S42; S45; S52].

The following sections review the remaining 49 studies, focusing in turn on the hyper-parameter search method, the hyper-parameter validation method, and the hyper-parameter validation data.

Q4a: Hyper-parameter search method

Of the 49 studies which use at least one conventional ML algorithm, 11 do not mention hyper-parameter values at all within the paper [S12; S20; S21.1; S21.2; S21.3; S25; S26; S41; S43; S46; S56]. A further nine studies either state hyper-parameter values without

explanation [S10; S17; S31; S36; S37; S47; S60], or state that they use default values [S27; S59].

This leaves 29 studies which use some form of hyper-parameter optimisation. Twelve studies [S2; S3; S4; S9; S13; S14; S16; S24; S39; S50; S51; S53] perform a manual search, or trial and error, in order to identify model parameters. Of these, S3 searches only for the kernel function in a SVM and uses default values for all other parameters and models, and S39 searches for the number of neurons in a single test layer, again using default values for other parameters.

Nine studies [S6; S18; S33; S34; S40; S48; S49; S55; S57; S58] specify a Multi-Layer Perceptron (MLP) with a single hidden layer and perform a linear search on the number of neurons in that layer. With the exception of S57, which performs a grid search for the SVM parameters (γ and C), default values are used for all other parameters of all models.

One study [S49] uses a repeated linear search, firstly on the loss-weight ratio of the two classes in each model, and secondly on the number of features used.

Four studies [S5.1; S5.2; S15; S32; S54] make use of a grid search. S32 tests a range of specific ANN structures, with different numbers of hidden layers and nodes, but uses a grid search to select the appropriate dropout ratio and learning rate.

One study [S19], tests two different search strategies in order to find optimal SVM parameters (γ and C): grid search and genetic algorithms. The study finds that whilst the two methods find optimal solutions with similar accuracies, the genetic algorithm finds the solution with the lower penalty parameter (C), and so is preferred.

Finally, one study [S44] states that cross-validation is used to select model parameters but does not state the search method used.

Q4b: Hyper-parameter validation method

Of the 29 studies which use some form of hyper-parameter optimisation, 10 do not state the validation method used to determine optimal values [S2; S3; S9; S13; S24; S39; S40; S48; S55; S58]. S3 states the parameters with best performance are used but does not state how this is determined.

Eight studies [4; 14; 16; 18; 32; 34; 54; 57] use holdout validation. Seven studies [15; 19; 44; 49; 50; 51; 53] use k-fold cross-validation. One study [S33] uses repeated holdout validation. One study [S6] uses in-sample validation.

Finally, two studies [S5.1, S5.2] use a complex multi-criteria assessment, involving relative performance on both the calibration and validation data.

Q4c: Hyper-parameter validation data

Of the 29 studies which use some form of hyper-parameter optimisation, 14 do not state the data used for hyper-parameter validation [S2; S3; S9; S13; S16; S19; S24; S34; S39; S40; S44; S48; S55; S58].

Of the seven studies which use k-fold cross-validation to test hyper-parameter performance, two use only the training data [S53; S57], one uses a random subset of 43 % of the data [S15], two use all of the data [S50; S51], and two do not state the data used (included above). The study which uses repeated validation also uses all of the data [S33].

Of the eight studies which use holdout validation, three use the data reserved for model testing [S4; S14; S32], two use only the train data, dividing it into a new test and train fold [S53; S54], one uses a separate 15 % validation sample which is not used for model testing or training [S18], and two do not state the data used (included above).

Finally, the two studies which use the multi-criteria assessment use both the train and test data.

3.3.5.1 Model optimisation - limitations

Four limitations are identified in relation the model optimisation techniques used in the studies. Three limitations are technical: (i) studies not performing any type of hyper-parameter optimisation, (ii) studies not using rigorous hyper-parameter search schemes, (iii) studies optimising hyper-parameters on validation data; and one is general: not presenting model hyper-parameters used within the study.

Three technical limitations are identified by the attributes related to hyper-parameter optimisation collected in the systematic review. Of the 49 studies which use one or more conventional ML models to investigate mode choice, Q4a identifies 20 studies which do not perform any type of hyper-parameter optimisation. This includes 11 which do not state hyper-parameter values at all, and nine which use default values or provide values without explanation. As model performance is highly dependent on hyper-parameter values, this step is essential for fair model comparison.

Q4a also identifies that no studies use a fully rigorous hyper-parameter search method. Many studies use inconsistent search methods, only searching over one parameter within one model (e.g. number of neurons in a hidden layer), whilst leaving all others with default values. Additionally, only one study uses an automated sequential search (genetic algorithm in S19) to optimise model hyper-parameters, the rest either using a pre-specified search space (linear search/grid search) or manual search/trial and error.

Finally, Q4c identifies eight studies which include the validation data in the hyper-parameter search [S4; S5.1; S5.2; S14; S32; S33; S50; S51], as well as 14 which do not state the data used [S2; S3; S9; S13; S16; S19; S24; S34; S39; S40; S44; S48; S55; S58]. Fitting hyper-parameters to the validation data allows for data-leakage, therefore allowing the model to overfit to the validation data using model hyper-parameters.

Holdout test data should not be seen by the model at any time during model development, until model testing.

Of the three technical limitations related to model optimisation, one represents bad-practice (optimising hyper-parameters on validation data), and two represent areas for improvement (not performing hyper-parameter optimisation, and studies not using rigorous hyper-parameter search schemes).

The discussion of Q4c also highlights one general limitation, that studies do not report the model hyper-parameters and hyper-parameter selection schemes with sufficient detail. As with the details of methodologies in Q2, this is problematic for repeatability of the mode choice experiments implemented in these studies. Hyper-parameter values and selection schemes should be recorded in detail in order to ensure repeatability of the studies.

3.3.6 How is the best model selected?

This section discusses the model selection techniques (i.e. selecting from different models with optimised hyper-parameters) used in the 63 studies in the review, reviewing the responses to Q5a.

Across all 63 studies, only four [S15; S32; S48; S51] conduct any analysis of the uncertainty or distribution of model performance. S15 uses 10-fold cross validation to estimate the accuracy of seven different classifiers. Firstly, the study uses a Kruskal-Wallis test at a 5 % significance level to test the null hypothesis that the performance estimates of all classifiers tested are not significantly different from one-another. Secondly, a two-sided Wilcoxon rank-sum test is applied pairwise between the classifiers to test whether different pairs of classifiers are significantly different from each other.

Three papers [S32; S48; S51] estimate the standard deviation of the metrics (accuracy in S32 and S48, and accuracy and recall in S51) across each run of k-fold cross-validation/ repeated holdout validation. These estimates of standard deviations are not used to form any formal significance tests in these studies.

Model selection - limitations

One technical limitation is identified in relation the model selection techniques used in the studies: studies not analysing uncertainty in performance estimates.

Q5a identifies that 59 out of the 63 studies do not analyse the expected distribution of the performance estimates. Whilst several papers discuss the relative performance of classifiers for the mode choice prediction task, only one [S15] applies any formal test to investigate the statistical significance of differences between the classifiers. Additionally, as a discontinuous scoring metric (accuracy) is used and the number evaluations is low (10 folds of cross-validation) the direct distribution of the metric cannot be analysed, and instead non-parametric pairwise testing is used.

This limitation represents an area for improvement.

3.4 Summary

This chapter conducts a systematic review of ML methodologies for modelling passenger mode choice. The review investigates five research questions covering classification techniques, datasets, performance estimation, model optimisation, and model selection.

A comprehensive search methodology across the three largest online publication databases is designed and used to identify 468 unique records. The record titles, abstracts, and publication details are screened for relevance, leaving 96 articles. The technical content of the full-text of these articles is assessed according to the eligibility criteria. In total, following the two screening processes, 60 full text peer-reviewed articles containing 63 primary studies are used for data extraction.

The studies are each reviewed in detail to extract 15 attributes covering the five research questions. Through this process, 17 limitations are identified: 10 technical limitations, and seven general limitations. The limitations are summarised in Table 3.11. As shown in the table, each technical limitation belongs to one of the classification stages out of classification techniques, datasets, performance estimation, model optimisation, and model selection.

Of the 10 limitations, five represent *bad-practice* modelling decisions which are likely to impact the results of an investigation, and five are identified as *areas for improvement* which are not incorrect but could be addressed in order to improve the reliability of the results and/or predictive performance of the models.

A full summary of the technical limitations present in each study is given in Table 3.12. All studies have at least three technical limitations within their methodology, and only one study does not have any of the *bad-practice* limitations [S18].

Table 3.11 Limitations identified within systematic review.

No.	Classification stage	Description	Type
	Technical limitations		
TL1	Datasets	Studies not including any attributes of the mode-alternatives	Area for improvement
TL2	Datasets	Studies using input features which are dependent on output choice	Bad-practice
TL3	Performance estimation	Studies using inappropriate validation schemes	Bad-practice
TL4	Performance estimation	Studies using incorrect sampling methods for hierarchical data	Bad-practice
TL5	Performance estimation	Studies not performing external validation	Area for improvement
TL6	Performance estimation	Studies using only discrete metrics	Bad-practice
TL7	Model optimisation	Studies not performing any type of hyper-parameter optimisation	Area for improvement
TL8	Model optimisation	Studies not using rigorous hyper-parameter search schemes	Area for improvement
TL9	Model optimisation	Studies optimising hyper-parameters on test data	Bad-practice
TL10	Model selection	Studies not analysing uncertainty in performance estimates	Area for improvement
	General limitations		
GL1	Classification techniques	Limited number of studies which systematically compare several classifiers on the same task	
GL2	Classification techniques	Relatively low number of investigations into EL algorithms, in particular GBDT	
GL3	Classification techniques	Inconsistent representation of DCMs in ML studies	
GL4	Datasets	Not describing the dataset and modelling process in sufficient detail	
GL5	Datasets	Shortage of studies using large datasets to investigate mode-choice	
GL6	Datasets	Lack of relevant, openly available datasets including mode-alternative attributes	
GL7	Model optimisation	Not presenting specific model hyper-parameters	

As discussed in Section 3.1, this chapter identifies the technical limitations, and quantifies their prevalence in existing research. In order to gain a deeper understanding of the theory behind each technical limitation, Chapter 4 introduces a new theoretical framework, and uses this framework to analyse each technical limitation. The framework follows the same structure as the review, covering each of the classification stages shown in Table 3.11 (with the addition of model prediction and model fitting). The analysis in Chapter 4 is used to formulate the methodology presented in Chapter 5, which specifically addresses each methodological limitation and demonstrates how each one is solved within this research.

The general limitations identified in this chapter are used to assess the impacts of the work of this thesis in Chapter 8.

Limitations of systematic review

This section analyses the limitations of the review with respect to the recommended PRISMA guidelines (Moher et al. 2009).

Whilst a comprehensive and exhaustive search methodology covering the three largest online databases was used to identify relevant literature, there may have been relevant studies which are not included. Additionally, the review does not consider grey literature or unpublished material. However, in this new, research-led field, the author is confident that the state-of-the-art techniques are well covered by the sample of papers assembled.

Table 3.12 Summary of limitations within each study in systematic review.

Number	Paper	TL1	TL2	TL3	TL4	TL5	TL6	TL7	TL8	TL9	TL10	Sum
S1	Andrade, Uchida, and Kagaya (2006)			✓		✓	✓	NA	NA		✓	4
S2	Assi et al. (2018)	✓	✓			✓	✓		✓	✓	✓	7
S3	Biagioni et al. (2009)		✓	✓	✓	✓	✓		✓	✓	✓	8
S4	Cantarella and de Luca (2003)					✓			✓	✓	✓	4
S5.1	Cantarella and de Luca (2005)					✓			✓	✓	✓	4
S5.2	-					✓			✓	✓	✓	4
S6	Chalumuri et al. (2009)				?	✓	✓		✓		✓	5
S7	Cheng et al. (2014)	✓			✓	✓	✓	NA	NA		✓	5
S8	Dell'Orco and Ottomanelli (2012)					✓	✓	NA	NA		✓	3
S9	Edara, Teodorović, and Baik (2007)					✓	NA		✓	✓	✓	4
S10	Ermagun, Rashidi, and Lari (2015)					✓	✓	✓	✓		✓	5
S11	Errampalli, Okushima, and Akiyama (2007)				?	✓	✓	NA	NA		✓	4
S12	Gao et al. (2013)	✓			?	✓	✓	✓	✓		✓	7
S13	Gazder and Ratrout (2015)	✓	✓			✓	✓		✓	✓	✓	7
S14	Golshani et al. (2018)				✓	✓	✓		✓	✓	✓	6
S15	Hagenauer and Helbich (2017)	✓			✓	✓	✓		✓			5
S16	Hensher and Ton (2000)						?		✓	✓	✓	4
S17	Hossein Rashidi and Hasegawa (2014)	✓			✓	✓	✓	✓	✓		✓	7
S18	Hussain et al. (2017)	✓				✓			✓		✓	4
S19	Jia, Cao, and Yang (2015)	✓			?	✓			✓	✓	✓	6
S20	Juremalani (2017)	✓		✓		✓	✓	✓	✓		✓	7
S21.1	Karlaftis (2004)		✓			✓	✓	✓	✓		✓	6
S21.2	-		✓	✓		✓	✓	✓	✓		✓	7
S21.3	-		✓			✓	✓	✓	✓		✓	6
S22	Kedia, Saw, and Katti (2015)	✓			?	✓	✓	NA	NA		✓	5
S23	Kumar, Sarkar, and Madhu (2013)					✓	✓	NA	NA		✓	3
S24	Lee, Derrible, and Pereira (2018)				✓	✓	✓		✓	✓	✓	6
S25	Li et al. (2016)	✓	✓	✓		✓	✓	✓	✓		✓	8
S26	Liang et al. (2018)	✓	✓	✓		✓	✓	✓	✓		✓	8
S27	Lindner, Pitombo, and Cunha (2017)	✓				✓	✓	✓	✓		✓	6
S28	Lu and Kawamura (2010)				✓	✓	✓	NA	NA		✓	4
S29	Ma (2015)	✓	✓		✓	✓	✓	NA	NA		✓	6
S30	Ma, Chow, and Xu (2017)		✓			✓		NA	NA		✓	3
S31	Moons, Wets, and Aerts (2007)					✓	✓	✓	✓		✓	5
S32	Nam et al. (2017)					✓			✓	✓		3
S33	Omrani (2015)				✓	✓			✓	✓	✓	5
S34	Omrani et al. (2013)				✓	✓	✓		✓	✓	✓	6
S35	Papaioannou and Martinez (2015)			✓	✓	✓	✓	NA	NA		✓	5
S36	Pirra and Diana (2017)	✓			✓	✓	✓	✓	✓		✓	7
S37	Pitombo et al. (2015)	✓	✓			✓	✓	✓	✓		✓	7
S38	Pulugurta, Arun, and Errampalli (2013)				✓	✓	✓	NA	NA		✓	4
S39	Raju, Sikdar, and Dhingra (1996)				?	✓	✓		✓	✓	✓	6
S40	Ramanuj and Gundaliya (2013)	✓	✓		?	✓	✓		✓	✓	✓	8
S41	Rasouli and Timmermans (2014)	✓			✓	✓	✓	✓	✓		✓	7
S42	Seetharaman et al. (2009)					✓	✓	NA	NA		✓	3
S43	Sekhar, Minal, and Madhu (2016)	?	?		?	✓		✓	✓		✓	7
S44	Semanjski, Lopez, and Gautama (2016)	✓				✓	✓		✓	✓	✓	6
S45	Shafahi and Nazari (2006)				?	✓	✓	NA	NA		✓	4
S46	Shukla et al. (2013)	✓	✓		✓	✓	✓	✓	✓		✓	8
S47	Subba Rao et al. (1998)					✓	✓	✓	✓		✓	5
S48	Tang, Xiong, and Zhang (2015)	✓			?	✓	✓		✓	✓		6
S49	Tang, Yang, and Zhang (2012)		✓		✓	✓	✓		✓		✓	6
S50	Van Middelkoop, Borgers, and Timmermans (2003)	✓		✓	✓	✓	✓		✓	✓	✓	8
S51	Wang and Ross (2018)		✓		✓	✓	✓		✓	✓		6
S52	Wang and Namgung (2007)	✓		✓		✓	✓	NA	NA		✓	5
S53	Xian-Yu (2011)	✓			?	✓	✓		✓		✓	6
S54	Xie, Lu, and Parkany (2003)	✓	✓		?	✓	✓		✓		✓	7
S55	Yin and Guan (2011)	?	?		?	✓	✓		✓	✓	✓	8
S56	Zenina and Borisov (2011)	✓	✓	✓		✓	✓	✓	✓		✓	8
S57	Zhang and Xie (2008)				?	✓	✓		✓		✓	5
S58	Zhao et al. (2010)					✓	✓		✓	✓	✓	5
S59	Zhou and Lu (2011)	✓			?	✓	✓	✓	✓		✓	7
S60	Zhu et al. (2017)				✓	✓	✓	✓	✓		✓	6
Sum		29	19	10	34	62	53	20	49	22	62	

This review focuses purely on the methodologies used and makes no attempt to draw conclusions on the findings reported by each paper. As such, no assessment is made of the quality of each paper, nor the publication bias of the field.

Whilst the procedure for the review is designed to be as objective as possible, the data extraction and discussion is carried out by a single author, under the guidance of the PhD supervisors. This is according to available resources for the PhD. All results and decisions have been double checked, but there may be remaining errors, which are the responsibility of the author.

Chapter 4

Theoretical framework

4.1 Overview

This chapter introduces a new theoretical framework in order to formalise the requirements for structuring investigations into classification problems. This chapter aims to (a) establish a rigorous theoretical framework for supervised classification which applies to both Machine Learning (ML) and statistical Random Utility Models (RUMs), (b) present a uniform notation which covers the relevant tasks within the classification problem, and (c) assess the technical limitations of existing ML approaches identified by the systematic review in Chapter 3 within this theoretical framework.

Firstly, Section 4.2 presents the formulation for the classification problem, broken down into six stages: (i) model dataset, (ii) model prediction, (iii) model fitting, (iv) model performance estimation, (v) model optimisation, and (vi) model selection. Then, Section 4.3 presents a theoretical analysis of the technical limitations of the existing ML studies found in Chapter 3, in order to demonstrate how each technical limitation may affect the results of the existing studies. Finally, Section 4.4 summarises the work in this chapter.

The theoretical framework presented in this chapter is used to formulate the methodology presented in Chapter 5, which specifically addresses each methodological limitation and demonstrates how each one is solved within this research.

4.2 Formulation of predictive classification problem

This section presents the formulation for the general-purpose classification problem. As such, it is intended to be relevant to all classification problems, including mode choice prediction.

The analysis focuses in turn on six classification stages: (i) model dataset, (ii) model prediction, (iii) model fitting, (iv) model performance estimation, (v) model optimisation, and (vi) model selection. Note that this sequence is determined for ease of presentation, so that each stage builds on the theory introduced in the previous stages, and does not represent the order in which the stages are actually carried out.

The formulation combines existing analysis from Hastie, Friedman, and Tibshirani (2008) and Bergstra and Bengio (2012) with new concepts and analysis developed for this thesis (Hillel, Bierlaire, et al. 2018b), and unites it under a uniform notation. Specifically, the sections on model datasets, prediction, and fitting (Sections 4.2.1 to 4.2.3) adapt existing classification theory, as detailed in Hastie, Friedman, and Tibshirani (2008), modifying the notation to fit that used in this thesis. The analysis in Section 4.2.1 of sampling of the feature vector x_n from the feature set $\mathcal{Z}_n$ is a new concept introduced in this thesis. The section on model performance estimation (Section 4.2.4) presents a novel extension of the notation to cover established concepts of in-sample, out-of-sample, and k-fold cross-validation. The section on model optimisation (Section 4.2.5) further extends the notation to cover concepts introduced by Bergstra and Bengio (2012). Finally, the analysis in the section on model selection (Section 4.2.6) introduces new concepts developed for this thesis, extending the previous work and notation in Hillel, Bierlaire, et al. (2018b).

4.2.1 Model dataset

We consider a set $\mathcal{C}$ containing $N_\mathcal{C}$ elements, called the *population*. The set $\mathcal{C}$ is supposed to be sufficiently large that it is not feasible to enumerate its elements explicitly. We also consider a partition composed of J subsets $\mathcal{C}_i$, $i = 1,\dots,J$, which we call *classes*. We will focus on single-label problems, where each element belongs wholly to a single class. We have

$$\mathcal{C} = \cup_{i=1}^{J} \mathcal{C}_i, \tag{4.1}$$

and

$$\mathcal{C}_i \cap \mathcal{C}_j = \emptyset, \quad \forall i \neq j. \tag{4.2}$$

Each element $n \in \mathcal{C}$ is associated with a set $\mathcal{Z}_n$ containing features which determine its class membership. The set $\mathcal{Z}_n$ is also assumed to be too large to evaluate directly.

For example, consider work commute trips which can either be made by car or public transportation, so that there are $J = 2$ classes. The population $\mathcal{C}$ comprises of the commute trips in a specific city on a specific day, partitioned into those trips made by car ($\mathcal{C}_1$), and those made by public transportation ($\mathcal{C}_2$). The choice for trip n in population $\mathcal{C}$ can be explained both by features which can be observed, such as the departure time, travel duration, travel cost, weather conditions, trip purpose, etc; as well as those which cannot be observed, such as the exact state-of-mind of the individual making the decision. Together, these form the feature set $\mathcal{Z}_n$.

As $\mathcal{C}$ is too large to evaluate directly, we instead consider a dataset $\mathcal{D}$ which comprises of a finite number ($N < N_{\mathcal{C}}$) of elements sampled from the population $\mathcal{C}$, so that $\mathcal{D} \subset \mathcal{C}$.

Additionally, we can only record and analyse a finite set of observable features for each element. We therefore associate each element $n \in \mathcal{D}$ with a finite vector of K features $x_n \in \mathbb{R}^K$. The features in x_n are sampled from the corresponding feature set $\mathcal{Z}_n$.

Each element $n \in \mathcal{D}$ is therefore associated with

1. a finite vector of features x_n, and,

2. a set $y_n \in \{0,1\}^J$ of class indicators (known as the *ground-truth*), such that

$$\sum_{i=1}^{J} y_{in} = 1. \tag{4.3}$$

We denote the dataset $\mathcal{D} = (x_n, y_n)_{n=1}^N = (x_{\mathcal{D}}, y_{\mathcal{D}})$.

4.2.2 Model prediction

A *probabilistic classifier* P is a model which maps the finite vector of features x_n into a probability distribution on the classes

$$P : \mathbb{R}^K \to [0,1]^J. \tag{4.4}$$

We use the notation $P(i|x_n)$ to represent the probability that element n belongs to class i, as provided by the classifier. We have

$$P(i|x_n) \geq 0, \quad i = 1,\ldots,J \quad \text{and} \quad \sum_{i=1}^{J} P(i|x_n) = 1, \quad \forall x_n \in \mathbb{R}^K. \tag{4.5}$$

We use the notation $P(x_n)$ to describe the J-dimensional probability distribution on the classes generated by the classifier for the vector x_n

$$P(x_n) = P(1|x_n), \ldots, P(J|x_n) \tag{4.6}$$

It is convenient to denote the index of the class associated with element n as i_n. From Eq. (4.3) it is defined as

$$i_n = \sum_{i=1}^{J} i y_{in} \tag{4.7}$$

As such, the probability predicted by the classifier for the ground-truth class for element n is denoted $P(i_n|x_n)$.

Following the work commute trip example above, the classifier provides the probability $P(1|x_n)$ that for trip n the traveller chooses to travel by car, and $P(2|x_n)$ that they travel by public transportation. If person n actually travelled by car so that $i_n = 1$, then

$$P(i_n|x_n) = P(1|x_n) \tag{4.8}$$

Each feature vector x_n is made up of a subset of the observable features from $\mathcal{Z}_n$, e.g. travel duration for each mode (car and public transportation), travel cost for each mode, trip purpose.

4.2.3 Model fitting

A trained classifier is an instance of a classification algorithm $\mathcal{A}$ *fitted* to a dataset $\mathcal{D}$. We use the notation $P_{\mathcal{D}}$ to represent a classifier trained on $\mathcal{D}$ so that

$$P_{\mathcal{D}} = \mathcal{A}(\mathcal{D}). \tag{4.9}$$

Model fitting (represented by $\mathcal{A}(\mathcal{D})$) typically has no analytical solution, and instead is carried out iteratively. The nature of the model fitting is dependent on the algorithm $\mathcal{A}$.

4.2.4 Model performance estimation

We use the notation $P(x_{\mathcal{D}})$ to describe the $N \times J$ probability matrix predicting the class indicators $y_{\mathcal{D}}$, which is generated by the classifier P when applied to the feature vectors $x_{\mathcal{D}}$. We can use this notation to define an aggregate measure of fit (performance metric) across the dataset $G(P(x_{\mathcal{D}}), y_{\mathcal{D}})$, which measures how well or how poorly the classifier is able to predict the class indicators $y_{\mathcal{D}}$ when using $x_{\mathcal{D}}$ as an input. To simplify the notation, we use

the shorthand

$$G(P;D) = G(P(x_{\mathcal{D}}), y_{\mathcal{D}}). \tag{4.10}$$

This metric provides a quantitative assessment of the model performance.

It is necessary to quantify model performance in order to

1. Determine optimal model hyper-parameters or utility specifications for a given algorithm on a given dataset for a given task (see Section 4.2.5)

2. Identify the best performing model from a set of models for a given task (see Section 4.2.6)

3. Assess how well the final model will likely perform on new data

In all cases, we need to evaluate how well a trained model predicts the class membership of unknown elements with feature vectors which have potentially not been observed in the dataset $\mathcal{D}$.

We consider an unknown set $\mathcal{U}$, which has the same form as $\mathcal{D}$. Within the context of mode choice prediction, $\mathcal{U}$ represents future trips which have not yet been made, and so does not overlap with the population $\mathcal{C}$, so that

$$\mathcal{U} \cap \mathcal{C} = \emptyset, \tag{4.11}$$

and by extension

$$\mathcal{U} \cap \mathcal{D} = \emptyset. \tag{4.12}$$

As with the dataset $\mathcal{D}$ we denote the unknown set $\mathcal{U} = (x_n, y_n)_{n=1}^{N_{\mathcal{U}}} = (x_{\mathcal{U}}, y_{\mathcal{U}})$. The predictive performance of the fitted model $P_{\mathcal{D}}$ is given by the value of a selected measure of fit G evaluated over $\mathcal{U}$

$$G(P_{\mathcal{D}}; \mathcal{U}) \quad \Big(\equiv G(P_{\mathcal{D}}(x_{\mathcal{U}}), y_{\mathcal{U}}) \Big). \tag{4.13}$$

As $\mathcal{U}$ is unknown, $G(P_{\mathcal{D}}; \mathcal{U})$ is a random variable, the value of which can only be estimated.

The simplest approach to estimating Eq. (4.13) used in the literature is *in-sample validation*, which simply uses the performance of the fitted model on the full dataset $\mathcal{D}$ (the same data the model is fitted to)

$$G(P_{\mathcal{D}}; \mathcal{U}) \approx G(P_{\mathcal{D}}; \mathcal{D}). \tag{4.14}$$

In-sample validation does not account for *generalisation error* from the *bias-variance trade-off*, and so is not appropriate for performance estimation.

Valid approaches to estimating Eq. (4.13) validate the model on data unseen during model fitting. The following notation first describes *holdout validation*, where only one validation

fold is used. It is then extended to *k-fold cross-validation*, and arbitrary repeated validation schemes.

In holdout validation the dataset $\mathcal{D}$ is divided into two subsets: a training set $\mathcal{T}$, composed of $N_\mathcal{T}$ elements; and a validation set $\mathcal{V}$, composed of $N_\mathcal{V}$ elements, so that

$$\mathcal{D} = \mathcal{T} \cup \mathcal{V}, \tag{4.15}$$

$$\mathcal{T} \cap \mathcal{V} = \emptyset. \tag{4.16}$$

Again, we denote the training and validation sets as $\mathcal{T} = (x_n, y_n)_{n=1}^{N_\mathcal{T}} = (x_\mathcal{T}, y_\mathcal{T})$ and $\mathcal{V} = (x_n, y_n)_{n=1}^{N_\mathcal{V}} = (x_\mathcal{V}, y_\mathcal{V})$ respectively.

The model is trained on $\mathcal{T}$, so that $P_\mathcal{T} = \mathcal{A}(\mathcal{T})$. The predictive performance of the model can be estimated by calculating the performance metric on $\mathcal{V}$

$$G(P_\mathcal{D}; \mathcal{U}) \approx G(P_\mathcal{T}; \mathcal{V}). \tag{4.17}$$

This is known as the *test error*. The *train error* is similarly given by $G(P_\mathcal{T}; \mathcal{T})$.

In k-fold cross-validation, $\mathcal{D}$ is divided into k approximately equally sized validation folds $\mathcal{V}_i$ with corresponding training sets $\mathcal{T}_i$, so that

$$\mathcal{D} = \bigcup_{i=1}^{k} \mathcal{V}_i, \tag{4.18}$$

$$\mathcal{V}_i \cap \mathcal{V}_j = \emptyset \quad \forall i \neq j, \tag{4.19}$$

$$\mathcal{T}_i = \bigcup_{j \neq i} \mathcal{V}_j. \tag{4.20}$$

These folds allow for k iterations of the model to be trained and validated, for k separate estimates of model performance

$$P_{\mathcal{T}_i} = \mathcal{A}(\mathcal{T}_i), \tag{4.21}$$

$$G(P_\mathcal{D}; \mathcal{U}) \approx G(P_{\mathcal{T}_i}; \mathcal{V}_i). \tag{4.22}$$

As each sample $n \in \mathcal{D}$ is predicted exactly one time by the different models $P_{\mathcal{T}_i}$, we can define a single performance estimate by calculating and aggregating G for the combined

results of the k models

$$G\left(P_{\mathcal{D}};\mathcal{U}\right) \approx G\left(P_{\mathcal{T}_1},\ldots,P_{\mathcal{T}_k};\mathcal{V}_1,\ldots,\mathcal{V}_k\right). \tag{4.23}$$

The same analysis follows for any arbitrary repeated out-of-sample validation scheme, including bootstrapping, repeated holdout validation, etc. The only distinction is that with these validation schemes, as opposed to k-fold cross-validation, each instance in the dataset may be present in no test folds, or more than one test fold.

4.2.5 Model optimisation

As discussed in Section 4.2.3 a trained classification model $P_{\mathcal{D}}$ is an instance of a classification algorithm $\mathcal{A}$ fitted to data $\mathcal{D}$. Furthermore, each algorithm has associated *hyper-parameters* (or *utility functions* for RUMs) which control how the model fits to the data during Eq. (4.9) (*model fitting*).

The following discussion focuses on ML hyper-parameter selection (for consistency with Chapter 3). Whilst hyper-parameter selection can be considered as analogous to the utility function specification problem for RUMs, the two tasks are executed using different methodologies. This is due to the fundamental difference between ML hyper-parameters, which are purely mechanistic and not intended to reflect behavioural assumptions, and utility specifications, which are hypothesised from a behavioural foundation. The procedure for RUM utility function specification is discussed in Section 5.2.5.2.

For a particular set of hyper-parameters $\boldsymbol{\lambda}$, we have

$$P_{\mathcal{D}} = \mathcal{A}(\mathcal{D};\boldsymbol{\lambda}). \tag{4.24}$$

Model performance is highly dependent on chosen hyper-parameter values $\boldsymbol{\lambda}$ (Hoos et al. 2014). To enable fair comparison between algorithms, *model optimisation* should be performed to select values of $\boldsymbol{\lambda}$ which minimise the relevant performance metric of the fitted model. We want to select optimal hyper-parameter values $\boldsymbol{\lambda}^{(*)}$ so that

$$\boldsymbol{\lambda}^{(*)} = \arg\min_{\boldsymbol{\lambda}} G\left(\mathcal{A}(\mathcal{D};\boldsymbol{\lambda});\mathcal{U}\right). \tag{4.25}$$

As with Eq. (4.13), as $\mathcal{U}$ is unknown the value of $G(\mathcal{A}(\mathcal{D};\boldsymbol{\lambda});\mathcal{U})$ must be estimated through out-of-sample validation. This validation should be performed on the training data $\mathcal{T}$ only, to prevent *data-leakage* from the test data. To achieve this, we split training data $\mathcal{T}$ into new training and validation sets $\mathcal{T}^*$ and $\mathcal{V}^*$ (or $\mathcal{T}_i^*$ and $\mathcal{V}_i^*$ for repeated cross-validation).

As such, for holdout validation and repeated cross-validation respectively we have

$$\boldsymbol{\lambda}^{(*)} \approx \arg\min_{\boldsymbol{\lambda}} G\left(\mathcal{A}(\mathcal{T}^*;\boldsymbol{\lambda});\mathcal{V}^*\right), \tag{4.26}$$

$$\boldsymbol{\lambda}^{(*)} \approx \arg\min_{\boldsymbol{\lambda}} G\left(\mathcal{A}(\mathcal{T}_1^*;\boldsymbol{\lambda}),\ldots,\mathcal{A}(\mathcal{T}_1^*;\boldsymbol{\lambda});\mathcal{V}_1^*,\ldots,\mathcal{V}_k^*\right). \tag{4.27}$$

Recalling from Section 4.2.3 that $\mathcal{A}(\mathcal{T}^*;\boldsymbol{\lambda})$ is carried out using an optimisation loop, Eqs. (4.25) to (4.27) contain two optimisation loops. The outer loop is where optimal hyper-parameter values are selected, and the inner loop is where the model(s) is/are trained using the set of candidate hyper-parameters.

Due to the nature of inner loop (model fitting) embedded in the hyper-parameter response function, it is difficult to perform the optimisation in Eq. (4.25) using gradient descent methods. Instead, we perform a finite set of trials S and search over candidate values $\boldsymbol{\lambda}' \in S$ to find the best performing set of hyper-parameters $\hat{\boldsymbol{\lambda}}$. For holdout-validation and repeated cross-validation respectively we have

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}' \in S} G\left(\mathcal{A}(\mathcal{T}^*;\boldsymbol{\lambda}');\mathcal{V}^*\right), \tag{4.28}$$

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}' \in S} G\left(\mathcal{A}(\mathcal{T}_1^*;\boldsymbol{\lambda}'),\ldots,\mathcal{A}(\mathcal{T}_1^*;\boldsymbol{\lambda}');\mathcal{V}_1^*,\ldots,\mathcal{V}_k^*\right). \tag{4.29}$$

The selected hyper-parameter values are therefore completely dependent on the explored search-space S. Evaluating the inner loop (model fitting) in Eqs. (4.28) and (4.29) for each candidate set of hyper-parameter values tends to be highly computationally expensive (dependent on the model algorithm). This is because a model (or set of models for repeated cross-validation) must be fitted to the training data and used to predict the validation data for each set of candidate values in S. Practically, this presents an upper limit on the number of trials evaluated in S.

4.2.6 Model selection

Suppose that we have a collection of M different fitted classifiers $P_{\mathcal{D}}^m$, $m = 1,\ldots,M$. We want to choose the most suitable classifier for a particular task.

This requires an assessment of different aspects of the classifier. These include both objective, quantifiable aspects such as the expected predictive performance, and computational cost of fitting and making predictions with the model; as well as subjective aspects, such as the perceived interpretability, reliability, robustness, and complexity of the model.

The subjective aspects are crucial and should be considered thoroughly in the model selection process. However, this section focuses on objective model performance, more specifically how to determine if one model has significantly better predictive performance than another. The subjective aspects of the suitability of the classifiers for mode choice prediction is discussed in Chapter 7.

We consider two classifiers, $P^m_{\mathcal{D}}$ and $P^r_{\mathcal{D}}$, where

$$P^m_{\mathcal{D}} = \mathcal{A}^m(\mathcal{D}; \boldsymbol{\lambda}^m). \tag{4.30}$$

Within this equation $\boldsymbol{\lambda}^m$ represents the optimised hyper-parameters or utility specifications for ML classifiers and RUMs respectively.

We want to investigate whether the candidate model $P^m_{\mathcal{D}}$ does not perform better than the reference model $P^r_{\mathcal{D}}$, i.e. to ascertain whether

$$G\left(\mathcal{A}^m(\mathcal{D}; \boldsymbol{\lambda}^m); \mathcal{U}\right) \leq G\left(\mathcal{A}^r(\mathcal{D}; \boldsymbol{\lambda}^r); \mathcal{U}\right). \tag{4.31}$$

This extends the current practice for comparing ML classifiers.

As discussed in Section 4.2.4, as $\mathcal{U}$ is unknown, the performance of each model can only be estimated using out-of-sample validation.

Consider a single instance of holdout validation of a model P^m on a validation set $\mathcal{V}$. Each element $n \in \mathcal{V}$ has been independently sampled from the population $\mathcal{C}$ with probability

$$\begin{aligned} \Pr(x_n, y_n) &= \Pr(y_n | x_n) \Pr(x_n), \\ &= \Pr(i_n | x_n) \Pr(x_n). \end{aligned} \tag{4.32}$$

where $\Pr(i_n|x_n)$ is the *true probability* of observing the output i_n given a set of features x_n, and $\Pr(x_n)$ is the probability of observing the feature vector x_n in the population $\mathcal{C}$. This analysis accounts for the stochasticity of the validation set due to the ground-truth class labels y_n being drawn from a probability distribution (the true probability). As such, it implies that different outcomes i_n may be generated for the same value of x_n. Any variability or loss of information associated with sampling the finite feature vector x_n from the original feature set $\mathcal{Z}_n$ is captured within this probability distribution. Within this context, the aim of the model P^m is to replicate the true probability distribution $\Pr(i_n|x_n)$.

For each element $n \in \mathcal{V}$ a measure of fit d^m_n is defined which measures how well or how poorly the classifier P^m is able to predict the class label i_n when using x_n as an input

$$d^m_n = d(P^m(x_n), i_n). \tag{4.33}$$

The quantity d_n^m is a random variable. Each realisation corresponds resampling the validation element n with a new individual drawn from the population. The expected value of d_n^m is defined as

$$\mathbb{E}[d_n^m] = \sum_{i=1}^{J} \int d(P^m(x), i) \Pr(i|x) \Pr(x) dx, \tag{4.34}$$

where the integral scans all the possible vectors of features in the population, therefore accounting for the variability of d_n^m due to the sampling of the feature vectors in the validation set (the *sampling noise*), and the summation accounts for the variability of d_n^m due to the class labels y_n themselves being drawn from the unknown probability distribution $\Pr(y_n|x_n)$. The variance of d_n^m can be calculated as

$$\mathrm{Var}[d_n^m] = \mathbb{E}\left[(d_n^m)^2\right] - (\mathbb{E}[d_n^m])^2 . \tag{4.35}$$

The cost function G is defined by aggregating d_n^m over the relevant dataset. For instance, consider taking the arithmetic mean of the element-wise measure of fit d_n^m over the validation set $\mathcal{V}$

$$G(P^m; \mathcal{V}) = \frac{1}{N_\mathcal{V}} \sum_{n=1}^{N_\mathcal{V}} d(P^m(x_n), i_n). \tag{4.36}$$

In this example, the expected value of $G(P^m; \mathcal{V})$ is defined as

$$\begin{aligned} \mathbb{E}[G(P^m; \mathcal{V})] &= \mathbb{E}\left[\frac{1}{N_\mathcal{V}} \sum_{n=1}^{N_\mathcal{V}} d_n^m\right], \\ &= \mathbb{E}[d_n^m], \end{aligned} \tag{4.37}$$

with variance

$$\begin{aligned} \mathrm{Var}[G(P^m; \mathcal{V})] &= \frac{1}{N_\mathcal{V}} \mathrm{Var}\left[\frac{1}{N_\mathcal{V}} \sum_{n=1}^{N_\mathcal{V}} d_n^m\right], \\ &= \frac{1}{N_\mathcal{V}} \mathrm{Var}[d_n^m]. \end{aligned} \tag{4.38}$$

In practice, it is infeasible to calculate the integral in Eq. (4.34), or even to obtain a good approximation of $\Pr(x)$. Additionally, by the nature of the problem, $\Pr(i|x)$ is unknown.

A further source of variability arises from the sampling noise related to the training dataset $\mathcal{T}$. In the above analysis, the model P^m is assumed to be fixed. In reality P^m is fitted to a training set $\mathcal{T}$ for each validation instance

$$P_\mathcal{T}^m = \mathcal{A}^m(\mathcal{T}; \boldsymbol{\lambda}^m). \tag{4.39}$$

As with the sampling noise associated with $\mathcal{V}$, sampling a finite training set $\mathcal{T}$ introduces variability to the estimate of model performance $G(P^m_{\mathcal{T}}; \mathcal{V})$. However, the effects of input sampling noise for model training are complex, and it is not possible to find an analytical solution.

As such, in order to investigate the inequality in Eq. (4.31), the distribution of $G(P_{\mathcal{T}}; \mathcal{V})$ must be estimated for both for P^m and P^r using a numerical solution.

4.3 Systematic critique of existing applications within theoretical framework

The following sections assess the technical limitations of existing ML approaches identified within the systematic review, focusing in turn on datasets, performance estimation, model optimisation, and model selection. The discussion in these sections is focused on mode choice modelling, but it makes use of the general-purpose classification formulation, presented in Section 4.2.

4.3.1 Datasets

Two technical limitations related to datasets are assessed within the theoretical framework: studies not including any attributes of the mode-alternatives, and studies using input features which are dependent on mode choice.

TL1: Studies not including any attributes of the mode-alternatives

The classifier maps the feature vector x_n to a probability distribution $P(x_n)$ on the classes (travel modes in mode choice modelling), as shown in Eq. (4.4) and Eq. (4.6). In order to model the impact that the transport network has on mode choice, it is necessary for x_n to contain attributes of the mode-alternatives, e.g. the expected duration and cost of travelling by each mode in the choice set $i = 1, \ldots, J$.

As significant correlations between attributes of each mode-alternative and mode choice are likely to exist, not including these variables in the feature vector will result in models with lower predictive performance.

Additionally, for statistical RUM models, omitting relevant predictors (features) in the input results in endogenous errors in the parameters of the remaining variables (Train 2009, Chapter 13). This can cause biased, inconsistent estimates of these parameters, which may be important for explaining behaviour (e.g. Value of Time (VoT)).

Finally, when using the choice model for simulation of future trips under unknown conditions (e.g. in an Agent Based Model (ABM)), the impacts of changes to the transport network on mode choice cannot be modelled if attributes of the mode-alternatives are not included in the feature vector.

TL2: Studies using input features which are dependent on mode choice

In order to generate the J-dimensional probability distribution on the classes $P(x_n)$ for a future trip, the feature vector for that trip must be known or estimated in advance. This requires the features in x_n to be independent of the mode selected i_n. If there are features in x_n which are dependent on the mode choice, then mode choice must be known before a prediction is made. This prevents the model from being used in a predictive context.

Input features which are directly and explicitly dependent on class membership, e.g. travel speed being dependent on travel mode, will be highly correlated with the class membership, and so will result in better *apparent performance* of the model than could be achieved using only valid independent variables (i.e. the performance of the model will be overestimated).

As with TL1, including input variables which are dependent on the output in a statistical model (RUM) can introduce endogeneity through reverse causality (Train 2009, Chapter 13). Again, this can cause biased, inconsistent estimates of model parameters.

4.3.2 Performance estimation

Four technical limitations related to performance estimation are assessed within the theoretical framework: studies using inappropriate validation schemes, studies using incorrect sampling methods for hierarchical data, studies not performing external validation, and studies treating choice prediction as a deterministic process.

TL3: Studies using inappropriate validation schemes

In-sample validation uses the same data $\mathcal{D}$ to fit and validate the model and can be interpreted as using the *train-error* to estimate model performance. As such, it presents only the explanatory power of the model, i.e. the ability of the model to replicate the training data, and not the predictive performance. This is discussed in detail by Shmueli (2010).

If a model has high *variance*, it can overfit to noise in the data during model fitting, without generalising to valid correlations between x_n and y_n. This will result in in-sample validation overestimating the predictive performance. Without testing the model on out-of-sample data, there is no way to assess whether overfitting has occurred.

Additionally, due to the nature of the *bias-variance tradeoff* (Hastie, Friedman, and Tibshirani 2008, Chapter 2), a classifier will tend to fit partially to noise in the data, even if it does not overfit. As such, the train-error will tend to overestimate predictive performance, even for well specified models which do not overfit.

Validating a model on unseen data is an essential step in predictive modelling, and as such, in-sample validation is an inappropriate validation scheme.

Furthermore, in order to make valid comparisons between performance estimates of different models, the same validation method must be used for all models. Otherwise, any apparent differences in performance may be due to differences in the respective validation schemes.

TL4: Studies using incorrect sampling methods for hierarchical data

In order for Eq. (4.17)

$$G(P_{\mathcal{D}}; \mathcal{U}) \approx G(P_{\mathcal{T}}; \mathcal{V}).$$

and Eq. (4.23)

$$G(P_{\mathcal{D}}; \mathcal{U}) \approx G(P_{\mathcal{T}_1}, \ldots, P_{\mathcal{T}_k}; \mathcal{V}_1, \ldots, \mathcal{V}_k).$$

to be valid estimates of model performance on new data, the correlation between $\mathcal{T}$ and $\mathcal{V}$ (or $\mathcal{T}_i$ and $\mathcal{V}_i$) must be equivalent to the correlation between $\mathcal{D}$ and $\mathcal{U}$.

Mobility datasets are hierarchical in nature; elements of the dataset are grouped by hierarchies, from which they inherit correlated features. These hierarchies can cause valid correlations which are relevant to the modelling scenario. For example, a modeller would be interested if students (socio-economic group) show a tendency towards cycling (correlation), or if trips made at the weekend (temporal grouping) were less likely to be made by public transport (correlation). In both these cases, the hierarchies (groups) are general, and described by information in the feature vector x_n. As such, these correlations are likely to be constant across $\mathcal{D}$ and $\mathcal{U}$, and so are relevant to the modelling scenario.

Many datasets used in the literature include hierarchies which are specific to the dataset (e.g. Household-Person, Person-Tour, and Tour-Trip, see Section 3.3.3). These hierarchies are not constant across $\mathcal{D}$ and $\mathcal{U}$ (i.e. the individuals and households in $\mathcal{D}$ are different from those in $\mathcal{U}$). As models do not *profile* individuals or households in the population, these hierarchies are not relevant to the modelling scenario.

Sampling trips randomly from $\mathcal{D}$ for hierarchical datasets (*trip-wise* sampling) allows trips made by the same individual or household to be split across both $\mathcal{T}$ and $\mathcal{V}$ (or $\mathcal{T}_i$ and $\mathcal{V}_i$). As these trips are not independent and identically distributed (i.i.d.), this introduces correlations between $\mathcal{T}$ and $\mathcal{V}$ (or $\mathcal{T}_i$ and $\mathcal{V}_i$) which will not be present between $\mathcal{D}$ and $\mathcal{U}$. This

will result in $G(P_{\mathcal{T}}; \mathcal{V})$ (or $G(P_{\mathcal{T}_i}; \mathcal{V}_i)$) overestimating the model performance on unknown data: $G(P_{\mathcal{D}}; \mathcal{U})$.

TL5: Studies not performing external validation

Any dataset $\mathcal{D}$ may contain systematic correlations between the input x_n and class labels y_n introduced during data collection. These correlations are specific to that dataset, and so not general to the population $\mathcal{C}$. They can arise from (i) sampling bias/recording errors when sampling the dataset $\mathcal{D}$ from the population $\mathcal{C}$, (ii) sampling bias/recording errors when sampling the finite feature vector x_n from the feature set $\mathcal{Z}_n$, or (iii) data-specific hierarchies which are not addressed in the sampling method (as in TL4). If these correlations are present, a model with high variance may overfit to them. If this is the case, and the validation data is sampled from the same dataset as the train data, the out-of-sample validation will be positively biased compared to the real-world performance.

Validating the model on data $\mathcal{V}$ collected separately from the training data $\mathcal{T}$ (external validation) reduces the possibility of these correlations being present across both the train and test data. This will therefore aid in preventing overestimation of model performance. The validation data could be either for a different time period, different geographical area, or collected using a different methodology.

TL6: Studies using only discrete metrics

In order to quantify the overall performance of a model, it is necessary to define a performance metric or aggregate measure of fit G. The predominant approach in the ML literature is to use discrete metrics based on the *confusion matrix*.

The probabilistic classifier P generates a probability for each class i. A discrete prediction $\hat{i}_n$ can be obtained by selecting the class with the maximum probability for each trip

$$\hat{i}_n = \arg\max_i P(i|x_n)\,. \tag{4.40}$$

As in Eq. (4.3), $\hat{y}_{in} \in \{0,1\}$ are set of class indicators corresponding to $\hat{i}_n$.

The confusion matrix is a $J \times J$ matrix based on the discrete predictions, where the rows represent the *ground-truth* class i_n, the columns represent the discretised prediction $\hat{i}_n$, and each entry in the matrix gives the total number of elements in the dataset with the corresponding ground-truth and predicted classes. From this, we can determine for each class i the true positives (TP_i), true negatives (TN_i), false positives (FP_i), and false negatives (FN_i). These

are defined respectively as

$$\mathrm{TP}_i = \sum_{n=1}^{N} y_{in}\hat{y}_{in}, \tag{4.41}$$

$$\mathrm{TN}_i = \sum_{n=1}^{N} (1-y_{in})(1-\hat{y}_{in}), \tag{4.42}$$

$$\mathrm{FP}_i = \sum_{n=1}^{N} (1-y_{in})\hat{y}_{in}, \tag{4.43}$$

$$\mathrm{FN}_i = \sum_{n=1}^{N} y_{in}(1-\hat{y}_{in}). \tag{4.44}$$

A number of metrics can be calculated from the confusion matrix.

The most commonly used metric in the literature, accuracy (also called zero-one score, classification rate), is given by the overall rate of correct predictions

$$G_{\text{accuracy}} = \frac{\sum_{i=1}^{J} \mathrm{TP}_i}{N}. \tag{4.45}$$

Recall is, for a mode i, the true positives of that mode divided by the total occurrences of that mode N_i (the combined total of true positives and false negatives)

$$G_{\text{recall},i} = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}. \tag{4.46}$$

Finally, precision is the number of true positives for a mode divided by the number of predicted positives

$$G_{\text{precision},i} = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i}. \tag{4.47}$$

There are a number of issues with the use of discrete metrics for choice prediction. Firstly, discrete classification is likely to result in non-representative mode-shares. Mode choice data is inherently imbalanced, i.e. there are likely more trips made by some modes (e.g. car, walking) than others (e.g. cycling). By assigning each prediction to the class with highest probability, the less frequent classes will be under-represented in the predicted outcomes, and the more frequent classes will be over-represented. For example, consider a biased random coin flip, where heads is 60 % likely to occur, and tails occurs with 40 % probability. The best possible predictive classification model will predict these outcomes at their respective probabilities for each coin flip event. However, discretising the prediction will result in heads always being predicted (and never tails) as heads is always more likely than tails. This clearly

results in non-representative class shares. Non-representative mode-shares are unacceptable for mode choice models, where the mode-shares are a crucial model output.

Similarly, by generating a discrete class for each observation, mode choice is treated as a deterministic instead of a stochastic process. As such, it is assumed that mode choice is constant under the same set of conditions and socio-economic characteristics. In reality, passengers have a degree of intra-heterogeneity, and their choice can be considered as being drawn randomly from a probability distribution (as with the coin-flip example). We define this distribution as the *true model*, which we aim to replicate with the classification model. In order to account for this stochastic heterogeneity in simulation, the predicted mode choice should be drawn from a probability distribution. The metric used to assess model performance should therefore represent how well the predicted probability distributions fit the data.

Additionally, discrete metrics do not assess how right or wrong model predictions are. For example, when using discrete metrics, the contribution to the model's score for a trip where the classifier predicts the selected mode at 1 % probability is the same as that for a trip where the classifier predicts the selected mode at 49 % probability. Analysing the probability distributions presents information on where the model performs well or poorly.

Finally, by taking the maximum of the class probabilities in Eq. (4.40), discrete predictions and the associated metrics are discontinuous. This results in discrete metrics having an expected score which is not differentiable or strictly convex. Additionally, accuracy and other discrete metrics are not *strictly proper* scoring rules, and as such do not have unique maximums (Gneiting and Raftery 2007). This makes discrete metrics poor metrics to use during model fitting, particularly where a continuous gradient is required (e.g. gradient descent).

4.3.3 Model optimisation

Three technical limitations related to model optimisation are assessed within the theoretical framework: studies not performing any type of hyper-parameter optimisation, studies not using rigorous hyper-parameter search schemes, and studies optimising hyper-parameters on test data.

TL7: Studies not performing any type of hyper-parameter optimisation

As discussed in Section 4.2.5, model performance is highly dependent on chosen hyper-parameter values λ. Additionally, optimal hyper-parameter values are highly task dependent, and will vary for different datasets, metrics, scenarios, etc. Using default hyper-parameter values, or values from previous studies with different modelling scenarios or data, limits

the search-space S in Eq. (4.28) and Eq. (4.29) to a single value. As such, optimal hyper-parameter values are highly unlikely to be identified, and the resultant model will perform worse than the optimised model.

If the hyper-parameters of each classifier have not been optimised, it is not possible to make valid comparisons between the respective algorithms, as any difference in model performance may be due to better hyper-parameter values selected for one algorithm than another.

TL8: Studies not using rigorous hyper-parameter search schemes

Similarly to TL3 and TL7, for valid comparisons to be made between models, all hyper-parameters for each classifier should be optimised. Optimising only the parameters for only a subset of classifiers being compared will tend to improve the performance of those classifiers over those which have not been optimised.

Additionally, the search space S should cover all dimensions of the hyper-parameter space, otherwise optimal values are unlikely to be found. Whilst certain hyper-parameters may have little/no effect on model performance, there is no way to determine this unless they are tested.

Finally, search schemes should be used which maximise the probability of finding optimal hyper-parameters in an unbiased manner. The predominant approaches for hyper-parameter selection in the literature (where it is performed) are to use either a manual search/trial-and-error, or grid-search.

The primary advantages of manual search are its simplicity and the ability to use the modeller's intuition (from previous trials and similar classification tasks) to influence subsequent guesses. However, manual search presents both high potential for the introduction of bias, and difficulty in reproducing results. Additionally, as the search is manual and cannot be parallelised, it practically limits the modeller to a small number of trials in S.

Grid-search predefines a set of candidate values for each hyper-parameter and use them to define a search space S containing each unique combination of values. Grid-search can be both automated and parallelised, and therefore enables a greater set of candidate values to be searched than with a manual search. However, grid-search is unable to learn from previous evaluations, and so spends a lot of time evaluating candidate values which are unlikely to perform well. Additionally, the same values for each hyper-parameter are repeated for each dimension of the search, limiting the likelihood of evaluating the optimal value for each hyper-parameter. As such, grid-search is highly inefficient for hyper-parameter selection and has been shown to perform poorly in practice at finding optimal hyper-parameter values compared to other search schemes, including random search (Bergstra and Bengio 2012).

TL9: Studies optimising hyper-parameters on validation data

Fitting hyper-parameters to the holdout validation data allows the model to select optimal hyper-parameters specifically for that data. In other words, this presents the potential for the model to fit to the validation data using the hyper-parameters. This can be seen by substituting in $\mathcal{T}^* = \mathcal{T}$ and $\mathcal{V}* = \mathcal{V}$ into Eq. (4.28)

$$\hat{\lambda} \approx \underset{\lambda' \in S}{\arg\min}\, G\left(\mathcal{A}(\mathcal{T}); \mathcal{V}\right). \tag{4.48}$$

In Eq. (4.48), the selected hyper-parameters are those that minimise the cost function of the trained model on the holdout validation data $\mathcal{V}$. This will upward bias the performance estimate over that which would be achieved with previously unseen data. This is explored by Varma and Simon (2006), who show that cross-validation provides an upward biased estimate of true performance if it is used for model optimisation.

As discussed in TL3, validating a model on previously unseen data is an essential step in predictive modelling. Holdout validation data should not be seen by the model at any time during model development (including hyper-parameter optimisation) until the testing of the finished model.

4.3.4 Model selection

One technical limitation related to model optimisation is assessed within the theoretical framework: studies not analysing uncertainty in performance estimates.

TL10: Studies not analysing uncertainty in performance estimates

As shown in Eq. (4.37), each evaluation of model performance on a validation sample (whether through holdout validation or repeated cross-validation) is a random variable, with three associated sources of variability: (i) the sampling noise associated with sampling a finite validation sample $\mathcal{V}$ from the population $\mathcal{C}$, (ii) the variability due to the class labels y_n being drawn from an unknown probability distribution $\Pr(y_n|x_n)$, and (iii) the sampling noise associated with training the classifier P on a finite training sample $\mathcal{T}$.

If the distributions of the performance estimates are not accounted for, any apparent differences between different classifiers' performance estimates may be due to this noise.

4.4 Summary

This chapter introduces the theoretical structure and framework for probabilistic classification and conducts a systematic critique of the existing practices. A general theoretical framework for supervised classification is established which applies to both ML and statistical RUMs. The framework combines existing analysis in the literature with new concepts and analysis developed for this thesis. The framework focuses on six stages in the classification problem: (i) model dataset, (ii) model prediction, (iii) model fitting, (iv) model performance estimation, (v) model optimisation, and (vi) model selection. A uniform notation is used to present the framework. Each of the technical limitations identified in Chapter 3 is analysed within this theoretical framework, providing a qualitative assessment of their potential impacts on the findings of the studies. All of the limitations are shown to have material implications for experimental results.

The theoretical framework presented in this chapter is used to establish the methodological approach for the thesis in Chapter 5. This approach specifically addresses each methodological limitation, demonstrating how it is solved within this research.

Chapter 5

Modelling methodology

5.1 Overview

This chapter introduces the modelling methodology used within this thesis to investigate travel mode choice. This chapter aims to (a) establish a methodological approach within the context of the theoretical framework (Chapter 4) which addresses the technical limitations identified in the systematic review (Chapter 3), (b) specify a formal modelling framework for both statistical Random Utility Models (RUMs) and Machine Learning (ML) classifiers, and (c) present the planned investigations into random utility, ML, and assisted specification models of passenger mode choice.

Firstly, Section 5.2 presents the methodological approach within the context of the framework presented in Chapter 4. The section refers to each technical limitation identified in Chapter 3, explaining how it is addressed within the methodology. Next, Section 5.3 formalises the specific details of the modelling framework implemented for this study which is common to both ML and RUM classifiers.

Section 5.4 introduces the RUM investigations carried out within the thesis. These investigations represent the application of state-of-practice techniques to the new data developed for this thesis and serve as a performance benchmark for the remaining classification techniques. The ML-based investigations are presented in Section 5.5. These investigations address the technical limitations of the existing ML research, and as such represent a rigorous and systematic study of ML classification techniques for mode-choice prediction. Finally, an assisted specification approach, where a fitted ML classifier is used to inform the utility function structure in a RUM, is proposed in Section 5.6. This approach attempts to combine the predictive power and flexibility of ML classifiers, with the interpretability and strong behavioural foundation of the random utility approach.

5.2 Methodological approach

This section presents the methodological approach within the thesis, within the context of the theoretical framework presented in Chapter 4. The general approach is to train probabilistic classifiers P to predict the likely travel mode choices made by passengers for a dataset of historic trips $\mathcal{D}$. Four modes are considered in the choice-set, so that there are $J = 4$ classes: walking ($i = 1$), cycling ($i = 2$), public transport ($i = 3$), and driving ($i = 4$).

The following sections present the elements of the modelling framework in detail using the same structure as the formulation of the classification problem in Section 4.2: (i) model dataset, (ii) model prediction, (iii) model fitting, (iv) model performance estimation, (v) model optimisation, and (vi) model selection.

A summary of how each technical limitation identified in Chapter 3 is addressed is given in Table 5.1. Details of each solution are given within the relevant sections.

Table 5.1 Solutions for each technical limitation identified in systematic review.

No.	Topic	Description	Solution
TL1	Datasets	Studies not including any attributes of the mode-alternatives	Framework to augment data with details of mode-alternative routes
TL2	Datasets	Studies using input features which are dependent on output choice	Data generation independent of mode choice
TL3	Performance estimation	Use of inappropriate validation schemes	Use OOS validation schemes (holdout, k-fold, bootstrapping)
TL4	Performance estimation	Studies using incorrect sampling methods for hierarchical data	Use household-wise sampling
TL5	Performance estimation	Studies not testing models on data collected separately from the training data	Holdout validation on separate (future) year of data
TL6	Performance estimation	Studies treating choice prediction as a deterministic process	Use probabilistic classification
TL7	Model optimisation	Studies not performing any type of hyper-parameter optimisation	Perform hyper-parameter optimisation for all classifiers
TL8	Model optimisation	Studies not using rigorous hyper-parameter search schemes	Use sequential optimisation for all hyper-parameter values
TL9	Model optimisation	Studies optimising hyper-parameters on test data	Optimise hyper-parameters using train data only
TL10	Model selection	Studies not analysing uncertainty in performance estimates	Estimate uncertainty in performance estimates using bootstrapping

An example from the study dataset is used to illustrate a conceptual overview of the methodology.

5.2.1 Model dataset

A new data generation framework for recreating choice-sets faced by a passenger at the time of travel has been developed for this thesis. The framework is used to build a dataset combining individual records from the London Travel Demand Survey (LTDS) with systematically matched trip trajectories alongside their corresponding mode-alternatives (i.e. the choice-set faced by the passenger at the time of travel) from an online directions service, and precise estimates of public transport fares and Vehicle Operating Costs (VOCs). The framework and dataset are presented in detail in Chapter 6.

Figure 5.1 depicts typical trip *n* from the dataset, of a journey made between Ilford and Wood Green, departing at 14:30 on a Tuesday in October 2014 (the details of the trip have been modified to preserve anonymity). Additional features in the base dataset include the journey purpose (employer's business); straight line distance between origin and destination (13.158 km); individual profile (38 years old, male, has driving licence, no discounts for rail or bus fares); and household car ownership (less than one car per adult in household, diesel car fuel type).

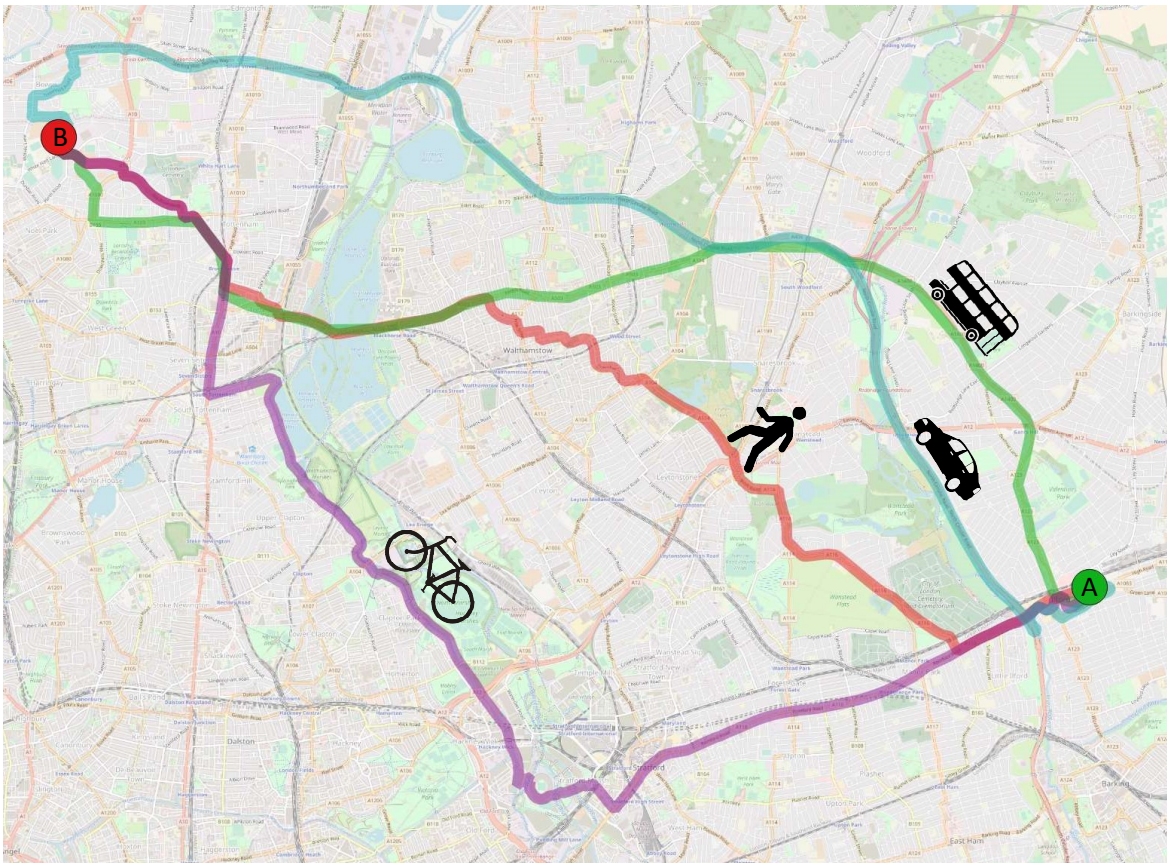


Fig. 5.1 Mode-alternative choice-set for example trip from A to B from study dataset

The data-generation framework additionally allows for details of the mode-alternatives to be determined for the full dataset, including the walking duration (3.294 h); cycling duration (1.056 h); public transport details (total duration of 1.435 h, comprising of 0.302 h of access/ egress and 1.133 h of on-board bus, no interchanges); public transport fare (£1.50); driving duration (0.473 h); driving costs (VOC of £2.04, no congestion charge); and road traffic variability (37.6 %). Together, these attributes form the feature vector x_n. The trip is actually made by driving, so that $y_n = (0,0,0,1)$ and $i_n = 4$.

TL1: Studies not including any attributes of the mode-alternatives

The data generation framework presented in Chapter 6 provides details of the choice-set faced by the passenger at the time of travel (mode-alternative attributes), out of walking, cycling, public transport, and driving. This allows the impacts of these attributes on mode choice to be modelled. This process also enables RUMs with fully specified alternative specific utility functions to be fitted to the data and compared to ML techniques.

The impact of adding mode-alternative variables is investigated experimentally (see Section 5.5.3).

TL2: Studies using input features which are dependent on mode choice

The synthesised features included in the dataset are generated using the same methodology for all modes, including the selected mode. The routes are generated using the original trip departure time, and duration information is not included in the feature vector, either explicitly or implicitly. As such, none of the input features are dependent on the mode choice.

Additionally, the framework presented in Chapter 6 allows for the generation of data for future Origin-Destination (O-D) pairs, given a socio-economic profile, trip purpose, and departure time. This allows the dataset methodology to be used as part of a predictive modelling framework.

5.2.2 Model prediction

All of the classifiers used in this thesis are capable of outputting probabilities, (or *probability-like* values). They map the feature vectors x_n to a probability distribution $P(x_n)$ over the four transport modes.

For example, the predicted probabilities generated by a simple RUM classifier for the example trip are shown in Table 5.2.

5.2.3 Model fitting

As discussed in Chapter 4, each classifier in the study is an example of an algorithm $\mathcal{A}$ fitted to a dataset $\mathcal{D}$. The nature of the mapping from x_n to $P(x_n)$, and how it is fitted to the data, is dependent on the model algorithm $\mathcal{A}$.

A simple (Multinomial Logit (MNL)) RUM classifier is used as an example to illustrate the concept (an ML based classifier could similarly be used). The RUM model can be defined using the following utility function for each mode

$$V_{in} = \beta_{0,i} + \beta_1 D_{in} + \beta_2 C_{in}, \tag{5.1}$$

where D_{in} and C_{in} are the duration and cost of trip n for mode i. Within this utility function $\beta_{0,i}$ represents the Alternative Specific Constant (ASC) for mode i, and β_1 and β_2 are the generic parameters for duration and cost respectively. The choice probabilities can then be calculated from the *softmax* formula

$$P(i|x_n) = \frac{e^{V_{in}}}{\sum_{j=1}^{J} e^{V_{jn}}}. \tag{5.2}$$

During Maximum Likelihood Estimation (MLE) (fitting) of the model to an input dataset, the values of the parameters are estimated as shown in Table 5.2 (the ASC for walking is fixed at zero). These parameters can then be used to work out the utilities, and therefore choice probabilities, for each mode for each trip (*model prediction*). These values are shown in Table 5.2 for the example trip introduced in Section 5.2.1.

Table 5.2 Parameter estimates for simple RUM with relevant variables and predicted utilities and probabilities for example trip.

Parameter	**Value**	**Variable**	**Walking**	**Cycling**	**PT**	**Driving**
$\beta_{0,1}$	0	Walking dummy	1	0	0	0
$\beta_{0,2}$	-3.85	Cycling dummy	0	1	0	0
$\beta_{0,3}$	-0.496	PT dummy	0	0	1	0
$\beta_{0,4}$	-1.2	Driving dummy	0	0	0	1
β_1	-5.4	Duration (D_{in})	3.294	1.056	1.435	0.473
β_2	-0.176	Cost (C_{in})	0	0	1.5	2.04
		V	-17.788	-9.552	-8.509	-4.113
		$P(i\|x_n)$	1.133×10^{-6}	0.004 272	0.012 13	0.9836

5.2.4 Model performance estimation

Three different validation techniques are used to estimate model performance. Each technique serves a different purpose within the study. Ten-fold cross-validation is used during ML hyper-parameter selection, in order to determine the optimal model hyper-parameters for each algorithm for the mode choice prediction task (see Section 5.2.5.1). Holdout validation of a future year of trips is used to assess how well the optimised model will likely perform on new (future) data. Finally, *bootstrap sampling* (bootstrapping) is used to estimate the distribution of the performance estimate, in order to identify the best performing models for the task (see Section 5.2.6).

Cross-validation folds and bootstrap samples are formed using *household-wise* sampling, where the trips are sampled grouped by household. This prevents trips made by the same individual or members of the same household from appearing in both the training and validation data, and as such ensures no additional correlations are introduced between $\mathcal{T}$ and $\mathcal{V}$. For all three validation techniques, the same train-test splits/folds are used across all models, so that results can be directly compared between them.

In order to emulate the use case of predicting future trips, the dataset is divided by survey year into a training set (first two years of data: April 2012-March 2014) and a holdout test set (final year of data: April 2014-March 2015). The training set is used for model optimisation, cross-validation performance estimation, and training the optimised test model whilst the holdout test set is reserved for performance evaluation of the optimised test models. The bootstrap sampling is performed on all of the data simultaneously.

Three metrics are used to assess model performance: (i) Discrete Classification Accuracy (DCA); (ii) Arithmetic Mean Probability of Correct Assignment (AMPCA); (iii) Geometric Mean Probability of Correct Assignment (GMPCA). GMPCA is the primary performance metric, with AMPCA and DCA provided as additional metrics. The Cross-Entropy Loss (CEL) is additionally used for statistical significance tests and as the objective function for the hyper-parameter search

Discrete Classification Accuracy (DCA) provides the number of correct assignments if each prediction is assigned to the highest probability class. The full DCA equation can be obtained by substituting Eq. (4.41) into Eq. (4.45)

$$G_{\text{DCA}} = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{J} y_{in} \hat{y}_{in}, \tag{5.3}$$

where

$$\hat{y}_{in} = \begin{cases} 1 & \text{when } i = \arg\max_i P(i|x_n), \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

As discussed in Section 4.3.2, DCA is a discrete metric, which has a number of related limitations. However, it is included both for comparison with previous studies' results and due to its ease of interpretation.

The Arithmetic Mean Probability of Correct Assignment (AMPCA) provides the expected number of correct assignments if mode choices are drawn randomly from (i.e. simulated from) the predicted probability distributions. It is calculated as

$$\begin{aligned} G_{\text{AMPCA}} &= \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{J} y_{in} P(i|x_n), \\ &= \frac{1}{N}\sum_{n=1}^{N} P(i_n|x_n). \end{aligned} \tag{5.5}$$

where i_n denotes the index of the class associated with element n. AMPCA is a continuous probabilistic metric, which directly analyses the predicted choice probabilities for the selected mode. It therefore addresses many of the limitations related to accuracy/error rate raised in Section 4.3.2. However, AMPCA still suffers from the *accuracy paradox*, where a trivial classifier can achieve low loss for imbalanced data by predicting the most likely mode. This is because it calculates the arithmetic mean of the choice probabilities, which reflects *absolute* (and not *relative*) differences in their values. For example, increasing a single assigned choice probability from 1 % to 5 % has the same impact on the AMPCA (increasing it by $0.04/N$), as increasing an assigned choice probability from 95 % to 99 %. As the event with probability 95 % is more likely to occur than that with a probability of 1 %, this results in AMPCA rewarding a model more for increasing the choice probabilities of the more frequent class.

The Geometric Mean Probability of Correct Assignment (GMPCA) provides a robust measure of the *average correctness* of the model, accounting for relative differences in the choice probabilities. It can be calculated as

$$\begin{aligned} G_{\text{GMPCA}} &= \left(\prod_{n=1}^{N}\left(\sum_{i=1}^{J} y_{in} P(i|x_n)\right)\right)^{\frac{1}{N}}, \\ &= \left(\prod_{n=1}^{N} P(i_n|x_n)\right)^{\frac{1}{N}} \end{aligned} \tag{5.6}$$

In order to understand the GMPCA, and what makes it a suitable performance metric, it helps to consider the total joint likelihood of the data given the model, which can be calculated (assuming the data are generated independent and identically distributed (i.i.d.)) as

$$\begin{aligned} G_{\text{likelihood}} &= \prod_{n=1}^{N}\prod_{i=1}^{J} P(i|x_n)^{y_{in}}, \\ &= \prod_{n=1}^{N} P(i_n|x_n). \end{aligned} \tag{5.7}$$

In practice, due to numerical issues when calculating the joint likelihood directly, it is more convenient to consider the log-likelihood, which is the natural logarithm of the likelihood

$$\begin{aligned} G_{\text{log-likelihood}} &= \ln\left(\prod_{n=1}^{N}\prod_{i=1}^{J} P(i|x_n)^{y_{in}}\right), \\ &= \sum_{n=1}^{N}\sum_{i=1}^{J} y_{in}\ln P(i|x_n), \\ &= \sum_{n=1}^{N} \ln P(i_n|x_n). \end{aligned} \tag{5.8}$$

Absolute differences in log-likelihood indicate relative model performances. For example, an increase of three in the log-likelihood score represents a relative increase of the joint likelihood of the data of $e^3 = 20$ times.

Log-likelihood is the metric traditionally used for RUM investigations, where the modeller wants to find the model which most accurately explains a given dataset (and so is dependent on the size of the dataset). The performance metrics used in ML have a different purpose, which is to identify how well a model will perform on unseen data (of unspecified size). As such, the CEL is typically used in ML applications, which is the normalised (average) negative log-likelihood

$$\begin{aligned} G_{\text{CEL}} &= -\frac{1}{N} G_{\text{log-likelihood}}, \\ &= -\frac{1}{N}\sum_{n=1}^{N} \ln P(i_n|x_n). \end{aligned} \tag{5.9}$$

Minimising the CEL is equivalent to maximising the overall probability of the observed choices given the model (i.e. a smaller CEL represents a better model fit). As such, using CEL as the objective function during model training results in MLE.

Whilst CEL is a robust objective function which does not suffer from the accuracy paradox, it is difficult to interpret. Additionally, both absolute and relative differences in CEL

appear to have little physical meaning. This is addressed by taking the negative exponential of CEL, which gives the GMPCA

$$\begin{aligned}\exp\left(-G_{\text{CEL}}\right) &= \exp\left(\frac{1}{N}\sum_{n=1}^{N}\ln P(i_n|x_n)\right),\\ &= \left(\prod_{n=1}^{N}P(i_n|x_n)\right)^{\frac{1}{N}},\\ &= G_{\text{GMPCA}}.\end{aligned} \tag{5.10}$$

The geometric mean reflects *relative* differences in choice probabilities. Repeating the previous example, increasing a single assigned choice probability from 1 % to 5 % increases the GMPCA by 4.8 times more than increasing an assigned choice probability from 95 % to 99 % ($\frac{0.01}{0.05}/\frac{0.95}{0.99} = 4.80$). As such, GMPCA does not suffer from the *accuracy paradox*.

The advantage of the GMPCA over CEL is that the GMPCA has a clear physical interpretation as a robust measure of the *average correctness* of the model. A GMPCA of one indicates a perfect (completely correct) classifier, and a GMPCA of zero indicates a completely incorrect classifier. Note that if any *single* choice probability predicted by the model is zero, the GMPCA will also be zero, as this would imply the data are impossible given the model. Relative differences in the GMPCA directly represent relative changes in the underlying predicted probabilities. For example, a relative increase in the GMPCA of 10 % (i.e. 1.1 times greater), would indicate a relative increase in the total likelihood of 1.1^n (where n is the number of observations), i.e. a relative change of 10 % across the individual choice probabilities.

Additionally, the GMPCA behaves similarly to the AMPCA and DCA in that it is in the range [0,1], and a higher score indicates better performance. The GMPCA is bound between zero and the AMPCA, with the two being equal only when $P(i_n|x_n)$ is identical for all n. Lastly, it is simple to calculate benchmarks for the GMPCA, including a uniform prior ($1/J$ where 1 is the number of classes), and a balanced prior ($\prod_i r_i^{r_i}$ where r_i is the ratio of class i).

The example trip introduced previously is from October 2014, and as such is in the holdout test set. The DCA, AMPCA, GMPCA, and CEL can be calculated for the RUM presented in Table 5.2 when using this single trip as the validation set (so that $N = 1$). Note that, as there is only one observation, the arithmetic mean and geometric mean are both equal.

From the table, $P(x_n) = (1.13\times10^{-6}, 4.27\times10^{-3}, 0.0121, 0.984)$. The trip is actually made by driving, so that $y_n = (0,0,0,1)$ and $i_n = 4$. Discrete classification predictions $\hat{y}_n$ can

be obtained from $P(x_n)$ using Eq. (5.4): $\hat{y}_n = (0,0,0,1)$ (so that $\hat{i}_n = 4$). Therefore

$$\begin{aligned} G_{\text{DCA}} &= \hat{y}_{4,n} \\ &= 1, \\ G_{\text{AMPCA}} &= G_{\text{GMPCA}} \\ &= P(4|x_n) \\ &= 0.9836, \\ G_{\text{CEL}} &= -\ln P(4|x_n) \\ &= 0.0165. \end{aligned}$$

Had the journey actually been made by public transport, so that $i_n = 3$, the DCA would be 1, the AMPCA and GMPCA would be 0.0121, and the CEL would be 4.4123.

TL3: Studies using inappropriate validation schemes

Three different validation techniques are used in the methodology. All three schemes involve validating the performance of a model on data unseen during the model training (out-of-sample validation). The same validation methods and splits/folds are used across all models, so that results can be directly compared between them.

The impacts of using inappropriate validation schemes are investigated experimentally (see Section 5.5.1).

TL4: Studies using incorrect sampling methods for hierarchical data

This research uses grouped household-wise sampling as an alternative to random trip-wise sampling for hierarchical data.

The impacts of using incorrect sampling methods are investigated experimentally (see Sections 5.5.2 and 6.4.2).

TL5: Studies not performing external validation

A separate year of data is withheld to test optimised model performance, which therefore represents external validation. By testing the model on a holdout sample from a future year, the use case for a mode-choice model of predicting future trips is emulated.

The impacts of not testing models on data collected separately from the training data are explored experimentally (see Section 5.5.2).

TL6: Studies using only discrete metrics

Three different metrics are used to analyse model performance, including two probabilistic metrics (GMPCA and AMPCA).

The impacts of using discrete classification on predicted mode-shares is explored experimentally (see Section 5.5.1).

5.2.5 Model optimisation

Model optimisation is the only step of the methodological approach which is performed differently between ML models and RUMs within the study. This is due to the fundamental differences between ML hyper-parameters, which are purely mechanistic and not intended to reflect behavioural assumptions, and utility specifications, which are hypothesised from a behavioural foundation. This section is therefore divided into the ML and RUM approaches.

5.2.5.1 Machine learning approach - hyper-parameter selection

Hyper-parameter selection for the ML classifiers is performed using Sequential Model-Based Optimisation (SMBO). As discussed in Section 4.2.5, hyper-parameter selection cannot be performed using traditional optimisation algorithms. This is due the embedded model fitting loop in the hyper-parameter optimisation problem (Eq. (4.25) from Section 4.2.5)

$$\boldsymbol{\lambda}^{(*)} = \arg\min_{\boldsymbol{\lambda}} \mathbb{E}\left[G\left(\mathcal{A}(\mathcal{D};\boldsymbol{\lambda});\mathcal{U}\right)\right].$$

SMBO solves this problem by approximating this inner loop with a surrogate *hyper-parameter response function* Ψ, so that

$$\boldsymbol{\lambda}^{(*)} \approx \arg\min_{\boldsymbol{\lambda}} \Psi(\boldsymbol{\lambda}). \tag{5.11}$$

Subsequent candidate values for $\boldsymbol{\lambda}$ are selected according to their expected performance in Ψ, which in turn is continually updated from the results of each trials. This creates an iterative process. SMBO has been shown to outperform other methods of hyper-parameter selection, including manual search, grid search and random search (Snoek, Larochelle, and Adams 2012; Bergstra, Komer, et al. 2014).

There are several existing approaches to SMBO, including Gaussian Processes (GPs) (Rasmussen 2004) and Tree-structure Parzen Estimators (TPEs) (Bergstra, Bardenet, et al. 2011). This study makes use of the TPE approach, which Bergstra, Bardenet, et al. (2011) find performs favourably compared to GPs on two complex hyper-parameter optimisation tasks.

All of the ML classifiers are optimised using the same procedure. A prior distribution is defined for each model hyper-parameter, which together define the hyper-parameter search space Λ. Distributions can be continuous (uniform, log-uniform, normal); discrete (uniform integer, log-uniform integer); categorical (uniform choice, probabilistic choice); or hybrid (combination of categorical with uniform/discrete), depending on the nature of the corresponding hyper-parameter.

One-hundred iterations of hyper-parameter optimisation are performed. The first 20 iterations use random-search, with candidate values for each hyper-parameter drawn randomly from the respective prior distribution in Λ. The results of the 20 random-search iterations are used to estimate the initial response function Ψ. The following 80 iterations use the TPE algorithm, with each set of candidate hyper-parameter values λ' drawn from Ψ, which in turn is updated after each draw.

Model performance for each set of candidate hyper-parameter values λ' is estimated using 10-fold cross-validation. This results in 1000 train-test instances of each model (10 folds for 100 iterations). The same validation fold splits are used for each iteration for each model. The folds are sampled grouped household-wise, and are stratified by travel mode, so that each of the 10-folds has equal ratios of trips made by each mode. The models are optimised using only the training data (first two years of data).

Model performance is determined by the weighted average CEL over the 10 folds, $G_{k\text{-fold}}$, given by

$$G_{k\text{-fold}} = \frac{\sum_{k=1}^{K} N_{\mathcal{T}_k} G_{\text{CEL}}\left(P_{\mathcal{T}_k,\lambda'}; \mathcal{V}_k\right)}{\sum_{k=1}^{K} N_{\mathcal{T}_k}}. \tag{5.12}$$

The optimal hyper-parameter values are selected which minimise $G_{k\text{-fold}}$ out of the 100 iterations.

5.2.5.2 Random utility models - utility function specification

The utility specifications for the RUMs in the study are optimised using a thorough manual sequential search, exploiting established behavioural theory as well as parametric significance tests. As with hyper-parameter optimisation for the ML models, the utility function optimisation is performed using only the training data (first two years of data). However, unlike the ML hyper-parameter search, the optimisation is based on the values of the model parameters, as well as predictive performance. As such, in order to ensure the maximum sample size for each parameter being estimated, the models are fitted using all of the training data simultaneously.

The approach used for the initial models is to hypothesise a full (complex) initial model within a set of behavioural assumptions, and then to simplify the model by applying re-

strictions based on Wald tests of the parameters. These restrictions can be either: (i) fixing related parameters which are not significantly different from each other to be equal, or (ii) fixing individual parameters which are not significantly different from zero to be zero. The simplified model is then assumed to be the optimal model within the set of behavioural assumptions made.

The optimal models from different initial behavioural assumptions (e.g. a different set of input features) may not be strictly nested. As such, in order to compare their performance, each model is assessed on the training data using the Akaike Information Criterion (AIC)

$$G_{\text{AIC}} = 2k - 2\ln G_{\text{log-likelihood}}, \tag{5.13}$$

where k is the number of estimated parameters in the model (Akaike 1974) (the optimised models are still tested on the holdout test data). The AIC adjusts the log-likelihood in order to penalise for the number of parameters in the model. This reflect the higher potential for overfitting in a model with more parameters.

Whilst the AIC is used to compare the optimised models, it is the model parameters which are used to inform the next step in the sequential search. During each model fitting (i.e. each iteration of sequential search), the variance-covariance matrix for the model parameters is estimated. This provides estimates of the values and standard errors for each parameter. These values are used to conduct tests in order to ensure all parameters are (i) significantly different from zero, (ii) significantly different from other linked parameters, and (iii) also have signs and magnitudes which are consistent with accepted behavioural theory.

For example, for (i), in the example model presented in Eq. (5.1) and Table 5.2, $\beta_{0,i}$ (ASC) for each mode, β_1 (duration parameter), and β_2 (cost parameter) are all significantly different from zero. For (ii), the linked parameters ($\beta_{0,i}$) are significantly different from each other, implying that each mode has a different ASC. Finally, for (iii), β_1 and β_2 are negative, therefore resulting in decreased utility for increased trip duration or cost (as is expected), and have relative magnitudes which imply a Value of Time (VoT) (β_1/β_2) which agrees with a-priori expectations.

All initial hypothesised models are subject to the assumption that attributes of a mode affect only the utility of that mode (e.g. cycling duration has no effect on driving utility). Each model will also have a further set of implied assumptions. For example, for the example model presented in Eq. (5.1) and Table 5.2, it is implied that (1) the disutility of increased duration and cost is the same for each transport mode (β for duration or cost does not vary with the transport mode i), (2) the variables from the dataset which have not been included have no impact on the mode utilities, and (3) the error terms ε_i in the utility are independent across all modes (i.e. MNL with no nesting structure).

In order to limit the fit time and number of parameters in the initial hypothesis models, the initial hypothesised models are limited to MNL models with independent error terms, which model only first order interaction of input variables with the utilities. Feature interactions and nested structures are then investigated using optimised models. For example, the direct impact of having a driving licence on the utility of each mode could be included in the initial model, through including it as a dummy variable (as in Eq. (5.14)), but the impact of having a driving licence on the duration parameter (through interacting the licence dummy variable with duration, as in Eq. (5.15)) is not modelled here.

$$V_{in} = \beta_{0,i} + \beta_1 D_{in} + \beta_{2,i} \times (\text{licence}_n = 1), \tag{5.14}$$

$$V_{in} = \beta_{0,i} + \beta_1 \times (\text{licence}_n = 0) \times D_{in} + \beta_2 \times (\text{licence}_n = 1) \times D_{in}. \tag{5.15}$$

This example can also be used to illustrate the two Wald test scenarios. For (i), if β_1 and β_2 in Eq. (5.15) were not significantly different from each other, they could be combined into one parameter. This would suggest there is no interaction between owning a driving licence and the disutility of trip duration. Similarly, for (ii), if any of $\beta_{0,i}$, β_1, or $\beta_{2,i}$ in Eq. (5.14) were not significantly different from zero, they could be fixed to be zero (i.e. removed from the model). Note that for ASCs and dummy variables (e.g. $\beta_{0,i}$ and $\beta_{2,i}$ in Eq. (5.14)), the two tests (i) and (ii) are actually equivalent: The β value for one mode (in this study, always walking), must be fixed to zero, and so fixing these parameters to zero is equivalent to combining it with the equivalent parameter for walking.

The process of model simplification is continued until the optimal model is found under the initial assumptions, with parameters which meet the significance and sign requirements. In the instance where a parameter is found to have an unexpected sign or magnitude, it is investigated further, and not simply removed from the utility specification. Note that adding a feature to the model is treated as a new initial assumption. As such, features are not added to an existing model. Instead a new initial hypothesis model is defined and simplified when adding a new feature. This is because certain parameters will only appear to be significant in the presence of other significant parameters (due to endogeneity), and so it prevents the removal of parameters which may become significant once another feature or is added.

Once the optimal MNL with first order feature interactions has been found (under the corresponding behavioural assumptions), the utility specification for that model may be modified in order to (i) investigate interaction of a covariates in the model with other parameters (as in Eq. (5.15)), or (ii) investigate nested structures (Nested Logit (NL) or Cross-Nested Logit (CNL) models). Any resulting models are then simplified again, by

combining and removing parameters. This process can be repeated iteratively to investigate multiple feature interactions simultaneously with nesting structures.

The reason for only investigating feature interactions and nested structures with optimised MNL models is for efficiency in the search. Modelling feature interactions in utility specifications suffers heavily from the *curse of dimensionality*. For example, fully interacting a categorical variable with just two categories (e.g. the driving licence variable in Eq. (5.15)) with all other input variables doubles the number of parameters in a model. Fully interacting this binary variable with a second covariate with three states with the input variables would result in six times the number of parameters in the model. By simplifying the model before investigating the interactions, this vastly reduces the amount of parameter simplification which is likely to be needed after testing the interactions (which will be slower to run, due to the increased number of parameters). For nested model structures, model fitting times are an order of magnitude higher than for MNL models, due to the increased complexity of calculating the gradients. Again, simplifying the model before investigating nested structures limits the number of computationally expensive NL and CNL model fits needed to optimise the model.

Instead of a limit on the number of iterations of optimisation (as with the ML hyper-parameter search), a hard limit is set on the fitting time for any single model (2 hours on an 8-core machine). This effectively puts a limit on the model complexity and number of model parameters.

TL7: Studies not performing any type of hyper-parameter optimisation

Hyper-parameter optimisation is performed for all classifiers, including ML models and RUMs (utility function specification).

The impacts of performing hyper-parameter optimisation is explored experimentally (see Sections 5.5.1 and 5.5.2).

TL8: Studies not using rigorous hyper-parameter search schemes

All of the ML models are optimised using the same procedure, which involves 100 iterations of SMBO. The search space Λ is defined over all model hyper-parameters.

A thorough manual sequential search is used to optimise the utility functions in the RUM, in order to accurately represent random utility methodology in comparison with ML. The search methodology exploits established behavioural theory as well as parametric significance tests.

The differences in the two methodologies reflect the fundamental differences between ML hyper-parameters and RUM utility specifications.

TL9: Studies optimising hyper-parameters on validation data

The cross-validation for hyper-parameter selection and validation for the utility function specification uses only the training data (first two years of data). As such, the test data (final year) is not seen by the model prior to model testing.

5.2.6 Model selection

Model selection is applied pairwise between a candidate model P^m, and a reference model P^r, as presented in Section 4.2.6. The cost function G used to compare models is CEL. Substituting Eq. (4.36) into Eq. (5.9)

$$\begin{aligned} G(P^m;\mathcal{D}) &= -\frac{1}{N}\sum_{n=1}^{N}\ln P^m(i_n|x_n), \\ &= \frac{1}{N}\sum_{n=1}^{N} d(P^m(x_n), i_n), \end{aligned} \tag{5.16}$$

gives

$$d_n^m = -\ln P^m(i_n|x_n), \tag{5.17}$$

Equation (4.34) therefore becomes

$$\mathbb{E}[d_n^m] = -\sum_{i=1}^{J}\int \ln\left(P^m(i|x)\right)\Pr(i|x)\Pr(x)dx. \tag{5.18}$$

However, as discussed, it is infeasible to calculate this integral, or even to obtain a good approximation of $\Pr(x)$. Additionally, Eq. (5.18) does not include the sampling noise from the finite dataset $\mathcal{T}$ used to train the classifier P^m.

Instead, distributions of the expected value for the CEL for each model are estimated using bootstrapping (Efron 1979). This study uses out-of-sample bootstrapping, where K models are each fitted to one of K bootstrapped samples, with each model used to predict the corresponding out-of-bootstrap sample. This provides K independent estimates of out-of-sample predictive performance.

Note that three alternative bootstrap methods were investigated: the *optimism bootstrap* (Harrell, Lee, and Mark 1996); the *632 bootstrap* (Efron 1983); and the *632+ bootstrap* (Efron and Tibshirani 1997). These methods attempt to compensate for the bootstrapped

classifiers not being fit to all of the available data, and therefore the out-of-sample performance estimate being pessimistically biased. However, all three methods were found to positively bias heavily overfit classifiers, overstating their predictive performance, and so the out-of-sample bootstrap validation method is used.

In order to create K household-wise bootstrap samples $\mathcal{T}_i$, h households are sampled with replacement from the full dataset (where h is the total number of unique households in the dataset $\mathcal{D}$). The sampled households are then used to sample the corresponding trips. The trips for each household are repeated the same number of times that that household appears in the bootstrapped sample. For example, if a household appears twice in the bootstrapped household sample, the trips from that household will each occur twice in the corresponding trip sample $\mathcal{T}_i$. The expected proportion of unique trips in $\mathcal{T}_i$ is 63.2 %, with the remaining 36.8 % of trips consisting of repetitions of the unique trips (Efron and Tibshirani 1997). The 36.8 % of trips in $\mathcal{D}$ which do not occur in the bootstrap sample $\mathcal{T}_i$ form the corresponding out-of-bootstrap validation sample $\mathcal{V}_i$.

These samples are used to provide K independent estimates of out-of-sample predictive performance $G_{\text{OOB},i,m}$

$$G_{\text{OOB},i,m} = G(\mathcal{A}^m(\mathcal{T}_i;\boldsymbol{\lambda}^m);\mathcal{V}_i). \tag{5.19}$$

For the investigations in this thesis, K is set to be 100. The same 100 samples are used for all classifiers.

The bootstrap results can be used to approximate the distribution of the performance estimates for the candidate model P^m and reference model P^r. These distributions can then be used to investigate the inequality given in Eq. (4.31) from Section 4.2.6

$$G\left(\mathcal{A}^m(\mathcal{D};\boldsymbol{\lambda}^m);\mathcal{U}\right) \leq G\left(\mathcal{A}^r(\mathcal{D};\boldsymbol{\lambda}^r);\mathcal{U}\right).$$

The 100 bootstrap samples provide 100 paired estimates of the out-of-sample performance of the two models. As such, we can conduct a paired t-test to investigate the null hypothesis that the true mean difference between the paired samples is zero, versus the alternative hypothesis that the true mean difference is not equal to zero

$$H_0 : \mu_d = 0, \tag{5.20}$$

$$H_1 : \mu_d \neq 0. \tag{5.21}$$

The test statistic for the t-distribution Probability Density Function (PDF) $f(t)$ is given by

$$t = \frac{\bar{\delta}_i}{s/\sqrt{K}} \tag{5.22}$$

where $\hat{\delta}_i$ and s are the mean and standard deviation of the differences δ_i across the K bootstrap iterations.

The above test can be used to determine if the distribution of of the bootstrapping results for one model is significantly different to another. However, it does not indicate if the differences between the expected performances are themselves significant for applying the models for prediction. Whilst such tests exist in the literature for parametric models, there are a lack of tests which can compare non-parametric classifiers for probabilistic classification.

To address this need, we investigate the impact of the test-sample size on the *power* of the two-sample t-test, by considering the total joint likelihood of the data given the model. This approach can be applied to any probabilistic classifier (whether parametric or non-parametric).

As discussed in Section 5.2.4, absolute differences in *log-likelihood* indicate relative differences in the joint likelihood of the data. By multiplying $E[\delta]$ (the expected difference in CEL between two classifiers) by the test sample size, the expected log-likelihood, and therefore relative difference in joint likelihood, can be estimated. For example, for a test sample of size $n = 500$ with $E[\delta]$ of 7.328×10^3, the difference in expected log-likelihood between the classifiers over the test sample is 3.664. This signifies that the test data are expected to be $e^{3.664} = 39$ times more likely under the candidate model P^m than the reference model P^r. This corresponds to a 5 % significance test of rejecting the two-tail hypothesis H_0' that the unknown test data are equally likely under the reference model than the candidate model ($0.975/0.025 = 39$). More generally, the conditional probability of accepting H_0' given a fixed difference in performance δ is

$$P(H_0'|\delta) = \frac{1}{(1+e^{n\delta})} \tag{5.23}$$

for a test sample of size n.

The distribution of the of the differences δ between a pair of models is defined by the t-statistic given in Eq. (5.22). As such, the marginal (unconditional) probability of accepting H_0 can be calculated by integrating Eq. (5.23) over the distribution of δ defined by the

t-statistic. Combining Eqs. (5.22) and (5.23) and integrating gives

$$P(H_0') = \int_{-\infty}^{\infty} \frac{f\left(\frac{\bar{\delta}_i - \delta'}{s/\sqrt{K}}\right)}{(1+e^{n\delta'})} d\delta'. \tag{5.24}$$

where $f(t)$ is the PDF of the t-distribution with $K-1$ degrees of freedom (d.o.f.).

Whilst no analytical solution exists for Eq. (5.24), it can be estimated numerically, e.g. by using the Simpson's rule. This test can then be used either to check whether one model will significantly outperform another for a given test sample size, and to provide the test sample size at which one model will significantly outperform the other.

TL10: Studies not analysing uncertainty in performance estimates

The uncertainty in performance estimates of the classifiers is estimated using 100 folds of the out-of-sample bootstrap. This accounts for the sampling noise from the finite dataset $\mathcal{D}$, and as such evaluates the external predictive performance of the algorithm $\mathcal{A}^m$ when using hyper-parameters/utility specifications λ^m, when trained on *any dataset* of size N sampled from the population $\mathcal{C}$.

A t-test which investigates the mean of the differences between the paired bootstrap performance distributions is specified. Finally, an approach which investigates the power of this test to determine whether an unknown test sample is less than or equally likely under a reference model P^r than a candidate model P^m, for any given test sample size n is also proposed.

5.3 Modelling framework

This section provides the details of the modelling framework implemented for this study. Whilst Section 5.2 presents the general approach and theory, this section focuses on the specific details of the implementation. Figure 5.2 shows a flowchart of the framework, highlighting the four stages: (1) data pre-processing, (2) model optimisation, (3) holdout validation, and (4) bootstrapping.

The process is fully automated for the ML classifiers, using the *scikit-learn* and *Hyperopt* python packages (Pedregosa et al. 2011; Bergstra, Yamins, and Cox 2013). The automated process requires only the hyper-parameter search space to be defined. As the process is automated, it is straightforward to generate results for any classification algorithm implemented in scikit-learn, as well as algorithms or packages with a scikit-learn wrapper e.g.

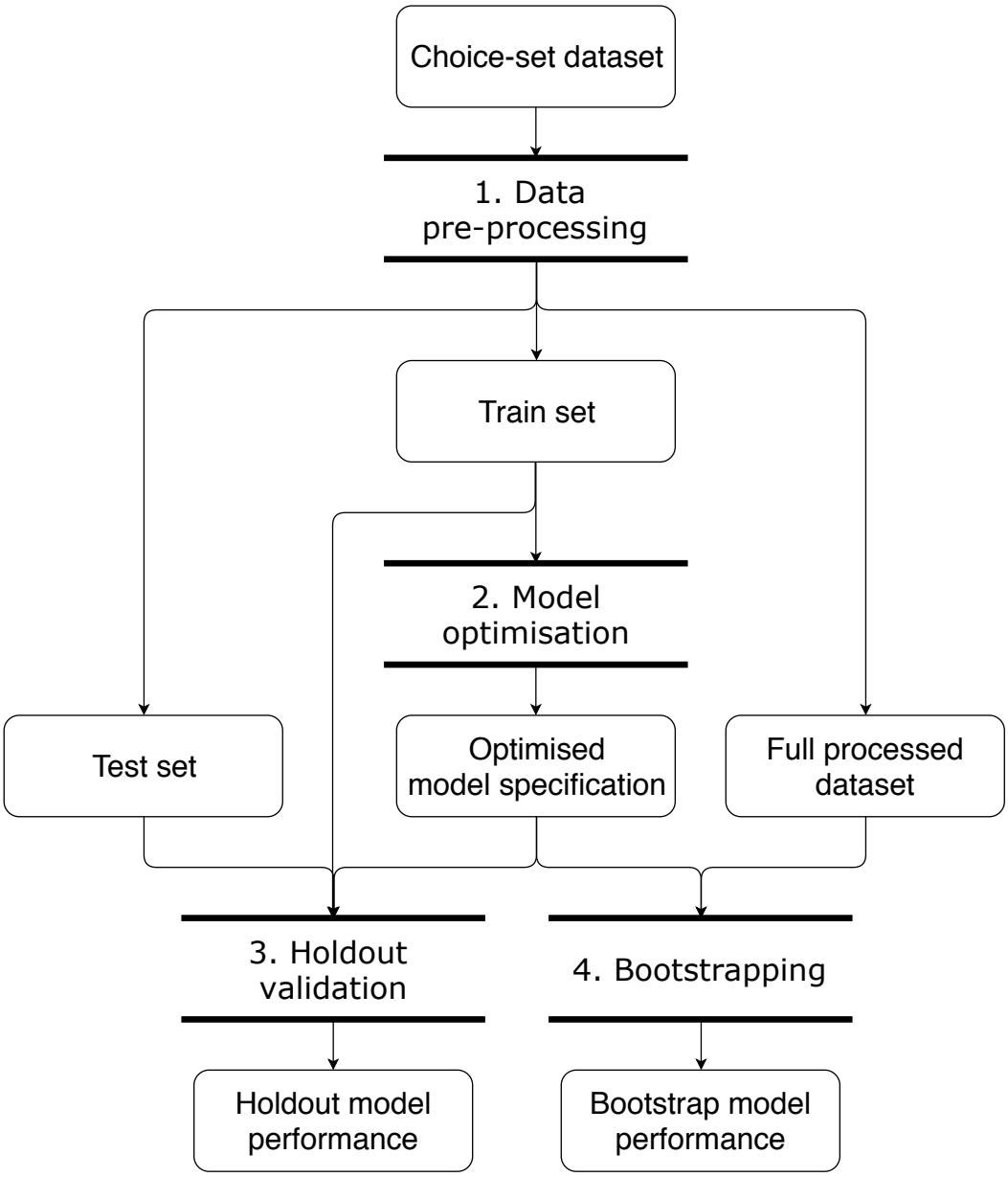


Fig. 5.2 Flowchart of modelling framework.

LIBSVM (Chang and Lin 2011); *XGBoost* (Chen and Guestrin 2016); and *Keras* (Chollet et al. 2015).

The RUM process is partially automated, with stage 2 (model optimisation) carried out manually. The implementation makes use of the *PythonBiogeme* software (Bierlaire 2016), with wrappers to automate stages 3 and 4 once optimised utility specifications are found.

5.3.1 Data pre-processing

The dataset is pre-processed prior to training the ML classifiers. The categorical data are one-hot encoded, so that an n-class categorical variable is replaced with n binary variables. An additional sine and cosine are calculated for cyclical data (`start_time`, `day_of_week`, and `travel_month`) to preserve the cyclical ordering. These are included as features alongside their linear representation. This results in a total of 44 input features in the full dataset. The additional features are not used by the RUM models.

The data are split into train and test data by survey year, with the first two years (2012/13 and 2013/14) forming the training set, and the final year (2014/15) forming the test set.

For models where scaling is important the data are scaled to zero-mean unit-variance (see Section 5.5.1). This scaling is *fitted* on the training data only, and the scaling from the training data is applied to the test data (this applies to all validation schemes - holdout, k-fold, and bootstrapping).

5.3.2 Model optimisation

As discussed in Section 5.2.5, model optimisation is carried out differently between the ML and RUM classifiers. The following sections discuss each in turn.

5.3.2.1 Hyper-parameter selection

Hyper-parameter selection for the ML classifiers is performed using the Hyperopt python library. Appropriate prior distributions are defined for each parameter (see Section 5.5.1). The search is carried out over 100 iterations using only the train data only. The first 20 iterations of the search draw suggested values randomly from the prior distributions. The subsequent 80 iterations use the TPE search algorithm, with the surrogate function updated from the previous trials. Stratified 10-fold cross-validation is used to estimate the CEL in each iteration of hyper-parameter selection. The same fold splits are used across all 100 iterations for all models. The folds are grouped by household, so that all trips made by all members from the same household appear in only one fold. The optimal hyper-parameters are those that result in the lowest mean cross-validation CEL across the 100 trials.

5.3.2.2 Utility function specification

Utility functions in the RUMs are optimised using sequential manual search. Separate utility functions are specified and optimised for each mode. Each model is fitted to all of the test data, during which the log-likelihood fit, AIC, and parameter distributions are also calculated. Restrictions are sequentially placed on parameters where required at a 0.05 significance level. The optimal utility specifications are those which result in the lowest AIC.

5.3.3 Bootstrapping

The distribution of the CEL of the final model is estimated using out-of-sample bootstrapping. One-hundred iterations of bootstrap sampling are used to estimate the distribution of the out-of-sample performance estimate. Bootstrap samples are generated grouped by household so that, in a given sample, all trips made by all members from the same household appear an equal number of times (possibly zero). The same 100 bootstrap samples are used for all models.

5.4 Random utility approach

This section presents the investigations into the RUM approach. These investigations represent the application of state-of-practice techniques to the new data developed for this thesis and serve as a performance benchmark for the remaining classification techniques. Three MNLs (non-nested, with independent error terms) are optimised using the modelling framework presented in Section 5.3.

Model 1 uses only the mode-alternative variables from the choice-set recreation (durations for walking and cycling; duration breakdown and fare for public transport; duration, fuel-cost, congestion-charge, and traffic for driving), with none of the profile data from the LTDS.

Models 2 and 3 make use of the socio-economic and trip profile (covariates) from the LTDS. It is trivial to include the categorical covariates with few categories (vehicle ownership, journey purpose, driving licence, sex), but more complex for continuous variables and categorical variables with several (> 5) categories (trip distance, age, start time, day of week, travel month). Note that, as explained in Section 6.3.1 vehicle ownership is grouped into three categories: (1) no vehicles in household, (2) less than one vehicle per adult (VEHICLE_OWNERSHIP_1), and (3) one or more vehicles per adult (VEHICLE_OWNERSHIP_2); and journey purpose is grouped into five categories: (1) Home-based work (HBW), (2) Employers business (B), (3) Home-based education (HBE), (4) Home-based other (HBO), and (5) Non-home based other (NHBO).

Distance is included in all models as a continuous feature, with a separate parameter (β_i) for each mode (fixed to zero for walking). All other parameters are binned (grouped) and included as dummy variables. Three bins are used for age, based on the commonly used behavioural groupings of child (<18), adult (18-64), and pensioner (65+). The Transport for London (TfL) peak/off-peak times are used define four departure time bins: AM peak (06:30-09:29), inter-peak (09:30-16:29), peak (16:30-19:29), and night (19:30-06:29). The day of the week is grouped into three bins: work-days (Monday-Friday), Saturday, and Sunday. Finally, the travel month is grouped into two bins: winter (December-February), and all other months.

Model 2 uses all of the socio-economic data described above, with none of the additional features from recreating passenger mode choice-sets. Model 3 uses all features (the mode-alternative attributes and the LTDS profile).

Both models 2 and 3 only investigate first order interaction of covariates with the utilities, due to the considerable complexity of testing for higher order interactions impartially. Instead, higher order interactions of covariates are investigated using the assisted specification approach (see Section 5.6).

5.4.1 Nested logit and cross-nested logit

The optimised utility specifications from model 3 (see Section 5.4) are used to investigate nested structures. All 13 possible NL structures with four classes are tested (six combinations of a single nest of two modes, four combinations of a single nest of three modes, and three combinations of two nests of two modes). A Wald test is applied to the nesting parameters in the utility specification to determine whether a nest is valid.

The valid nests identified from the NL tests are combined to form CNL models. The highest performing nested model structures are compared to the ML classifiers (see Section 5.5.1).

5.5 Machine learning investigations

This section introduces the ML-based investigations carried out within the thesis. This includes a comparative study of machine learning classifiers; as well as investigations into sampling methods for hierarchical data, and the impacts of using mode-alternative attributes in the feature vector. These investigations address the technical limitations of the existing ML research, and as such represent a rigorous comparison of ML classification techniques for mode-choice prediction when applied to the new generation of data.

5.5.1 Comparative study of machine learning classifiers

This investigation compares the performance of six ML classifiers alongside traditional RUM models for predicting mode choice. This includes linear models, Artificial Neural Networks (ANNs), Ensemble Learning (EL) methods, and Support Vector Machines (SVMs). Specifically, the following algorithms are investigated:

- RUMs
- Logistic Regression (LR)
- Feed-Forward Neural Networks (FFNNs)
- Random Forests (RFs)
- Extremely randomised Trees (ET)
- Gradient Boosting Decision Trees (GBDT)
- SVMs

An overview of each of these classification techniques is given in Sections 2.2.1 and 2.3. Note that the LR model in this comparison is treated as a ML algorithm (i.e. does not employ a utility-based optimisation method).

The modelling framework presented in Section 5.3 is applied to each ML algorithm. The cross-validation, holdout-validation, and bootstrapping results are used to compare the relative performance of each model. By comparing the out-of-sample validation results with the train-error, the impact of TL3 (studies using inappropriate validation schemes) is assessed quantitatively. The predicted mode shares using probabilistic classification are compared to those obtained using discrete classification in order to investigate TL6 (studies using only discrete metrics). The improvement in model performance during the hyper-parameter optimisation is investigated in order to assess TL7 (studies not performing any type of hyper-parameter optimisation) experimentally.

Hyper-parameter search spaces for each algorithm are given in Appendix A.1. A brief explanation for each classifier is given below.

Note that RFs, ET, and SVMs do not inherently output choice probabilities found using MLE. For RFs and ET, the voting ratios of each node in the tree are a probability-like distribution bound to [0,1]. In this study these values are treated as probabilities. For SVMs the decision function is not probability-like. Instead a heuristic method is applied to estimate the probabilities from the decision function (see below). These assumptions are investigated by plotting *reliability curves* of predicted probabilities vs observed mode ratios for each classifier for each mode.

Logistic regression: The *scikit-learn* implementation is used for the LR classifier. The *saga* solver is used as it handles both *L1* and *L2* regularisation, as well as inherently dealing with multiclass problems (Defazio, Bach, and Lacoste-Julien 2014). For the search space, a suitable distribution is used for the regularisation parameter C. The remaining hyper-parameters are categorical, and all values are included in the search space for each. Both *L1* and *L2* regularisation and the convergence of the *saga* algorithm are sensitive to the relative scaling of the features, and so data are scaled to zero-mean unit-variance.

Feed-forward neural networks:

The *Keras* package is used for the high-level interface to develop the FFNN models. *TensorFlow* is used as the *backend* to implement the models.

A generator script is written which uses the hyper-parameter search functionality to create FFNNs with adaptive network architecture. The input and output layers are fixed according to the dataset, with one node in the input later for each feature (43 nodes), and one node in the output layer for each mode (4 nodes). The output layer is passed through a softmax function in order to generate choice probabilities. CEL is used as the cost function during model training, resulting in MLE.

The rest of the network structure is determined by the selected hyper-parameters. The *number of hidden layers* (between one and three), *optimiser*, *batch size*, and number of *epochs* are all set by global hyper-parameters. The default parameters are used for each optimiser, as suggested by the *Keras* documentation. Each hidden layer specified has a different number of nodes, activation function, and dropout rate, which are all specified within the model hyper-parameters. ANNs are more data-efficient with scaled data (so they do not need to use data to learn extra scaling weights), and so data are scaled to zero-mean unit-variance.

Gradient boosting decision trees:

The *XGBoost* library is used for the GBDT models. The number of boosting rounds is set dynamically using the *early stopping rounds* variable, i.e. additional rounds of boosting are performed until the cross-validation error score does not improve for 50 consecutive boosting rounds (capped at 6000 rounds). As the total number of boosting rounds is dependent on the *learning rate*, a fixed *learning rate* is determined which allows the models to fit in reasonable time. The remaining values are taken from the *Hyperopt-sklearn* package. *XGBoost* uses a *one-vs-rest* approach for multiclass problems, so that each boosting round contributes one decision tree for each class in the dataset. Each leaf node in a given tree contributes a

continuous value to a raw regression score for the corresponding mode. These raw regression values are summed over all trees and passed through a softmax function to generate choice probabilities. By training to minimise CEL, this results in well calibrated choice probabilities. Decision trees are robust to scaling of the data, and so no scaling is applied.

Random forests and extremely randomised trees: The *scikit-learn* implementations are used for the RF and ET models. The default search spaces from the *Hyperopt-sklearn* package (Komer, Bergstra, and Eliasmith 2014) are used for each algorithm. The decision trees in the ensemble are inherently multi-class. Probability estimates are obtained by taking the mean of the ratios of each class in the corresponding leaf nodes across all trees in the ensemble. Decision trees are robust to scaling of the data, and so no scaling is applied.

Support vector machines: There are two inherent limitations with SVMs which need to be addressed for this study: (i) SVMs are binary classifiers and so do not inherently support multiclass problems, and (ii) SVMs cannot inherently output choice probabilities. The *LIBSVM* library is used within *scikit-learn* for the SVM models. *LIBSVM* uses a *one-vs-one* decision strategy for multiclass problems, meaning six binary classifiers are used to implement the four-class classification problem. The method proposed by Wu, Lin, and Weng (2004) is used to provide calibrated probability predictions from the SVM decision function. Five-fold cross-validation is used to estimate the parameters for the probability calibration.

When combined, the 10-fold cross validation for performance estimation, five-fold cross-validation for probability calibration, and six binary classifiers for the four-class binary problem, result in 300 train-test cycles for each of the 100 iterations of hyper-parameter selection. As discussed in Section 2.3.5, SVM efficiency scales at a minimum with $O(n^2)$ (where n is the number of rows in the data), rising to $O(n^3)$ for high regularisation values. As such, in order to ensure completion in reasonable time, hyper-parameter optimisation for the SVM-based models is performed on a 10 % sample of the training data (sampled grouped by household).

Initial experiments showed C values larger than ~ 50 result in very large fit times (over 24 hours for a single iteration of hyper-parameter selection). As such, the upper limit of the prior distribution for C is set at 500. The remaining values are taken from the *Hyperopt-sklearn* package. SVMs are sensitive to data scaling, and so the data are scaled to zero-mean, unit-variance.

5.5.2 Sampling methods for hierarchical data

In order to investigate the potential issues with trip-wise sampling of numerical data, the modelling framework presented in Section 5.3 is applied again to each algorithm introduced in Section 5.5.1 using *random trip-wise sampling* (instead of sampling grouped by household) for the cross-validation folds for model optimisation.

The apparent (using random-sampling) and actual (using external validation/grouped bootstrapping) performance of the trip-wise models are compared to investigate the overestimation of model performance using incorrect sampling methods for hierarchical data (TL4). By applying household-wise sampled bootstrapping to the trip-wise optimised models and comparing it to the results of the household-wise optimised models, significance tests can be executed to investigate whether models optimised using incorrect sampling methods perform significantly worse than those optimised using correct sampling.

5.5.3 Mode-alternative attributes

In order to show the contribution of the framework for recreating passenger mode choice-sets introduced in Chapter 6 to the model predictions, an additional GBDT model is optimised and trained on the dataset, using only the socio-economic and demographic profile data from the LTDS. This is referred to as the *raw-data model*. The raw-data model has 30 input features, compared to 44 in the unrestricted dataset (see Table 6.4 in Chapter 6 for a list of corresponding attributes). Aside from the input features, the raw-data model is optimised using the same modelling framework as the original GBDT model from Section 5.5.1 (referred to as the *choice-set model*).

The bootstrapping results are used to conduct a significance test to investigate TL1 (studies not including any attributes of the mode-alternatives) from a predictive performance basis. By comparing the feature importances for each model, the importance of the additional choice-set features is investigated.

5.6 Assisted specification of RUMs

This section presents a new assisted specification approach, where a fitted ML classifier is used to inform the utility function structure in a RUM. This approach attempts to combine the predictive power and flexibility of ML classifiers, with the interpretability and strong behavioural foundation of the random utility approach.

The RUMs discussed in Sections 5.4 and 5.4.1 only model first order interactions of the covariates from the LTDS profile with the choice utilities. This is due to the complexity

of exploring the possible utility specifications in high dimensionality using an open-ended manual search, as explained in detail in Section 2.2.2. In practice, this has restricted modellers to using coarse categorical covariates and only investigating first order interactions of the covariates with model inputs.

To address this issue, this investigation explores the possibility of using the structure of a fitted EL model to inform the specification of the utility functions in a RUM. This enables a directed manual search into higher order and non-linear feature interactions in the RUM utility specification.

EL algorithms have a number of features which make them well suited to this task. Firstly, by analysing the individual Decision Trees (DTs) in an ensemble, the relative importances of each feature can be extracted. These feature importances can inform the modeller which features to focus on or pre-process further when developing the RUM. Secondly, the values at which each feature is split in the DTs in the ensemble provide information on the relationship between a feature and output class predictions. As discussed in Section 2.3.3, the DTs in an EL algorithm create binary splits using only the rankings of feature values. As such, DTs are independent of feature scaling, or any monotonic transformation of the features. As well as making the models more robust to varying input data, this provides the EL algorithms with the flexibility to approximate any monotonic non-linear function of the features. By analysing the distribution of the split points for each feature, it is possible to identify these non-linear relationships in the model. These can then be added to the RUM utility specification. Finally, the process used to identify feature importances can also be applied to feature interactions, by analysing the information contribution from each combination of sequential split. This can inform the modeller which feature interactions to test in a complex utility specification.

In order to extract the necessary information, the full EL model is processed using the *Xgbfir* python library (Kostenko 2018). The library extracts and analyses each decision tree in the fitted ensemble, in order to identify the split points and total gain for each feature. The hierarchical structure of the splits in the tree is also analysed to identify second, third, and higher order feature interactions, and rank them according to their importance (total gain).

The histograms of the split points for the continuous covariate features (e.g. distance) are analysed to spot underlying non-linear interactions of input features with mode choice. These non-linear interactions are then tested in a RUM utility specification. Next, the histograms for binned continuous variables (e.g. age) are used to identify they key split points for groupings. These split points define heuristic bins in the features, which are tested against the a-priori bins specified in Section 5.4.

Finally, the most important second and third order interactions of input features in the EL model are used to identify the interactions to add to the RUM model. Again, these are tested in the RUM utility specification.

The final resulting RUM, which combines the changes inferred from the EL model, is then compared with the ML classifiers from Section 5.5.1.

As a first look into an assisted specification approach, this investigation is intended to illustrate some ways in which ML classifiers can be used to inform the utility specifications of a RUM, and to determine the possible benefits from doing so. A formal framework for automating the specification is left to further work.

5.7 Summary

This chapter presents the methodological framework used within this thesis to investigate travel mode choice. A theoretical approach is established which specifically addresses all 10 technical limitations of previous ML-based studies identified in Chapter 3.

The approach is implemented in a modelling framework, consisting of data pre-processing, model optimisation, holdout-validation, and bootstrapping. Separate model optimisation schemes are executed for ML and RUM classifiers. The different optimisation schemes represent the fundamental differences between ML hyper-parameters, which are purely mechanistic and not intended to reflect behavioural assumptions, and utility specifications, which are hypothesised from a behavioural foundation.

The RUM approach is established for both MNL and nested structures for comparison with the ML classifiers. The ML investigations into mode-choice prediction are described, including a comparative study of machine learning classifiers, an investigation into sampling methods for hierarchical data, and an investigation into mode-alternative attributes.

Finally, an assisted specification approach is described, which uses the structure of a fitted EL model to inform the specification of the utility functions in a RUM.

Chapter 6

Recreating passenger mode choice-sets

6.1 Overview

A new framework for recreating passenger mode choice-sets has been developed for this research. The framework is used to build a dataset combining individual records from the London Travel Demand Survey (LTDS) with closely matched trip trajectories alongside their corresponding mode-alternatives (i.e. the choice-set faced by the passenger at the time of travel), and precise estimates of public transport fares and car operating costs. This represents the most comprehensive and closely tailored travel dataset for estimating travel choices in a major metropolitan area. The dataset is openly available online from the website of the journal article detailing this work (Hillel, Elshafie, and Jin 2018).

The framework has been implemented within an automated process, which can be adapted for the fast assembly of similar datasets from historic trip data for any geographical region.

This chapter presents an overview of the data generation framework as well as the dataset itself. Firstly, Section 6.2 presents the input data sources used to create the dataset. Section 6.3 then outlines the methodology used to construct the dataset. Section 6.4 provides an overview of the finished dataset, and presents two preliminary investigations using the data, analysing the correlation between trip length and mode choice and the impacts of using trip-wise sampling with hierarchical data. Finally, Section 6.5 summarises the chapter and presents conclusions of the preliminary investigations.

6.2 Input data sources

The data generation process makes use of several input data sources. These are presented in Table 6.1. A detailed overview of the LTDS and Google Directions Application Program-

ming Interface (API) is given in the following sections. The cost model and associated data sources are described in the relevant section of the methodology (Section 6.3.4).

Table 6.1 Input data sources for generation of mode-alternative attributes. Sources marked with * are assembled manually.

	Data source	Information
Historic trip data	LTDS	Trip O-D pairs, trip related information, individual socio-economic profiles, vehicle fuel types
Directions service	Google Directions API	Mode-alternative routes and durations, traffic information
Cost model	TfL Unified API	Public transport rail fares
	WebTAG	Driving Vehicle Operating Costs (VOCs) formulae and coefficients
	London congestion charge/ resident discount zones	Driving congestion charge and discounts
	Bank holiday dates*	Public transport and congestion charge discounts
	NaPTAN	Station locations and codes
	Oyster free-interchange stations*	Free station interchanges

6.2.1 London Travel Demand Survey

The LTDS is a continuous survey carried out by Transport for London (TfL) of a sample of households within M25 orbital motorway. The survey consists of three parts: a household questionnaire, an individual questionnaire, and a trip diary (Transport for London 2015). Each household is surveyed on one day of the year.

The household questionnaire contains details of the household structure and characteristics.

The household questionnaire also records details, including fuel type, of all vehicles available to members of the household on the survey date.

Each household member over five years of age completes the individual questionnaire and trip diary. The individual questionnaire details socio-economic and travel-related information, including working status, driving licences, public transport season tickets/discounts held, and disability information.

In the trip diary, the individual gives details of all trips made on the survey date, including the origin and destination purposes, origin and destination locations, and trip start time and duration. The origin and destination locations are used to calculate the straight-line distance.

Trips using multiple transport modes or with a transfer on the same mode are given in multiple stages. For example, a public transport trip may compose of

- A walking *access* stage from the origin to the first public transport stop
- One or more stages for each separate public transport service (bus/London Underground Line/National Rail train etc)
- One or more walking *transfer* stages for each transfer between services, where a change in platform/stop is required (train interchanges at the same platform and bus interchanges at the same stop have no transfer stage)
- A walking *egress* stage from the final public transport stop to the destination

The origin, destination, start time, duration, and transport mode are recorded for each stage. As with the trip distance, the stage origins and destinations are used to calculate the straight-line distance. The cumulative straight-line distances of the stages for each mode within a trip are used to determine the distance-based main mode for that trip.

The data are organised grouped by *household, vehicle, person, trip,* and *stage*. Fig. 6.1 shows the grouping and inheritance structure of the data, alongside the selected attributes. The attributes `fuel_type` and `travel_mode` are shown with the classes they are determined by (*vehicle* and *household*), rather than the *trip* they are associated with.

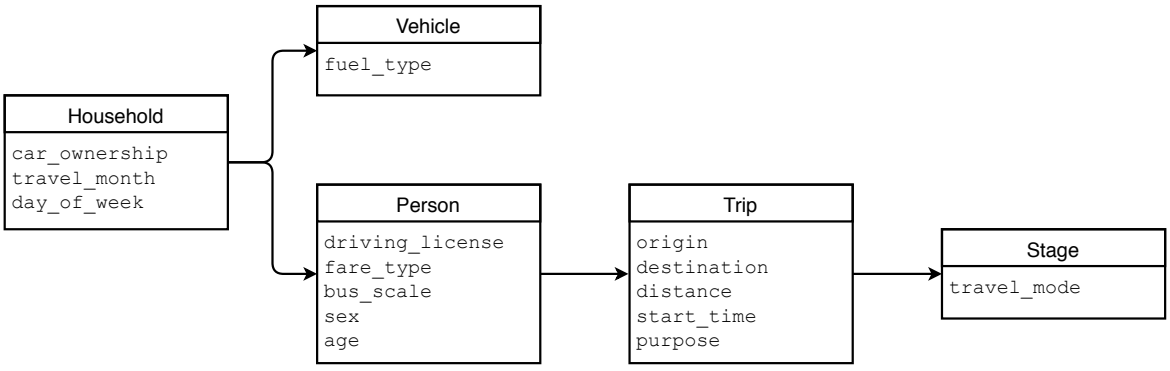


Fig. 6.1 Class structure of LTDS.

The survey is grouped by financial year (April-March), starting in 2005/06. This thesis uses the three most recent available years of data, from 2012/13-2014/15, containing 134 486 trips made by 57 640 individuals within 24 248 households.

The data from 2012 onwards is used in order to ensure good consistency between the transport conditions at the time of travel with those used to model the mode-alternative route options. This is particularly important for the public transport network, which underwent major investment in preparation for the 2012 London Olympics.

6.2.2 Google Directions Application Programming Interface

The Google Directions API, hereafter referred to as the *directions service*, is an online travel planning service. It allows for the retrieval of predicted journey information, including optimal route and duration, using the Google Maps service. It has almost full worldwide coverage for driving and walking directions (Google Developers 2018), and also provides full public transport and bicycling directions for cities across the UK.

Google maintains an accurate graph of London's public transport and road infrastructure and services using data from a range of sources. This includes: satellite, aerial, and street view imagery; publicly available data from TfL, Network Rail, and other public transport providers; user submitted map corrections from the Waze and Google maps applications; and mobile phone Global Positioning System (GPS) location information (Madrigal 2012; Miller 2014; Google 2018).

In order to determine link travel speeds on the graph of the road network, Google generates real-time traffic flows using crowd-sourced GPS data. For the public transport network, Google also receives up-to-date travel information from network operators, including TfL and Network Rail.

The directions service requires, as a minimum, an origin and destination to be specified in the Hypertext Transfer Protocol (HTTP) request. There are several optional values which can also be specified, those used in this study are presented in Table 6.2.

Table 6.2 Directions service optional request parameters.

Parameter	**Values**
`mode`	`driving`, `walking`, `bicycling`, `transit`
`departure_time`	(Driving and PT only)
`traffic_model`	`best_guess`, `pessimistic`, `optimistic` (Driving only)

Commercial access to the Google Directions API has been obtained for this research, allowing for 100 000 separate directions service requests per day.

The calculation of the route and duration returned depend on the mode specified:

Walking The route and duration returned are independent of departure time, with routes prioritising footpaths and pavements.

Cycling The route and duration are independent of departure time, with routes prioritising cycle paths and quieter roads where available.

Public transport (Transit) The routes and durations are extracted for specific times of the day and days of the week. The route and duration are calculated using timetable information. The returned route is broken into separate stages for each walking/bus/rail leg of the journey (as with the trip data in the LTDS). Transfers between services represent separate stages. For routes where there is not a convenient public transport option (e.g. very short trips or trips in areas with low public transport coverage) the suggested public transport route may use no public transport services, instead using walking only (matching the walking route).

Driving The routes and durations are extracted for specific times of the day and days of the week. The route and duration are calculated using a traffic model which represents the typical traffic conditions on that day and time. Three traffic levels can be specified which impact both route and duration: best guess (typical traffic for that time of day and day of week); optimistic (lighter than typical traffic); and pessimistic (heavier than typical traffic).

6.3 Methodology

In order to create a dataset of trip trajectories alongside their corresponding route-alternatives, an input dataset of recorded trips is combined with an online directions service and a closely tailored cost model to generate predicted routes, durations, and costs for each mode. The LTDS (see Section 6.2.1) is used as the input dataset of historic trip trajectories. The Google Directions API (see Section 6.2.2) is the online directions service used to add the corresponding routes and durations for the mode-alternatives that the passenger could have taken. The cost model is presented in Section 6.3.4.

Each trip in the LTDS has a recorded origin, destination, and departure date and time. The available trip-alternative routes and durations are obtained by requesting directions for each mode from the directions service for each trip in the LTDS. The directions are requested with matching origin and destination as the LTDS trip for all modes, as well as matching departure time and day-of-the-week for public transport and driving directions. This process is illustrated with an example of the routes generated by the directions service for an example trip in the LTDS in Fig. 6.2.

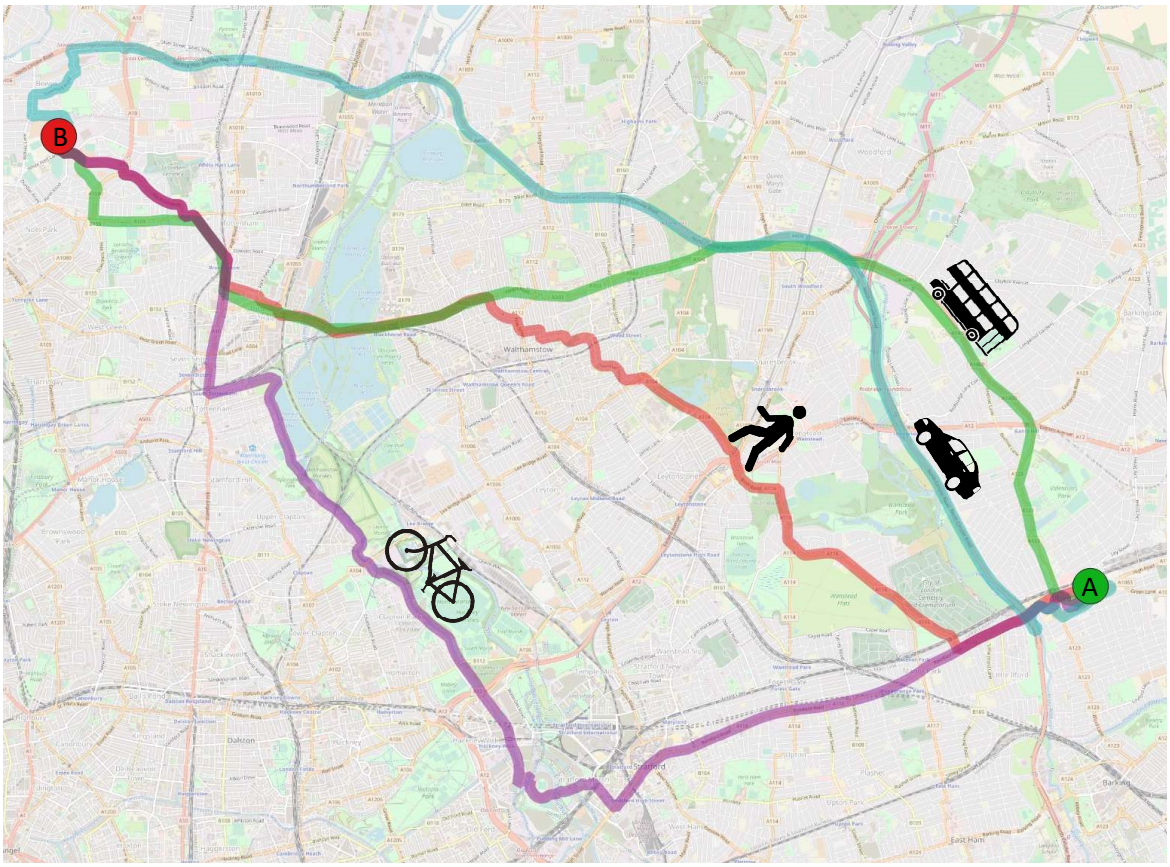


Fig. 6.2 Mode-alternative travel routes generated by directions service for a single Origin-Destination (O-D) pair (A to B).

The routes from the directions service are then screened for relevance to the study.

Finally, estimates of public transport and driving costs are generated for each trip using a closely tailored cost model. The costs, durations, and details of the suggested routes are stored alongside relevant information from the LTDS to form the new dataset.

The overall dataset building process is presented in Fig. 6.3. The following sections focus in turn on the individual steps in the process: (1) pre-processing trips from the LTDS, (2) generating mode-alternative routes and durations using the directions service, (3) screening trips, (4) adding mode-alternative cost estimates. This develops the initial work from Hillel, Guthrie, et al. (2016).

The dataset building framework has been fully automated and can be easily adapted for the fast assembly of similar datasets from historic trip data from any geographical region. Both the raw input data from the LTDS and the finished dataset are stored in a Structured Query Language (SQL) database. The Python Programming Language is used to automate each step in the process, and forms the interface between the SQL database, directions service, and TfL Unified API.

6.3.1 Pre-processing trips

A consistent set of individual records from the LTDS from April 2012 to March 2015 is used to build the dataset. The 2012/13 year is selected as the start date to minimise discrepancies between the transport network at the point of travel and that used to generate routes using the directions service. Several infrastructure projects were completed in advance of the 2012 London Olympics, and so trips prior to this year used a substantially different public transport network. These three years of LTDS data comprise of 134 486 trips made by 57 640 Greater London residents.

The trips in the LTDS are pre-processed before building the new dataset. This section describes the process, including: (i) removing trips missing either the origin or destination location, (ii) removing trips with zero reported length, (iii) assigning each trip to a primary transport mode, (iv) assigning each trip to a journey purpose, and (v) determining the vehicle ownership category for each trip.

Firstly, trips missing either the origin or destination locations are removed (33 trips). Trips with zero reported length (circular trips with the same origin and destination) are also removed (291 trips). This leaves 134 162 trips.

Next, each trip is assigned to one of the four main modes (walking, cycling, public transport, and driving). The mapping, along with the associated frequency of each mode in the input dataset is shown in Table 6.3. For mixed mode journeys, the assigned mode is the

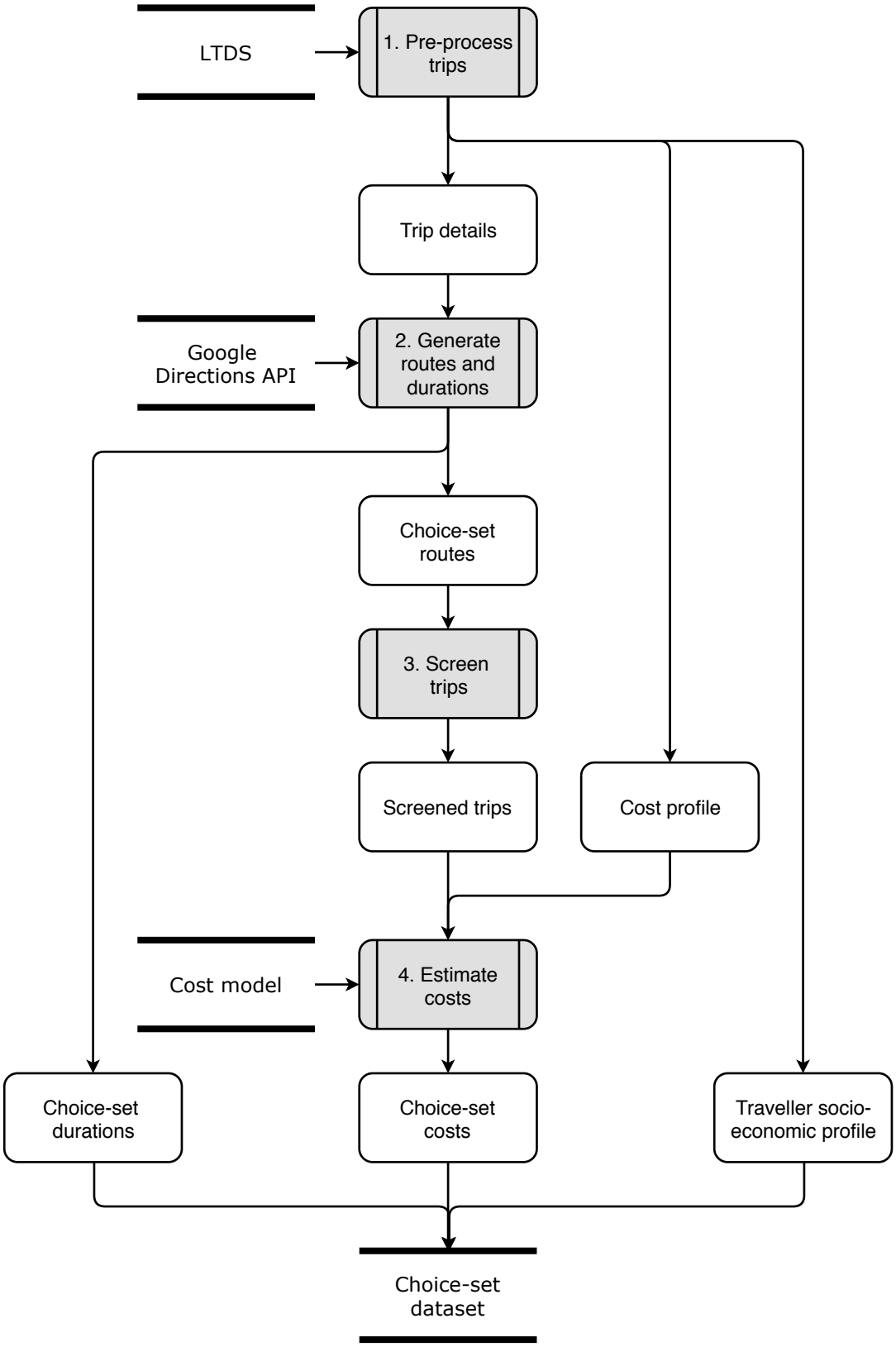


Fig. 6.3 Flowchart of dataset building process.

mode used to travel the most distance, based on the individual stages which make up that journey, as per the norm of transport modelling practice.

Trips made by other modes (school/work bus, dial-a-ride, coach, plane, boat, other) account for less than 0.5 % of trips, and so are omitted for the purpose of this thesis. The same approach can be applied to extend coverage to these modes.

Next, each trip is assigned to one of five journey purposes, derived from the origin and destination purposes in the LTDS: (1) Home-based work (HBW) - between home and regular workplace or home and pick-up/drop-off at workplace; (2) Employers business (B) - between home/usual workplace and other work purpose or to/from work delivery/loading; (3) Home-based education (HBE) - between home and education or home and pick-up/drop-off at education; (4) Home-based other (HBO) - between home and non-work/education; and (5) Non-home based other (NHBO) - all other trips.

Finally, each trip is assigned to a vehicle ownership category according to corresponding household vehicle ownership, out of (1) no vehicles in household, (2) less than one vehicle per adult, and (3) one or more vehicles per adult.

6.3.2 Generating mode-alternative routes and durations

Six directions service requests are made for each LTDS trip: walking, cycling, public transport, and driving under optimistic, pessimistic, and best guess traffic conditions. For the public transport and driving requests the requested departure time and date are matched by day-of-week and start time to the original trip. The departure time is set for two weeks after the request date to ensure the routes are calculated for typical conditions and do not include planned public transport disruptions or real-time traffic. In total, 801 174 requests are made to the directions service (six requests for each of 133 529 trips).

Figure 6.2 illustrates the four routes generated by the directions service for a single trip (walking, cycling, public transport, and driving under best guess traffic conditions). Only the durations (and not the routes) of the optimistic and pessimistic traffic conditions requests are used in the dataset.

The durations of each separate stage in the public transport route are analysed to calculate the access duration, interchange duration, and on-board durations for bus and rail.

A measure of the traffic variability v for the driving route is calculated as

$$v = \frac{d_o - d_p}{d_t}, \tag{6.1}$$

where d_o, d_p and d_t are the durations for optimistic, pessimistic, and best guess traffic respectively, as predicted by the directions service.

Table 6.3 Grouping of LTDS transport modes for refined dataset.

Category	LTDS mode	Trips
Walking	Walk (/roller-blades / scooters)	40 319
	Total	**40 319**
Cycling	Pedal cycle	3366
	Total	**3366**
PT	Bus (public)	16 643
	Underground	9050
	DLR train	394
	National Rail train	6058
	Tramlink	214
	London Overground	862
	Total	**33 221**
Driving	Car Driver	34 571
	Car passenger	18 354
	Motorcycle rider	469
	Motorcycle passenger	11
	Van (small) driver	920
	Van (small) passenger	198
	Van/lorry (other) driver	492
	Van/lorry (other) passenger	126
	Taxi - London black cab	507
	Taxi - other/minicab	975
	Total	**56 623**
Uncategorised	Bus (school/work)	287
	Dial-a-ride	37
	Coach	156
	Plane/boat/other	116
	No main mode listed	37
	Total	**633**
Overall	Total (inc. uncategorised)	134 162
	Total (exc. uncategorised)	**133 529**

There are 401 trips for which directions were not available from the directions service for one or more of the modes. These trips are excluded from further analysis. This leaves 133 128 trips with predicted routes and durations for all modes.

6.3.3 Screening trips

The next stage is to remove trips which are out of the scope of the study, based on the results from the direction service. Trips which meet the following criteria are retained:

1. the routes for all modes are completely contained within the bounding box of the combination of the London Boroughs, M25 orbital motorway and all TfL stations;
2. all stages within the suggested public transport route use only TfL services and/or stations;
3. the suggested public transport route has at least one public transport step (i.e. it is not purely walking).

In total, 52 042 trips are excluded during screening, leaving 81 086 trips. Criterion 1 excludes 5561 trips, Criterion 2 excludes 16 801 trips, and Criterion 3 excludes 29 680 trips. These criteria are discussed in the sections below.

A summary of the screened trips is given by Fig. 6.4, which shows straight-line trip length histograms for each mode for trips removed during screening. The bins have a fixed width of 50 m. Trips longer than 3 km are not shown on the plot.

The vast majority of the screened trips are short (< 500 m) walking trips, which are mostly excluded through Criterion 3. There are also a number of short driving trips removed, (again mostly due to Criterion 3) as well as a smaller number of longer driving trips (primarily due to Criterion 1 and 2).

Criterion 1

The first criterion is that the routes for all modes are completely contained within the bounding box of the combination of the London Boroughs, M25 orbital motorway, and all TfL stations.

Whilst the LTDS only surveys households within the M25 orbital motorway, it contains all domestic trips made by any member of the household who was in London at any point on the survey date. This includes journeys that start or end outside of the London area. This is illustrated by Fig. 6.5, which shows the straight-line trajectories of all of the trips in the LTDS for the study period, including many trips spread across the UK. It is therefore necessary to extract only the trips which occur fully within the London area.

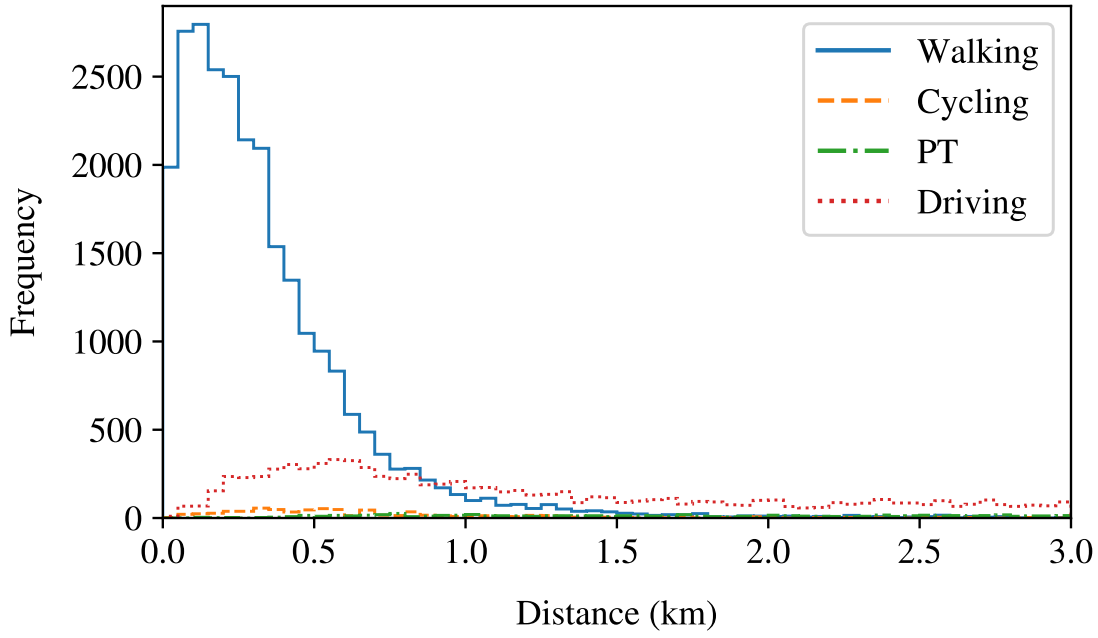


Fig. 6.4 Straight-line trip length histograms for each mode (LTDS), with 50 m bins between 0 km to 3 km for all trips removed during screening.

There are many different definitions of the London area. Three possibilities are illustrated in Fig. 6.6. The formal boundaries of Greater London are those of the 32 London boroughs and City of London, shown in the figure in dark red. The road and public transport networks both extend outside of the area of London Boroughs. The study area for the LTDS is the area inside the M25 orbital motorway, shown on the figure in purple, which does not fully enclose the London Boroughs. The London Underground and Rail network extends outside both the Greater London and M25 boundaries. The outer bounding box of these three areas is used to define the study area for this research.

A spatial search is used to identify and paths which are not fully enclosed by the boundary area. These trips are removed from the dataset.

Criterion 2

The second criterion is that all stages within the suggested public transport route use only TfL services and/or stations. This ensures that (i) the dataset only includes trips served by London's public transport network, and (ii) public transport fares can be accurately predicted.

It is straightforward to check from the directions service routes as to whether London buses and trams are used. However, the solution for rail journeys is more complex. As part

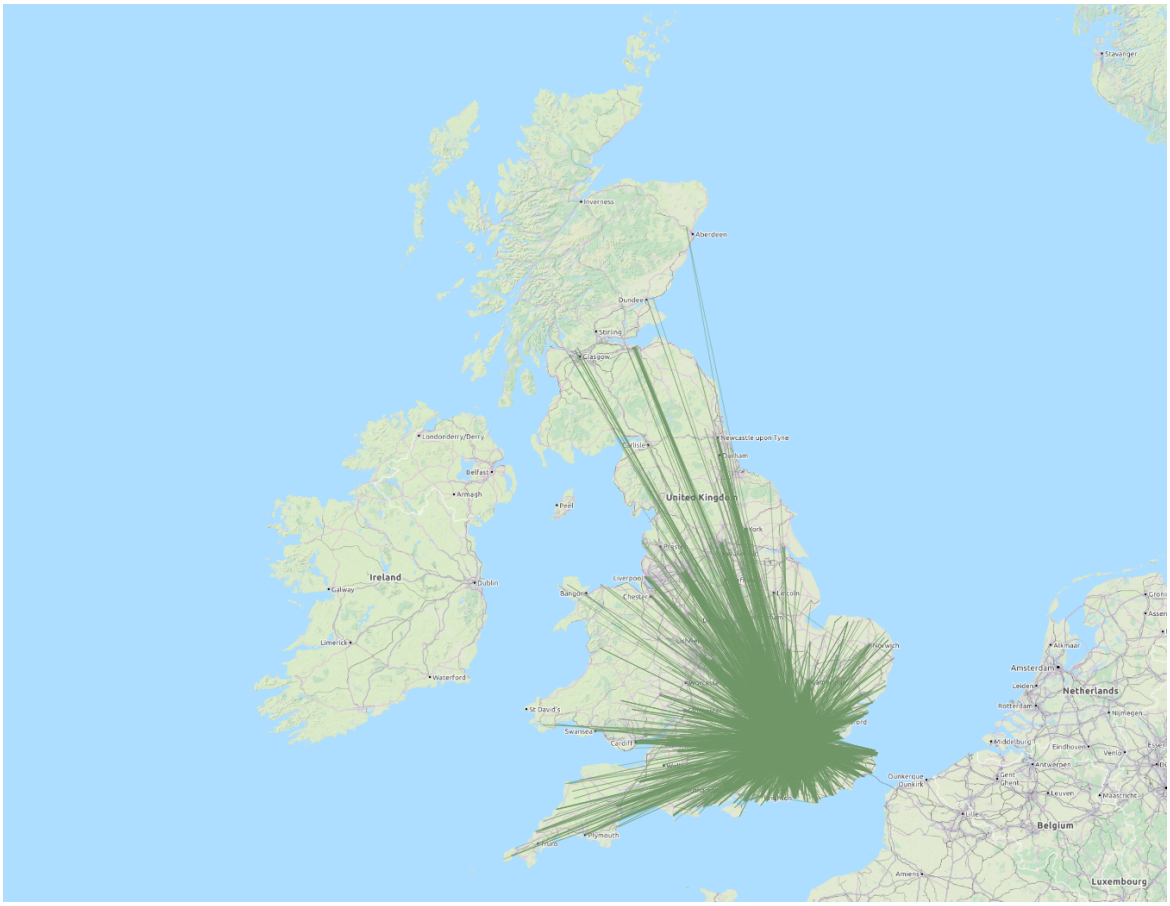


Fig. 6.5 Diagram of straight-line trajectories of all trips in LTDS for study period.

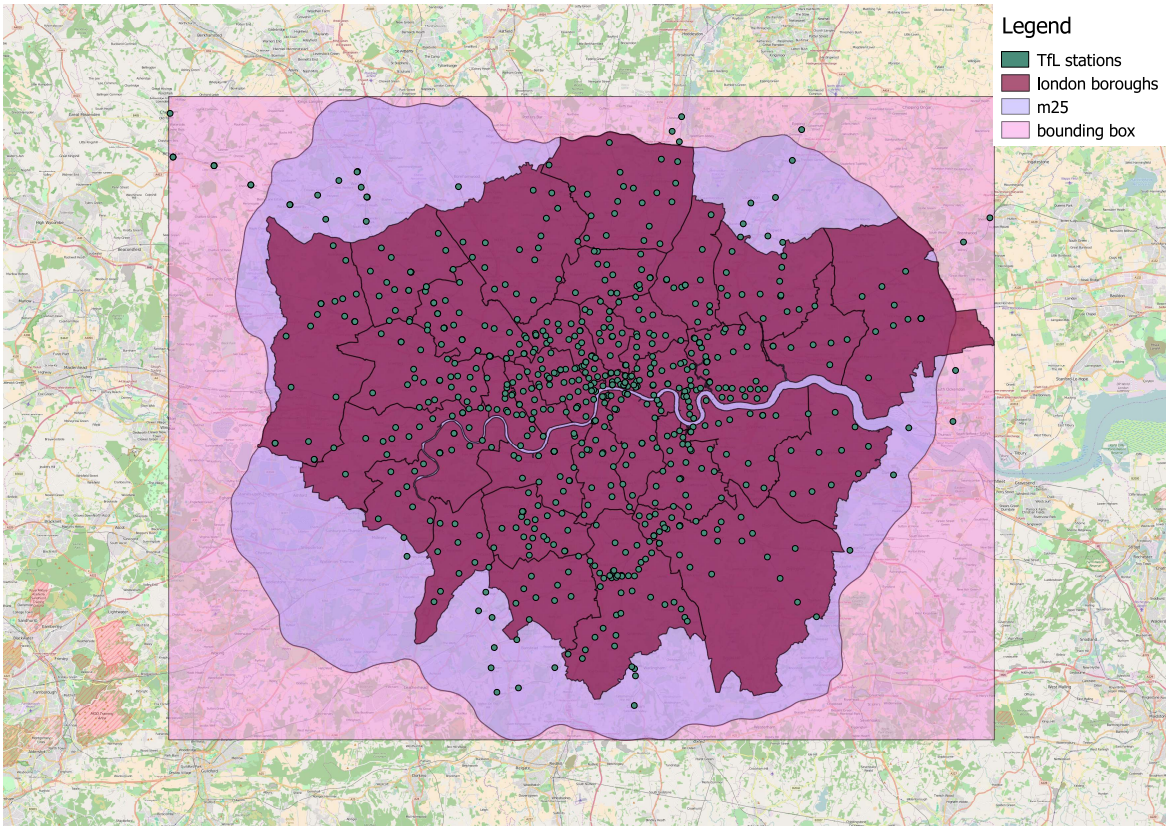


Fig. 6.6 Diagram of different definitions of London area.

of London's zonal fare system, Oyster cards and other TfL tickets can be used on National Rail services within the London zones. As such, it is not sufficient to only include stages using London Underground and the Docklands Light Railway (DLR). Instead, it is necessary to check that TfL tickets are valid on rail services by checking all of the stops are within TfL zones. In order to achieve this, a dataset of all of the stations in London, along with their zone and location, is collated. Using this stations dataset, it is possible omit all public transport trajectories with rail stages which stop at stations outside of TfL's zones.

Criterion 3

The third criterion is that the suggested public transport route has at least one public transport step (i.e. it is not purely walking). This ensures that (i) very short trips, for which walking is the only valid option, are omitted and (ii) there is a valid public transport alternative for all trips in the dataset. This criterion is straightforward to implement directly from the directions service routes.

6.3.4 Adding mode-alternative cost estimates

Finally, each remaining trip is processed to add the public transport and driving costs.

The estimation of travel costs makes full use of the socio-economic and demographic profiles from the LTDS with the corresponding route and duration data from the directions service. The costs are closely tailored to accurately represent public transport fares, fuel prices, and the Central London Congestion Charge.

6.3.4.1 Public transport fares

Public transport fares are determined for single trips using Oyster card/contactless payment, based on the TfL's pricing scheme. A total fare is calculated for each trip, through analysing the individual stages in the public transport route returned by the directions service.

The fares for buses and trams are charged per boarding, independent of trip length (the trips in the dataset took place before the new bus *Hopper fares* were introduced in 2016). There is no peak/off-peak pricing or zoning of fares.

Bus fares are therefore only dependent on the three bus fare levels: full; half (for reduced bus fare holders or children aged 16-18 who live outside London); or free (for all children under 16, children aged 16-18 living in London, TfL staff, Police, or national concession buspass holders). The bus fare levels are determined for each passenger from the LTDS information.

Fares for National Rail, London Underground, London Overground and the Docklands Light Railway are more complex, and are dependent on four variables: fare zones (zones 1-9 plus extension fares for specific stations outside these zones); services used (7 fare-types for different rail services); time of day (peak from 06:30-09:30 and 16:30-19:30 on weekdays except bank holidays); and rail fare-type (normal, child under 16, 16+, disabled persons' railcard, other discount railcards, free).

The first three variables are determined from details of the relevant stage from the suggested public transport route. For instance, the peak/off-peak classification is determined from the start time of each corresponding stage (recalling that the trip start time is matched to that of the original LTDS trip). As with the bus fares, the rail fare-type is determined for each passenger from the LTDS information.

The TfL Unified API (Transport for London 2016) has been used to collect the correct fares across the variants above. The station names returned by the directions service are not an exact match to those used by the TfL Unified API. In order to ensure the correct fare is selected, a spatial search is conducted for each interchange location from the directions service public transport routes against the stations in the National Public Transport Access Nodes (NaPTAN) database (Slevin and Griffin 2016). This allows the unique NaPTAN code for each interchange station to be identified, for use with the TfL Unified API. As with any major metropolitan area, there are some exceptions to address within the train fare scheme.

First, there are several pairs of stations on the TfL network where a free walking-interchange between lines is permitted, when exiting from one station and re-entering at a paired station. As such, the corresponding separate rail stages should be combined under a single fare. To allow for this, a dataset of all free transfer station pairs on the TfL network has been assembled to identify routes where separate services constitute one continuous journey. Each public transport route is checked against this dataset in order to join the separate stages in the cost model.

Second, there is often more than one route available between two stations, each with different fares. In the real world, these fares are determined either by tapping the contactless payment card on a special interchange card reader at certain stations, or by exiting and re-entering a station with a free-interchange. If more than one fare is available, the TfL Unified API returns a list detailing the available fares and the required transfer stations. To ensure the correct fare is assigned, a list of transfer stations from the public transport route is assembled, and this is used to determine if the required transfer station is passed through.

For complex, multi-legged journeys, the total public transport fare is calculated as the sum of the individual fares for each separate part. A new fare is recorded each time a bus

is boarded, the first time a station is entered, and each time a journey exits and re-enters a station without a free interchange.

6.3.4.2 Driving

The driving costs consist of the operating cost and the congestion charge cost. Parking costs are not directly included here as there is no data to determine parking rates at the destination. Instead, variability of mode-choice due to parking costs is left to be included in the model error terms (i.e. is accounted for as part of the residual in model estimation), following Jin, Williams, and Shahkarami (2002). The operating cost is calculated using the Vehicle Operating Costs (VOCs) formula presented in the UK DfT Transport Analysis Guidance (WebTAG) (Department for Transport 2014), with the fuel-type determined by the vehicle(s) available to the household. The lowest cost fuel-type is assumed if there are more than one vehicle available on the travel date. If the household owns no vehicle, an average fuel-type is used.

The congestion charge is included if the driving route crosses the congestion charge zone. The charge is ignored for journeys on weekends, bank holidays, and outside hours, as well as for other exemptions and zero-charge vehicles (determined from the LTDS data).

6.4 Processed dataset

The finished dataset contains entries for 81 086 journeys across three years (2012/13-2014/15). The journeys are made by a total of 31 954 individuals within 17 616 households. The dataset is naturally imbalanced, with different ratios of trips for walking (17.6 %); cycling (3.0 %); public transport (35.3 %) and driving (44.2 %).

The dataset is openly available under the Creative Commons Attribution (CC-BY) licence from the website of a journal paper where this work is published (Hillel, Elshafie, and Jin 2018), with kind permission from TfL.

Figure 6.7 is generated by projecting the path for the driving option (under best guess traffic) for all trips in the dataset, showing both the geographical coverage and detail within the dataset through recreating the shape of London's road network. The trips are projected at high transparency, so the line intensity represents how frequently the underlying road is used in the dataset.

A summary of the attributes for each trip in the dataset is given in Table 6.4. As discussed in Chapter 5, the base dataset includes only the attributes listed under socio-economic and demographic profile data (from the LTDS), and the choice-set dataset additionally includes the mode-alternative attributes (from the directions service and cost models). Summary

Fig. 6.7 Diagram of driving paths in study dataset.

statistics for the continuous attributes in the dataset are given in Table 6.5 (categorical variables and times/dates are omitted).

As discussed in Chapter 5, the dataset is split into train and test sets by year. The train dataset contains the first two years of data (2012/13 and 2013/14), and the test dataset contains the final year of data (2014/15). The train dataset contains 54 766 trips, and the test set contains 26 320 trips. This is equivalent to a 68:32 train-test split.

The following sections present preliminary investigations using the dataset, covering: (1) the correlations between trip length and mode choice (Section 6.4.1); and (2) the impacts of using trip-wise sampling with hierarchical data (Section 6.4.2).

6.4.1 Correlations between trip length and mode choice

This section investigates the correlations between trip length and mode choice. Figure 6.8 shows straight-line trip length histograms for each mode for all trips in the processed dataset. The bins have a fixed width of 200 m. Trips over 15 km in length are not shown in the plot.

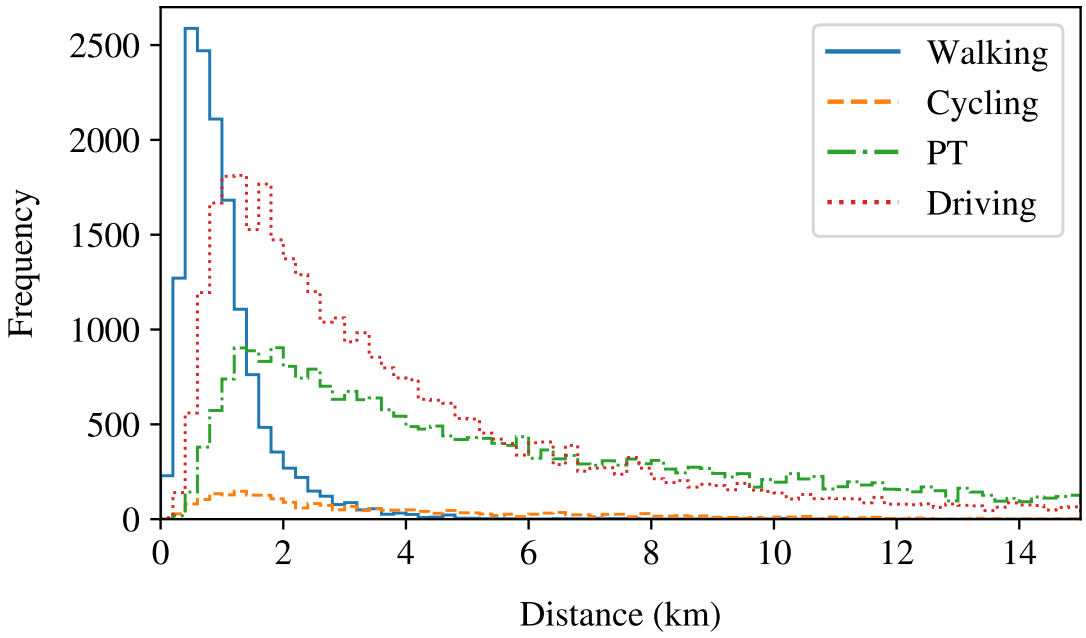


Fig. 6.8 Straight-line trip length histograms for each mode (LTDS), with 200 m bins between 0 km to 15 km for processed dataset.

Figure 6.8 shows that, whilst driving is the most common mode in the dataset, the frequency of use of each mode is dependent on the trip distance. Walking is the most

Table 6.4 Study dataset attributes and description.

Type	Group	Attribute	Description
ID and context	Context	trip_id	Unique ID for each trip
		travel_mode	Mode of travel chosen for LTDS trip
		ori_postcode	Origin postcode of the trip
		desti_postcode	Destination postcode of the trip
Socio-economic and demographic profile (from LTDS)	Categorical	purpose	Journey purpose for trip (B, HBW, HBE, HBO, NHBO)
		fare_type	Public transport fare-type of passenger (16+, child, disabled, free, full)
		fuel_type	Fuel-type of passenger's vehicle (Diesel/petrol/hybrid car or diesel/petrol LGV)
		driving_license	Whether the traveller has a driving license
		sex	Gender of passenger
	Ordered numerical	age	Age of passenger in years
		distance	Straight-line trip distance
		car_ownership	Car ownership of household (no cars, less than one car per adult, one or more cars per adult)
		bus_scale	Percentage of the full bus-fare paid by the passenger
		start_time	Start time of trip
		day_of_week	Day of the week of travel
		travel_month	Month of year of travel
Choice set data (from directions service and cost models)	Walking	dur_walking	Duration of walking route
	Cycling	dur_cycling	Duration of cycling route
	Public transport	dur_pt:rail	Duration spent on rail services on public transport route
		dur_pt:bus	Duration spent on bus services on public transport route
		dur_pt:access	Duration walking to/from first/last stop on public transport route
		dur_pt: interchange	Total duration of public transport interchanges
		dur_pt:total	Duration of whole public transport route (dur_pt:access + dur_pt:rail + dur_pt:bus + dur_pt:interchange)
		cost_pt	Cost of whole public transport route
		n_ints	Total number of public transport interchanges (rail-rail, bus-bus, bus-rail, rail-bus)
	Driving	dur_driving	Duration of driving route
		cost_driving: VOC	Vehicle operation costs of driving route
		cost_driving: con_charge	Congestion charge for driving route
		traffic_var	Traffic variability (shown in Eq. (6.1))

Table 6.5 Summary statistics for continuous attributes in dataset.

Attribute	Unit	mean	std	min	25 %	50 %	75 %	max
age	years	39.362	19.227	5	25	38	52	99
distance	km	4.605	4.782	0.077	1.309	2.814	6.175	40.941
dur_walking	hours	1.129	1.118	0.025	0.351	0.723	1.514	9.278
dur_cycling	hours	0.362	0.352	0.006	0.117	0.232	0.385	3.052
dur_pt:rail	hours	0.090	0.177	0	0	0	0.100	1.367
dur_pt:bus	hours	0.172	0.190	0	0.033	0.112	0.250	2.147
dur_pt:access	hours	0.160	0.092	0	0.093	0.144	0.211	1.189
dur_pt:interchange	hours	0.044	0.078	0	0	0	0.083	0.865
dur_pt:total	hours	0.367	0.310	7.500×10^{-3}	0.226	0.389	0.643	2.735
cost_pt	£	1.563	1.535	0	0	1.500	2.300	13.500
n_ints	-	0.369	0.619	0	0	0	1	4
dur_driving	hours	0.282	0.252	2.780×10^{-4}	0.108	0.192	0.369	2.061
traffic_var	-	0.336	0.201	0	0.173	0.306	0.382	1.250
cost_driving:VOC	£	0.821	0.808	0	0.280	0.530	1.080	9.720
cost_driving:con_charge	£	0.992	3.046	0	0	0	0	10.500

frequently used mode for trips under 1 km. Driving is the most frequently used mode for trips from 1 km to 6 km. For trips between 6 km to 8 km there are a similar number of driving and public transport trips. Finally, for trips over 8 km public transport is most frequently used.

From the histogram, it can be seen that walking has a modal peak for trips around 500 m, driving around 1.1 km, cycling around 1.5 km, and public transport around 1.8 km. The rate of walking trips sharply declines for trips greater than 800 m, and walking is less frequently used then public transport for trips greater than 1.6 km. Cycling is infrequently used at all distances compared to driving and public transport, but is more frequently chosen over walking for trips greater than 3.6 km.

6.4.2 Trip-wise sampling of hierarchical data

In order to investigate the possible effects of using random trip-wise sampling with hierarchical data, corresponding trips within the training dataset are analysed, based on their origins and destinations. Three different cases for corresponding trips are identified, which each arise from one of the hierarchies identified by Q2g in Section 3.3.3: (i) return trips, with reversed origin and destination, made by the same individual - *tour-trip*; (ii) repeated trips, with matching origin and destination, made by the same individual - *person-tour*; and (iii) shared trips, with matching origin and destination, made by members of the same household - *household-person*.

Whilst corresponding trips are not the only source of possible correlations between hierarchical groups in trip data (see the other examples in Q2g in Section 3.3.3), they represent a definite and quantifiable dependence between trips.

Corresponding trips occur in pairs or sets (e.g. an outward leg and a return leg form a return pair of trips). These sets are identified in the dataset through targeted searches of the trip origins, destinations, and household or person ID. The total number of corresponding trips can then be determined by summing the size of the sets. Finally, the number of corresponding trips with matching modes can be counted, by repeating the search grouped by transport mode.

Table 6.6 shows, for each type of corresponding trip described above, the number of corresponding sets; trips; and the proportion of corresponding trips made with matching transport mode. Additionally, the table shows these figures for all corresponding trips (i.e. all trips which are part of one or more return, repeated, or shared sets). Trips may belong to multiple corresponding groups, e.g. a trip may be both a return trip made by the same individual and shared by multiple members of a household. As such, the total number of all corresponding trips is less than the sum of the separate types.

Table 6.6 Corresponding trips in train dataset, for return; repeated; and shared trips.

Type	**Pairs/sets**	**No. Trips**	**No. Trips matching mode**	**Proportion matching mode**
Return	15 605	32 471	30 898	95.2 %
Repeated	1315	2711	2496	92.1 %
Shared	8541	20 623	20 051	97.2 %
All	15 814	40 520	39 357	97.1 %

Whilst there are different numbers of corresponding trips of each type, all three types individually represent a substantial proportion of the data. As such, all three hierarchies identified by Q2g in Section 3.3.3 are problematic if improper sampling methods are used.

The majority of corresponding trips in the dataset are return trips. However, shared trips occur in larger sets (2.4 trips per set on average) and have a higher proportion of matching modes. As such, they are more likely to result in matching trips across train/test samples if trip-wise sampling is used.

In total, of the 54 766 trips in the training dataset, 40 520 (74.0 %) belong to pairs or sets of return; repeated; or shared trips. Of these, the vast majority (97.1 %) are made by the same transport mode.

If data is sampled trip-wise into train and test data, there will therefore be a significant proportion of trips in the test data which have corresponding matching trips in the remaining data, made by the same mode. This proportion can be investigated using Monte-Carlo simulations. Two scenarios are investigated, holdout validation with a 70:30 split, and

10-fold cross-validation. These represent the most commonly used ratio and number of folds for holdout validation and k-fold cross-validation respectively (see Section 3.3.4).

To simulate 70:30 holdout validation, random 30 % samples (of 16430 trips) are selected trip-wise from the dataset. The number of trips with one or more corresponding matching trips in the remaining (training) data are counted. The estimation is averaged over 10000 repetitions. An average of 9518.1 trips, or 57.9 % of the test data, has corresponding matching trips made by the same mode in the training data.

The ratio of corresponding matching trips is higher for k-fold cross-validation. To simulate 10-fold cross-validation, random 10 % samples (of 5477 trips) are selected. Again, the average number of corresponding matching trips over 10000 repetitions is recorded. On average 3694.4 trips, or 67.5 % of each fold, has corresponding matching trips with the same transport mode in the remaining data.

Corresponding trips will have highly correlated attributes. In particular, the straight-line distance will be exactly the same for each trip in a corresponding set, and the durations and costs for each mode will be matching or very similar. Algorithms can therefore overfit to this data, returning the matching mode for each unique distance or set of durations. For example, decision trees can assign each unique straight-line distance to an individual leaf node with the corresponding transport mode, therefore increasing the perceived performance of the algorithm.

The proportion of corresponding trips will vary across datasets, depending on the average sizes of each of the hierarchical groupings (household-person, person-tour, and tour-trip). The larger the hierarchical groupings, the higher the likely ratio of corresponding matching trips. In particular, the longer the period of the trip diary, the more tours each individual will record in the trip diary. The LTDS data is a single day trip diary, where each person makes an average of 2.8 trips. The ratio of repeated trips in each fold is likely to be higher in multi-day trip diaries, where each individual reports more trips.

These issues can be solved by using grouped *household-wise* sampling. There are only 42 sets of repeated trips in the whole dataset which are between the same (or reversed) origin and destination made by members of different households. Of these, only 29 pairs/sets share the same mode choice across households. In total, 87 repeated trips share the same mode across households, representing less than 0.2 % of the training data. When performing 10-fold cross-validation sampling grouped by household, so that all trips made by one household will occur only in single fold, there are an average of 5.2 matching trips with matching mode between each fold and the remaining data. This represents less than 0.1 % of the trips in each fold.

6.5 Summary

This chapter introduces a new framework for recreating passenger mode choice-sets which has been developed for this research. Crucially, by recreating passenger mode choice sets for observed trip data, the framework in this chapter allows random utility models to be fitted and compared to Machine Learning (ML) methods (see Chapter 7). The framework has been implemented within an automated process, which can be adapted for the fast assembly of datasets from historic trip data from any geographical region.

The framework presented in this chapter has been used to create a dataset covering historic trips in the London area. This dataset represents the most comprehensive and closely tailored travel dataset for estimating travel choices in a major metropolitan area. The dataset is openly available online, with kind permission from TfL.

Two preliminary investigations are presented that make use of the dataset to investigate the effect of trip length on mode choice and the impacts of using random trip-wise sampling for hierarchical data. The latter investigation establishes that on average for 10-fold cross-validation 67.5 % of the trips in each fold have corresponding matching trips made by the same mode in the remaining data. As such, performance estimates using trip-wise sampling for similar hierarchical data are likely to be heavily positively biased, particularly for classifiers with high variance which can easily overfit to training data. Household-wise sampling is tested as an alternative sampling method, which is shown to vastly reduce the level of corresponding matching trips to less than 0.1 % of each fold. This shows household-wise sampling to be an appropriate method for validating classifiers trained on household travel diaries.

Chapter 7

Results and discussion

7.1 Overview

This chapter presents the experimental results, following the methodology outlined in Chapter 5. Firstly, Section 7.2 presents the results for the random utility approach. This represents the application of state-of-practice techniques to the new data developed for this thesis. Multinomial Logit (MNL) models are initially trained to identify the form of the utility specifications (Section 7.2.1), and then Nested Logit (NL) and Cross-Nested Logit (CNL) models are used to investigate nesting structures (Section 7.2.1.1). The highest performing NL model is used as the benchmark Random Utility Model (RUM), to which the Machine Learning (ML) classifiers are compared in Section 7.3.1.

Next, Section 7.3 presents the results of the ML investigations. This includes the comparative study of ML classifiers (Section 7.3.1), and investigations into sampling methods for hierarchical data (Section 7.3.2) and the impacts of adding mode-alternative attributes to the input data (Section 7.3.3). These investigations address the technical limitations of the existing ML research, and as such represent a rigorous and systematic study of ML classification techniques for mode-choice prediction.

Section 7.4 presents the results of the assisted specification approach, where the highest performing ML classifier is used to inform the utility specification structure in a RUM. This approach attempts to combine the predictive power and flexibility of ML classifiers, with the interpretability and strong behavioural foundation of the random utility approach.

Finally, Section 7.5 summarises the findings of the experimental results.

In order to aid the reader in navigating and understanding the results, the main findings and the figures/tables which illustrate them are summarised as follows:

1. the Gradient Boosting Decision Trees (GBDT) model is the highest performing classifier for the mode choice prediction task: Figures 7.1 and 7.2;

2. trip-wise sampling with hierarchical data can result in substantial overestimates of model performance and poor performing hyper-parameters being selected, both of which are successfully addressed by using grouped household-wise sampling: Tables 7.10 and 7.11 and Fig. 7.7;

3. the data generation framework to add mode-alternative attributes to the feature vectors substantially improves model performance whilst using the same original input data (i.e. it represents a *free* performance improvement from the point of view of data requirements): Figure 7.9;

4. the overall performance improvement from the data generation framework is greater than that for the choice of model algorithm: Figures 7.2 and 7.9

5. the assisted specification approach can be used to create RUMs which outperform all but the highest performing ML classifier whilst maintaining a robust and interpretable utility specification: Figure 7.14.

7.2 Random utility approach

This section presents the results for the RUMs fitted to the study dataset. Firstly, Section 7.2.1 documents the three MNL models described in Section 5.4. Next, Section 7.2.1.1 documents the investigations into nested (NL and CNL) model structures, which use the full combined MNL structure identified in Section 7.2.1 as a basis. Finally, Section 7.2.2 compares the fit statistics and holdout validation results for each classifier and identifies the benchmark RUM to which the ML classifiers are compared in Section 7.3.1. This represents the application of state-of-practice techniques to the new data developed for this thesis.

The fitted parameters for each model in this section are presented in Appendix B. All parameters follow the same naming scheme:

- Alternative Specific Constants (ASCs) - ASC_MODE;

- β parameters - B_VARIABLE_CATEGORY_MODE (where the CATEGORY is given only for categorical variables or the public transport duration breakdowns);

- Nesting parameter (μ) for NL models - MU_NESTNAME.

All models are fitted to the training data only, so that the number of observations used during fitting is 54 766. This gives a null log likelihood (with all parameters set to zero) of −75 921.80.

7.2.1 Utility function specification

The first three RUMs are MNLs (non-nested, with independent error terms). The three models are fitted using the following variables respectively: (1) only the mode-alternative attributes, (2) only the socio-economic and trip profile from the London Travel Demand Survey (LTDS), and (3) the combined mode-alternative attributes and the socio-economic/trip profile.

Table B.1 in Appendix B.1 shows the model parameter estimates for the mode-alternative attribute model. The initial hypothesised model contains 15 parameters. During utility function optimisation, two β parameters (walking interchange time and number of interchanges, both for public transport) are found not to be significantly different from zero (i.e. $p > 0.05$) and so are removed, leaving 13 parameters.

The signs of the remaining parameters are consistent with behavioural theory (e.g. negative utility for increasing trip duration and cost). For example, the Value of Time (VoT) can be calculated by dividing the time parameter for a given mode by the cost parameter

$$\text{VOT} = \frac{\beta_{\text{time}}}{\beta_{\text{cost}}} \tag{7.1}$$

The different cost parameters for driving and public transport give different VoTs for the different transport modes. The driving cost parameter suggests values of 83.00, 47.18, 19.13, 16.70, and 40 £/h for walking, cycling, on-board bus, on-board rail, and driving respectively. The public transport cost parameter suggests corresponding values of 43.40, 24.67, 10.00, 8.73, and 20.91 £/h. The large discrepancies in these values are likely due to endogeneity from not including variables correlated with both trip costs and the output mode choice. In particular, the trip distance is omitted, which is both highly correlated with trip cost and mode choice (longer journeys, which have higher Vehicle Operating Costs (VOCs), are more likely to be made by driving).

The parameter estimates for the socio-economic and trip profile model are given in Table B.2. From the initial hypothesised model, which has 54 parameters, eight parameters are removed due to not being significantly different from zero: four for cycling (age - child, vehicle ownership - 1, vehicle ownership - 2, day - week); three for public transport (departure - AM peak, departure - inter-peak, winter); and one for driving (purpose - home-based education).

Again, the signs of the 46 remaining parameters are consistent with behavioural theory, e.g. positive utility for increasing trip distance for the non-walking modes, and positive utility for driving/negative utility for public transport with increasing vehicle ownership.

The parameter values show that age, day of week, departure time, driving licence ownership, gender, journey purpose, vehicle ownership, and time of year all have a significant impact on mode choice.

Finally, the parameter estimates for the combined MNL model are given in Table B.3. From the initial hypothesised model, which contains 65 parameters, 10 are removed for not being significantly different from zero: four for cycling (age - child, day - weekday, day - Saturday, purpose - home-based education, purpose - home-based other); three for public transport (departure - AM peak, departure - inter-peak, winter); and three for driving (day - Saturday, departure - AM peak, purpose - home-based other). Furthermore, the two vehicle ownership parameters for cycling are found to not be significantly different from one another, and so are combined into a single parameter. This leaves the 54 parameters shown in Table B.3.

Adding the additional socio-economic and trip profile parameters (including trip distance) results in a much closer agreement between the cost parameters for public transport and driving in the combined model (-0.0925 and -0.118 respectively) compared to the mode-alternative attribute model (-0.197 and -0.103 respectively). The smaller cost parameter for public transport trips in the combined model is possibly due to endogeneity associated with not having access to information related to daily usage caps and/or period travel cards in the cost model. Passengers who reach the daily Oyster usage fare cap or have a period travel card have no incremental cost for additional public transport journeys, whilst the dataset assumes the single Oyster card fare. As such, these passengers may appear more cost-insensitive when choosing public transport, bringing the public transport cost parameter down.

7.2.1.1 Nested and cross-nested logit

The optimised utility specifications from the combined mode-alternative attributes and LTDS socio-economic/trip profile model (shown in Table B.3) are used to investigate possible nesting structures for NL and CNL models.

All possible nesting structures are tested, as shown in Table 7.1. Two nesting structures are found to be significant: (1) Walk-PT, referred to as *flexible* modes, as they do not require a vehicle or bicycle (and so can be combined in one tour); (2) PT-Drive, referred to as *powered* modes. All other nested structures are found to be insignificant (μ not significantly different from 1).

Table 7.1 Tested nested logit structures.

NL structure	**Significant μ**
Walk-Cycle	
Walk-PT	✓
Walk-Drive	
Cycle-PT	
Cycle-Drive	
PT-Drive	✓
Walk-Cycle-PT	
Walk-Cycle-Drive	
Walk-PT-Drive	
Cycle-PT-Drive	
Walk-Cycle/PT-Drive	
Walk-PT/Cycle-Drive	
Walk-Drive/Cycle-PT	

The parameter estimates for the two significant NL structures are given in Tables B.4 and B.5. Note that the t-statistic for the μ parameters, marked with *, are calculated from one (and not zero, as with the ASC and β parameters).

For the flexible modes NL model, all parameters from the combined MNL model remain significant, alongside the additional μ parameter, resulting in a total of 55 parameters. All parameter signs remain consistent with behavioural theory. The driving cost parameter for the combined model suggests VoTs of 43.00, 18.08, 17.75, 12.42, and 33.83 £/h for walking, cycling, on-board bus, on-board-rail, and driving respectively, which are consistent with values obtained in similar studies. The public transport cost parameter results in higher estimates for VoT, however this may be due to the endogeneity from not having access to daily usage caps/period travelcard information, as discussed above.

In the powered modes NL model, the combined vehicle ownership cycling parameter is no longer significantly different from zero, and so is removed, leaving 54 parameters.

A CNL model including both the flexible modes and powered modes nests is tested. However, the powered modes nest is found to be insignificant when the flexible modes nest is present, and so the CNL reduces to the flexible modes NL model.

7.2.2 Random utility results

Table 7.2 shows the details and results for the RUMs described in Section 7.2.1. The flexible modes NL model is the highest performing of the tested random utility classifiers, both in terms of Akaike Information Criterion (AIC) during fitting, as well as out-of-sample test

performance. As such, this model is used as the benchmark RUM to be compared to the ML classifiers in Section 7.3.

Table 7.2 Results for RUMs

Model	Description
1	MNL with mode-alternative attributes only
2	MNL with socio-economic/trip profile only
3	MNL with combined data (mode-alternatives and socio-economic/trip profile)
4	NL with combined data - Flexible modes (Walk-PT)
5	NL with combined data - Powered modes (PT-Drive)

			Fit		Train		Test	
Model	**Params**	**Fit time**	**LL**	**AIC**	**GMPCA**	**DCA**	**GMPCA**	**DCA**
1	13	00:05	-45585.3	91196.56	0.4350	0.6410	0.4274	0.6398
2	46	01:17	-40622.1	81336.12	0.4763	0.7056	0.4650	0.6964
3	54	01:40	-37281.8	74671.53	0.5062	0.7390	0.4954	0.7297
4	55	05:29	**-37195.1**	**74500.24**	**0.5070**	0.7386	**0.4960**	**0.7303**
5	54	04:46	-37259.9	74627.74	0.5064	**0.7395**	0.4957	0.7302

All models significantly outperform a uniform prior (Geometric Mean Probability of Correct Assignment (GMPCA) of 0.25) and a balanced prior (GMPCA of 0.317).

There is a great reduction in both the fit AIC and the out-of-sample GMPCA between Models 2 and 3. This shows, alongside the significance of the corresponding parameters, that adding mode-alternative attributes to the utility specification significantly improves the performance of the mode choice classifiers.

For all of the RUMs, the out-of-sample test scores are lower than the in-sample train scores. This implies that all classifiers have partially overfit to the training data, and as such, the in-sample performance estimates are optimistically biased. However, the differences in performance are small, showing that all models are able to generalise and predict trips from a future year of data.

7.3 Machine learning investigations

This section presents the results of the ML investigations in the thesis. Section 7.3.1 compares the performance of six ML classifiers with the RUM benchmark identified in Section 7.2. Next, Section 7.3.2 investigates sampling methods for hierarchical data by comparing the original models optimised using household-wise sampling with an alternative set of models trained using trip-wise sampling. Finally, Section 7.3.3 uses the highest performing classifier from Section 7.3.1 in order to investigate the impacts of adding mode-alternative attributes to

the data for ML models, by training a new model using only the original data (without the mode-alternative attributes).

These investigations address the technical limitations of the existing ML research, and as such represent a rigorous and systematic study of ML classification techniques for mode-choice prediction.

7.3.1 Comparative study of ML classifiers

This section compares the performance of six ML classifiers, out of Logistic Regression (LR), Feed-Forward Neural Networks (FFNNs), Random Forests (RFs), Extremely randomised Trees (ET), GBDT, and Support Vector Machines (SVMs); with the flexible modes NL RUM introduced in Section 7.2.1.1. The NL model represents a benchmark performance achieved by the state-of-practice mode choice modelling techniques on the study dataset.

The ML classifiers are optimised using the modelling framework presented in Section 5.3. Selected hyper-parameter values for each algorithm are given in Appendix A.2.1.

Over the following sections, these optimised models are used to investigate holdout validation performance of the different classifiers (Section 7.3.1.1), model significance testing (Section 7.3.1.2), the differences between probabilistic simulation vs discrete classification (Section 7.3.1.3), the impacts of different validation schemes (Section 7.3.1.4), the importance of model optimisation (Section 7.3.1.5), and the suitability of the different models for practical use in mode choice simulation (Section 7.3.1.6).

7.3.1.1 Holdout validation performance

The holdout validation results (train on 2012/13-2013/14, test on 2014/15) of the comparative study are shown in Table 7.3. This table shows the holdout validation GMPCA, Arithmetic Mean Probability of Correct Assignment (AMPCA), and Discrete Classification Accuracy (DCA) for each optimised model.

This table is summarised visually in Fig. 7.1, which shows the relative differences in the performance metrics of each of the ML classifiers compared to the baseline RUM. The plot is ordered according to the GMPCA of the corresponding model.

The GBDT model achieves the best holdout test-performance in all three metrics, compared to the other classifiers. This is followed by the two remaining Ensemble Learning (EL) models (RF and ET), which have very close performance to each other across all three metrics. The flexibility and efficiency of Decision Tree (DT) ensembles for modelling non-linear relations in the data explain their relative advantages over the other families of classification algorithms. Each tree in the optimised RF and ET models has a maximum

Table 7.3 Holdout-validation results for optimised ML classifiers.

		Test - score			Test - rank		
		GMPCA	**AMPCA**	**DCA**	**GMPCA**	**AMPCA**	**DCA**
Linear	**RUM**	0.4960	0.6210	0.7303	7	6	7
	LR	0.5000	0.6246	0.7356	6	5	4
NN	**FFNN**	0.5025	0.6207	0.7347	4	7	5
EL	**RF**	0.5082	0.6294	0.7416	2	2	2
	ET	0.5067	0.6271	0.7412	3	3	3
	GBDT	0.5215	0.6490	0.7484	**1**	**1**	**1**
SVM	**SVM**	0.5006	0.6249	0.7316	5	4	6

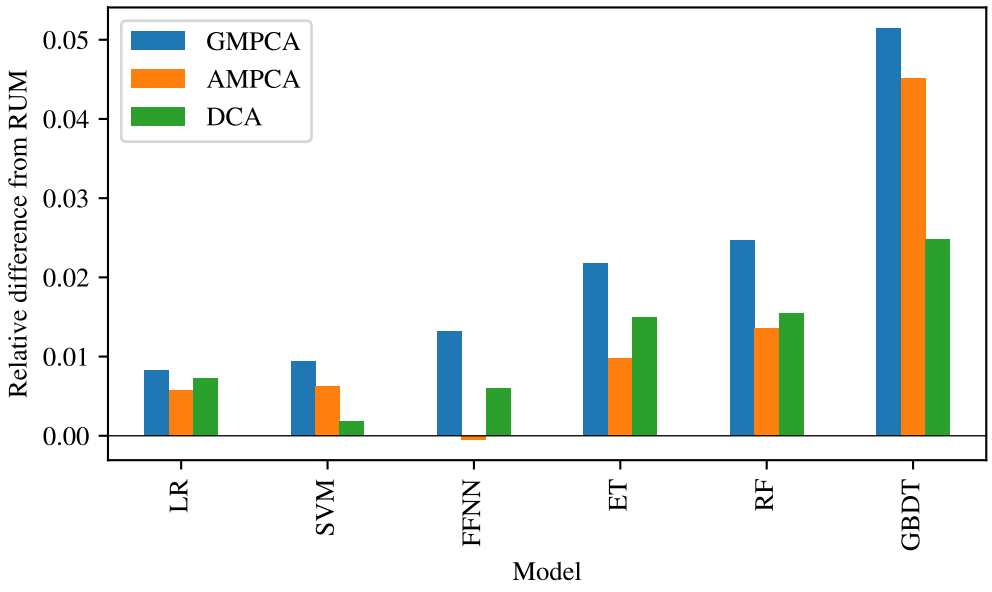


Fig. 7.1 Bar chart of relative differences in performance metrics of ML classifiers compared to baseline RUM (Refer to Table 7.3).

depth of 13 splits, with a maximum depth of six splits in the GBDT model (see Tables A.8 to A.10). This allows these classifiers to model very high order input feature interactions. Whilst the interactions in each individual tree are limited to successive binary splits, all three EL models use ensembles of over 1400 decision trees. When combining the results of each split across all of the trees in the ensemble, this allows complex non-linear relationships to be approximated.

Whilst the GBDT classifier outperforms the other EL classifiers across all metrics, the relative gap is far larger for the probabilistic metrics (GMPCA and AMPCA) than for DCA. This is due to gradient boosting making use of sequential regression trees, with each tree predicting the residual of the previous trees in the ensemble. As such, the trees in the GBDT model directly evaluate the probability distribution $P(x_n)$ of each element n over the J classes. This is more efficient for probabilistic classification than the technique used in bagging, where each individual tree in the ensemble attempts to discretely classify each element n directly into its class i_n, and the ratios of each class are averaged across all child-nodes for all trees in the ensemble to generate the output choice probabilities. A secondary performance advantage of fitting sequential trees over bagging is the ability to set the ensemble size adaptively using the *early stopping rounds* variable (see Section 5.5.1). Again, this is more efficient than bagging, where the size of the ensemble must be identified using a traditional hyper-parameter search.

Of the remaining models, the FFNN achieves the highest GMPCA (though the lowest AMPCA). As discussed in Section 2.3.2, Artificial Neural Networks (ANNs) are highly flexible, and a FFNN of sufficient complexity can (in theory) represent any function. However, complex FFNNs have many parameters to estimate, and as such require a lot of data to discover the underlying relationships between the input features and output. This is in contrast to EL algorithms based on DTs, which can fit to data very efficiently. This explains the performance difference between the EL models and the FFNN. There may be alternative neural network structures which allow the model to exploit data of this type more efficiently than the implemented FFNN, by giving the network a *head start* regarding the structure of the data (considering, for example, the successes of Convolutional Neural Networks (CNNs) for image classification).

The LR and SVM follow the FFNN classifier in terms of the GMPCA, also achieving similar performance to each other. This is likely due to the optimised SVM model making use of a linear kernel (see Table A.11). As such, the optimised SVM is algorithmically very similar to the LR model. The primary differences between the two models are how choice probabilities are calculated, and how regularisation is applied. However, the implementation for the LR algorithm is much more efficient than the SVM algorithm, in particular considering

the extra steps needed to calibrate the choice probabilities for the SVM model. Whilst optimal hyper-parameter values were found for the SVM classifier using a 10 % data sample for the sequential search, fitting the optimised SVM to the training data takes a very long time (see Section 7.3.1.6). This is due to the computational complexity of the SVM model when used to predict probability-like values, as discussed in Section 5.5.1. The computational cost of fitting the proposed SVM model rules it out of practical use for datasets of this complexity and scale. Bootstrapping results were therefore not obtained for the SVM model (see Section 7.3.1.2).

The LR model used as a ML classifier marginally outperforms the structured baseline RUM in all three metrics, with a relative improvement of under 1 %. This is due to the greater flexibility in the LR model. For example, in the LR model, every feature is included indiscriminately for every mode, e.g. the predicted cycling duration is an input for the score for all four modes. Whilst this flexibility does allow for a marginally better predictive power, it comes at the sacrifice of the behavioural foundations and interpretability of the RUM model. This can introduce behaviours in the model which do not match reality; for example, if the parameter for driving cost is positive in the LR model, increasing the fuel costs in simulated trips would cause the predicted mode share of driving to increase. The manually specified utility functions in a RUM allow the modeller to check for, and if necessary avoid, these effects in these models. In theory, a RUM with optimal utility specification would outperform the LR classifier, particularly if higher order interactions of input variables were modelled. However, as discussed in Section 5.2.5.2, there is a practical limit on the complexity of the utility specifications in RUMs when using an open-ended manual search. A solution to this problem is explored in Section 7.4.

Whilst there are clear performance differences between all of the models, the performance differences between RUMs and ML models is smaller than has been suggested by previous research, e.g. the comparative study conducted by Hagenauer and Helbich (2017). In particular, the relative difference in DCA between any two models in this investigation is under 2.5 %, which is far smaller than that found in other research. This suggests that the results of previous studies may have been impacted by the technical limitations identified in Chapter 3.

7.3.1.2 Significance testing

As discussed in Section 5.2.6, 100 iterations of out-of-sample bootstrap validation are used to estimate the performance distributions for each classifier. The Cross-Entropy Loss (CEL) is used to analyse the distributions as both the arithmetic mean and absolute differences of/in the CEL are directly linked to the joint likelihood of the data (unlike the GMPCA,

for which geometric distributions would need to be estimated). It is trivial to convert between the GMPCA and CEL by taking the natural logarithm/exponential respectively (see Section 5.2.4).

The distributions of the negative CEL are shown in Fig. 7.2. The closer to zero the negative CEL (more to the right in Fig. 7.2), the better performing the classifier. As discussed in Section 7.3.1, bootstrapping results are not obtained for the SVM classifier, due to the computational complexity of fitting the model.

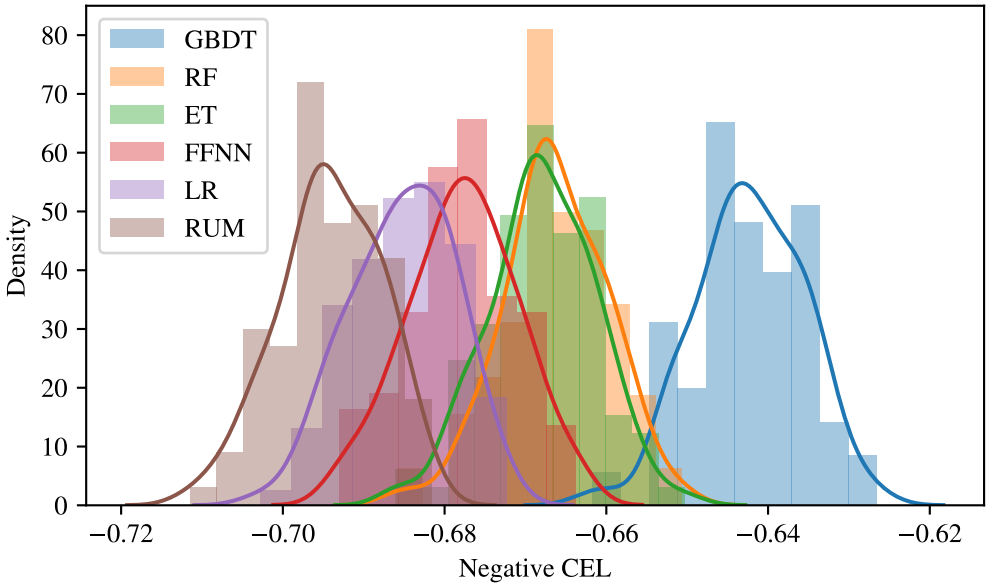


Fig. 7.2 Kernel density plots and histograms of out-of-sample CEL for 100 iterations of bootstrapping for each classifier.

Figure 7.2 clearly illustrates the performance groupings described in Section 7.3.1; with the GBDT classifier achieving the stand-out best performance, followed by the RF and ET models, then the FFNN, LR (and SVM, not shown in the graph), and finally the RUM.

As discussed in Section 5.2.6, the same bootstrap samples are used for each classifier, and so the t-statistic for the significance test is given by Eq. (5.22)

$$t = \frac{\bar{\delta}_i}{s/\sqrt{K}}.$$

where δ_i is the difference in CEL between the models on the ith bootstrap sample.

Figure 7.3 shows the Probability Density Function (PDF) of t-distribution alongside the histogram of the differences in CEL for the RF and ET classifiers (ET is reference model P^r,

and RF is the candidate model P^m). Note the PDF does not share the same y-axis as the histogram, and instead is normalised to be the same height.

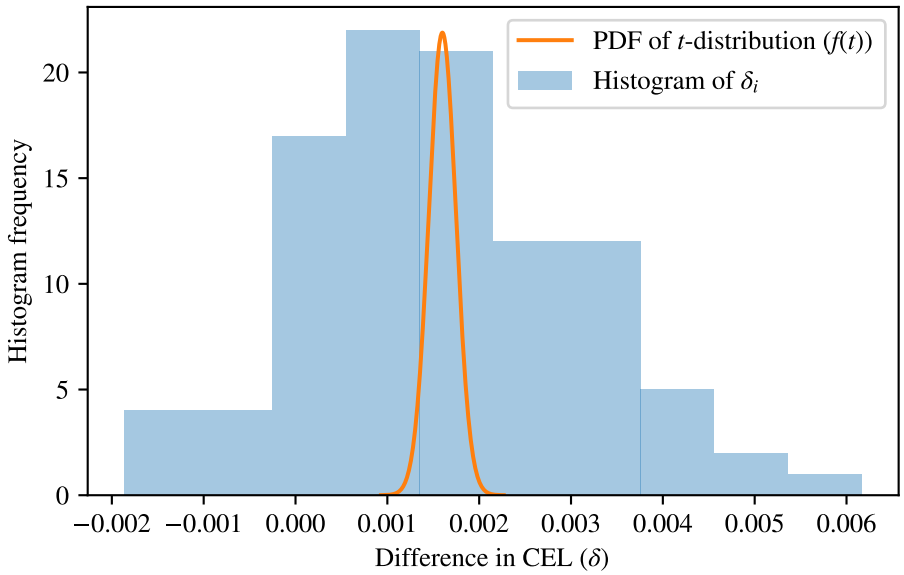


Fig. 7.3 Histogram and Central Limit Theorem (CLT) distribution of the mean of difference in CEL between RF and ET model for 100 paired bootstrapping iterations.

Whilst the two classifiers have similar, overlapping distributions of performance scores (as shown in Fig. 7.2), their relative performances on each sample are highly correlated: the RF model almost always achieves a marginally better performance estimate than the ET model across the bootstrap samples. The ET model achieves a better score than the RF model in only 11 of the 100 bootstrap iterations. As such, the differences between the scores have low variance, and so the significance test finds the mean difference in the performance estimates to be clearly different from zero ($t = 10.72$ corresponding to $p = 5.92e^{-18}$).

Table 7.4 shows the test statistic for differences between the bootstrap performance distributions for each model pair. All statistics indicate significance differences between the distributions of the bootstrap performance scores for the corresponding classifiers at any reasonable significance level (all $p \leq 5.92e^{-18}$).

As discussed in Section 5.2.6, the above significance test investigates only if the mean difference between the bootstrap scores is not equal to zero. It does not investigate if the differences in the CEL are significant in terms of the predicted probability distributions for a test sample. To investigate this question, the approach proposed in Eq. (5.24) is used, which investigates the power of the t-test using the significance of the marginal relative likelihood of an unknown test sample of known size n. Table 7.5 shows the test sample size at which

Table 7.4 Pairwise grid of test statistic for paired t-tests for significant differences in performance distributions between classifiers based on 100 iterations of bootstrapping CEL.

RF	130.7				
ET	120.5	10.72			
LR	123.1	33.65	30.24		
FFNN	191.0	71.49	62.16	22.50	
RUM	211.9	98.71	93.50	44.06	44.25
	GBDT	**RF**	**ET**	**LR**	**ANN**

classifier performance is expected to be significantly different for each pair of classifiers at the 5 % level (two-tailed), i.e. the test sample size n at which

$$\int_{-\infty}^{\infty} \frac{f\left(\frac{\bar{\delta}_i - \delta'}{s/\sqrt{K}}\right)}{(1+e^{n\delta'})} d\delta' = 0.025 \tag{7.2}$$

Table 7.5 Pairwise grid of test size at which classifier performance is expected to be significantly different at 5 % significance level.

RF	152				
ET	142	2326			
FFNN	103	322	374		
LR	85	194	212	491	
RUM	70	131	139	222	408
	GBDT	**RF**	**ET**	**LR**	**ANN**

The table shows that when predicting 70 or more trips, the GBDT model has significantly different performance from the RUM, whereas a test-set of 2326 trips is needed before the performance of the RF model is significantly different from that of the ET model.

7.3.1.3 Probabilistic simulation vs discrete classification

In order to be used for probabilistic simulation, a predictive model must be able to output well calibrated choice probabilities. As discussed in Section 5.5, unlike the other classifiers used in this comparison, RFs, ET, and SVMs do not inherently output choice probabilities found using Maximum Likelihood Estimation (MLE). In order to check whether the values outputted by each model represent well calibrated choice probabilities, Fig. 7.4 shows *reliability curves* for each classifier, which illustrate how closely the predicted probabilities for each mode match the empirical mode shares for those trips.

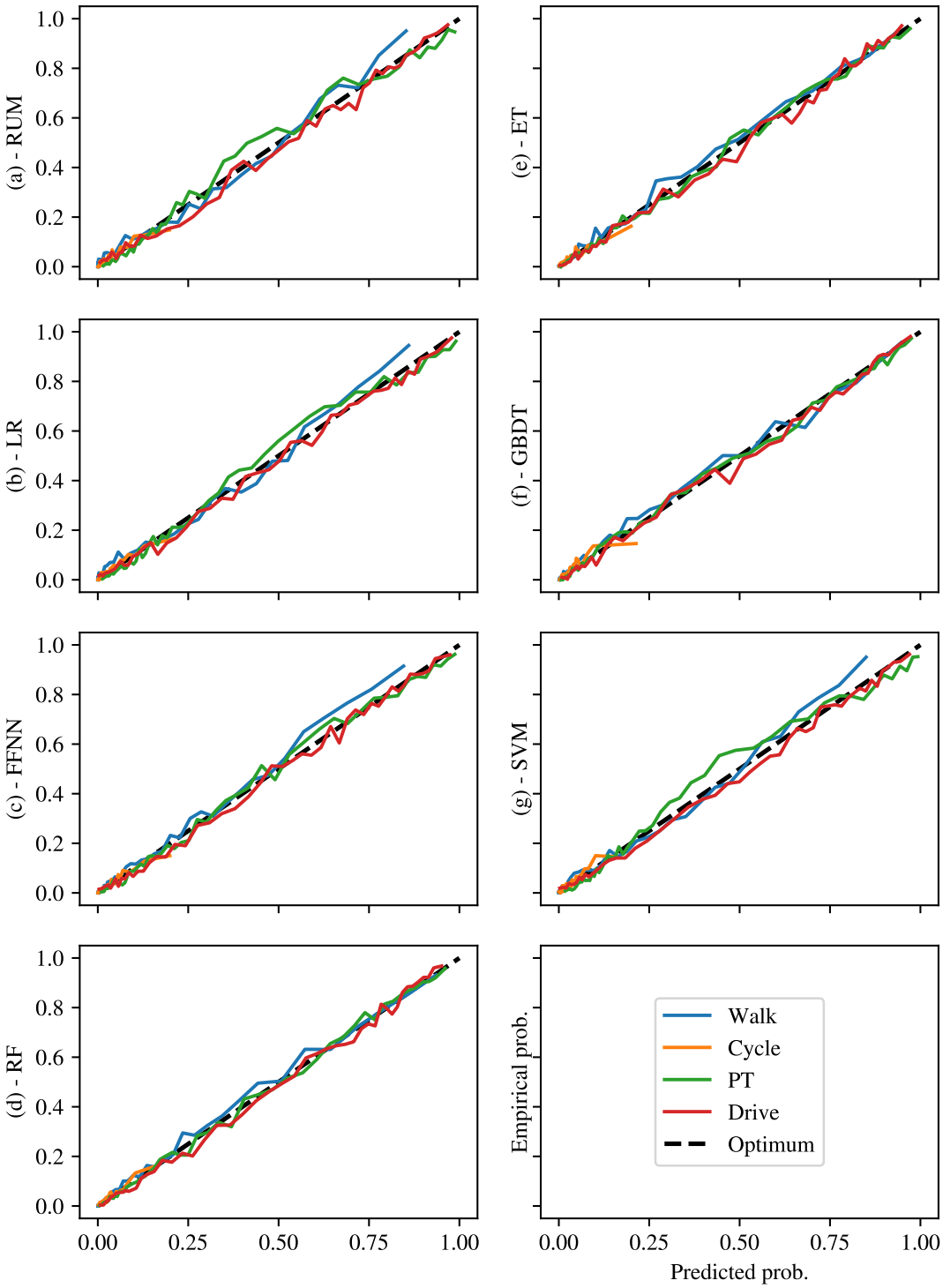


Fig. 7.4 Reliability curves of predicted probabilities against empirical probabilities per mode for each classifier using holdout validation data. 50 variable width bins are used, with an equal number of trips in each bin.

The predicted probabilities for each mode of each trip in the validation set are sorted into 50 variable width bins of equal number (i.e. each bin contains 2 % of the trips). For each bin, the empirical share of the selected mode is then calculated according to the recorded mode choice for each trip. The mean predicted probability for each bin for each mode is then plotted against the empirical probability. Classifiers with well calibrated probabilities will have values that sit on or close to the line $y = x$ (so that the predicted probability equals the empirical probability).

As can be seen from Fig. 7.4, all classifiers, including the RF model, ET model, and SVM, have reliability curves which sit close to the optimum, and as such can be considered to output well calibrated choice probabilities. As each classifier has been optimised for maximum GMPCA, hyper-parameter values are selected which result robust probability predictions (e.g. large ensembles for the EL classifiers).

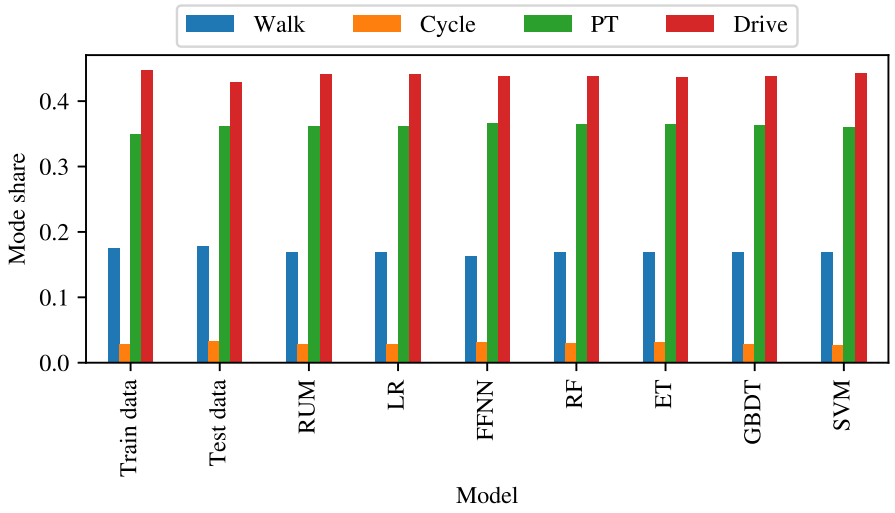
Figure 7.4 indicate that the three EL classifiers output predicted probabilities which closest match the empirical probabilities, compared to the other models.

Figure 7.5 shows the predicted mode shares for each classifier for probabilistic simulation and discrete classification, alongside the mode shares in both the train and test data. The probabilistic simulation mode shares for each classifier are calculated by taking the mean of the predicted probabilities for each mode ($1/N \sum_n^N P(i|x_n)$). The discrete classification mode shares are calculated by assigning each trip to the mode with highest predicted probability and calculating the ratios of the assignments. There is a slight difference in the mode shares between the train data and the test data, with a reduction in the proportion of driving trips, and an increase in the proportions of public transport and cycling trips.

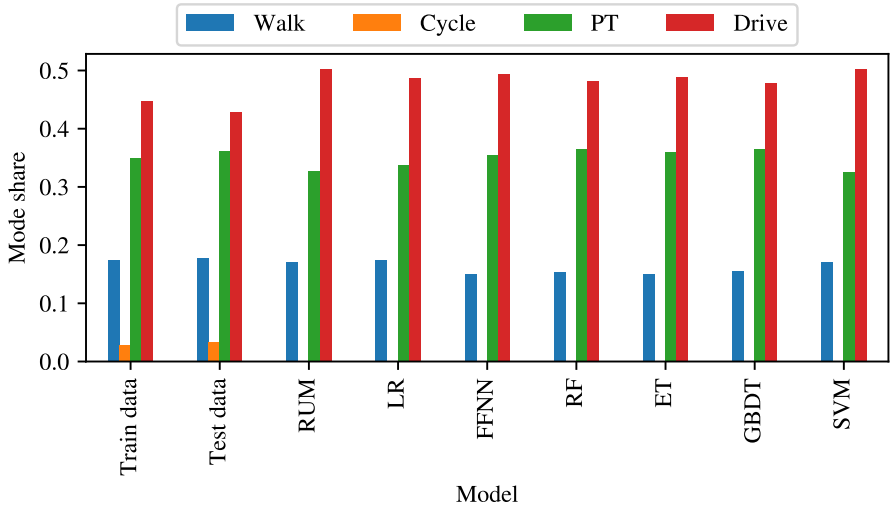
The differences between Fig. 7.5 highlights the issues with assigning predictions to the highest probability mode: the discrete mode shares for all classifiers over-represent the most common mode (driving), and severely under-represent the minority mode (cycling), providing poor estimates of the relative mode shares. As the mode shares are a key output for mode choice models for informing network operation and investment decisions, this makes discrete classification an unsuitable use case for transport modelling and simulation.

As shown in Fig. 7.5a, each classifier provides a much better estimate of the true mode shares when using probabilistic simulation (drawing mode choice randomly from the probability distributions over the modes).

Table 7.6 shows (for each classifier) the Mean Squared Error (MSE) of the differences between the number of trips of each mode in the test data and the number of trips predicted by the classifier for that mode, for both probabilistic simulation and discrete classification. The MSE for discrete classification is at least 17 times higher than that for probabilistic simulation across all classifiers.



(a) Probabilistic simulation.



(b) Discrete classification.

Fig. 7.5 Bar chart of predicted mode shares for each classifier for (a) probabilistic classification and (b) discrete classification, with train and test mode shares.

Table 7.6 MSE (in number of trips) of predicted mode shares for probabilistic simulation (PS) and discrete classification (DC) for each classifier.

	PS	DC
RUM	10308	329375
LR	10999	219536
FFNN	15024	268095
RF	8179	199467
ET	7013	238285
GBDT	8420	178443
SVM	14027	346970

The GBDT achieves the lowest MSE for the discrete classification mode shares, but both the ET and RF models achieve lower MSE for probabilistic classification. The FFNN achieves the highest MSE for probabilistic simulation, and the SVM for discrete classification.

7.3.1.4 Validation schemes

Table 7.7 shows the test (2014/15 holdout-sample), mean cross-validation (10-fold, grouped by household), and train (in-sample) GMPCA for each classifier in the comparative study. The cross-validation GMPCA is calculated from the arithmetic mean CEL, and so represents the geometric mean of the individual GMPCA scores. Both the train and cross-validation performance estimates are optimistically biased estimates (over-estimates) of the true test performance.

Table 7.7 GMPCA scores for each classifier for (i) holdout test, (ii) 10-fold Cross-Validation (CV), and (iii) in-sample train validation.

	Test	CV	Train
RUM	0.4960	0.5031	0.5070
LR	0.5000	0.5072	0.5098
FFNN	0.5025	0.5071	0.5148
RF	0.5082	0.5188	0.6118
ET	0.5067	0.5180	0.6641
GBDT	0.5215	0.5303	0.5794
SVM	0.5006	0.5070	0.5096

The train GMPCA is calculated on the same data the models are fitted to. As such, it continues to increase as a model overfits to the training data. This makes in-sample validation an unsuitable scheme for performance prediction. This is particularly true for the highly-

flexible EL models, which can very easily overfit to the data. The RF model train GMPCA is 27.5 % higher than the test GMPCA. This is also the true of the other metrics (not shown in Table 7.7), e.g. the RF train DCA is 84.8 %, significantly higher than its test DCA of 74.1 %. Due to these differences in overfitting between models, in-sample performance rankings are not a reliable indicator of true out-of-sample performance rankings.

Unlike the in-sample training set validation, the CV performance estimates are calculated on out-of-sample data. However, the CV GMPCA is consistently optimistically biased compared to the test GMPCA. This is despite the models in the CV estimate only being fit to 90 % of the data, thus having less opportunity to fit to relationships between input features and mode choice.

There are two sources of positive bias in the CV performance estimate. Firstly, as discussed in Section 5.3, hyper-parameter optimisation is performed using 10-fold CV, with the parameters which result in the lowest CEL (and therefore highest GMPCA) selected. As a result, this allows for *data leakage* from the class labels from the out-of-sample trips, as the model hyper-parameters are effectively *fit* to the validation folds. Using the same data for performance validation as model optimisation therefore introduces positive bias in the estimate. This demonstrates why the model must always be tested on data not used for model training or optimisation.

Secondly, whilst the CV folds are sampled grouped by household, the households are all sampled randomly from the same data. As such, the CV does not validate the model on *external* data, collected separately from the training data. This can result in a positive performance estimate bias, e.g. when compared to using a model to predict a future year of data.

7.3.1.5 Model optimisation

The hyper-parameters for each ML classifier are selected using 20 iterations of random search followed by 80 iterations of Sequential Model-Based Optimisation (SMBO) using a Tree-structure Parzen Estimator (TPE) algorithm. The selected hyper-parameter values for each classifier are given in Appendix A.2. Figure 7.6 shows the k-fold CV CEL for each iteration of model optimisation for each classifier, alongside the cumulative minimum CEL. For all classifiers, the majority of the improvement in CEL occurs within the first 10 iterations of random search, though all show further improvements during the SMBO iterations (see Table 7.8).

There is large variation in the CEL between different hyper-parameter trials for all classifiers except for the GBDT model. The GBDT model has less variation as the *early stopping rounds* variable is used to set the most import hyper-parameter (number of boosting

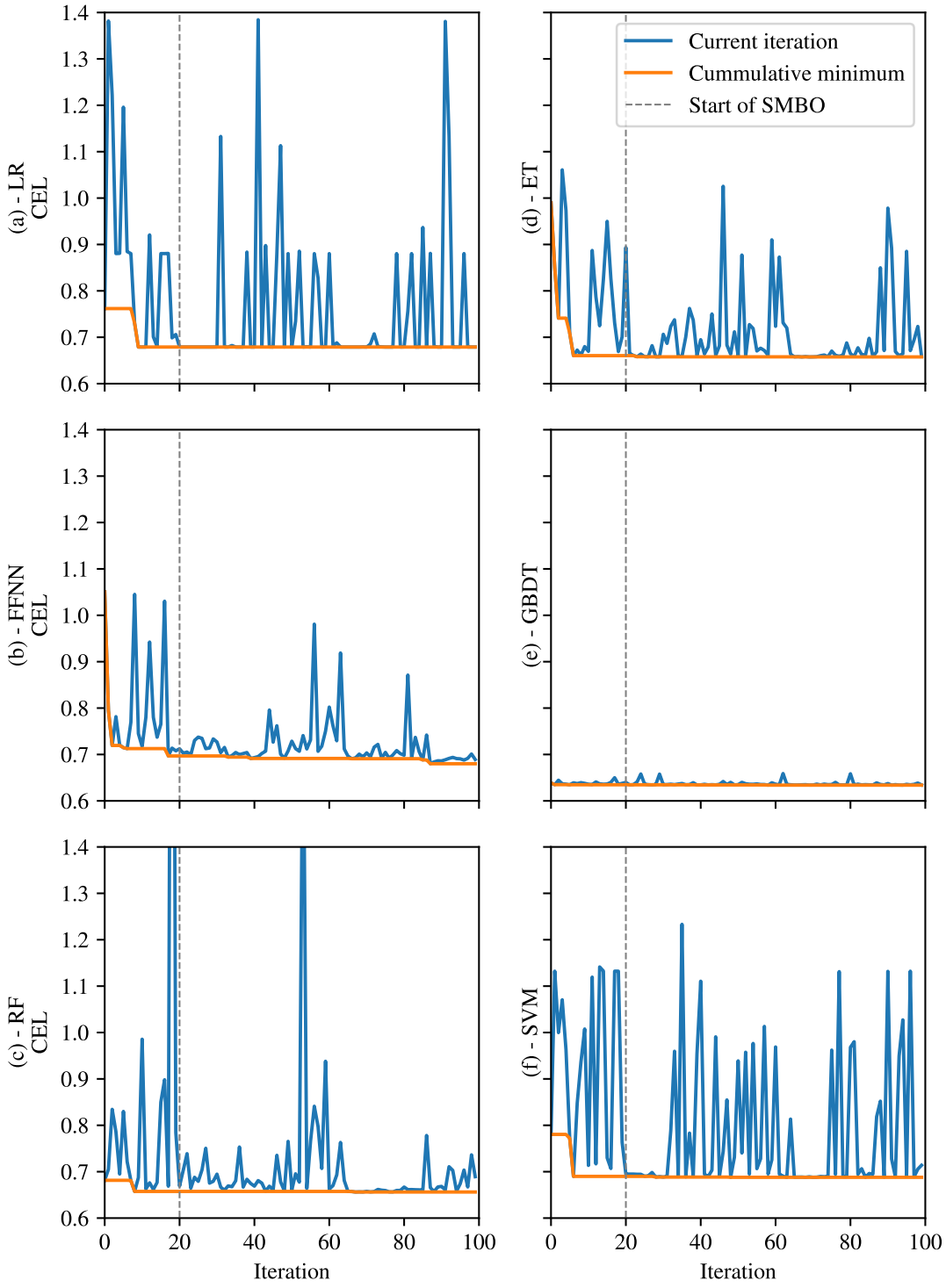


Fig. 7.6 Graphs of current iteration CEL and cumulative minimum CEL for each ML classifier over 100 iterations of SMBO.

rounds) dynamically, based on the improvement in performance in successive boosting rounds. The number of stopping rounds is therefore always optimised to the other hyper-parameter values, causing less variability in the resulting scores.

The variability in performance of the algorithms for different hyper-parameters highlights the importance of selecting appropriate hyper-parameter values. Without performing appropriate model optimisation, any differences in relative performance between two models may be due to the use of inappropriate hyper-parameter values. This is further demonstrated in Table 7.8, which shows the initial, worst case, and best case (optimised) CV GMPCA. The rankings for the initial and worst-case scores do not match those for the optimised scores. There are substantial differences between both the best case and worst case GMPCA scores for all classifiers. In particular, the GMPCA for the optimised FFNN model is 1.45 times higher than that for the initial conditions, and the GMPCA for the worst-case RF hyper-parameters is over 11 times smaller than the optimised value.

Table 7.8 Initial, highest, and lowest GMPCA for each classifier over 100 iterations of hyper-parameter optimisation.

	Initial score	**Lowest score**	**Optimised score**	**Best iteration**
LR	0.4668	0.2505	0.5072	69
FFNN	0.3498	0.3498	0.5066	88
RF	0.5059	0.0468	0.5188	68
ET	0.3717	0.3461	0.5181	70
GBDT	0.5281	0.5174	0.5305	59
SVM	0.4582	0.2913	0.5026	49

7.3.1.6 Model suitability

The previous sections of this comparative study focus purely on the out-of-sample predictive power of the different classifiers being compared. As discussed in Section 4.2.6 model selection is also dependent on other aspects, such the perceived interpretability, reliability, robustness, and complexity of the model, as well as model fitting times. This section discusses these aspects of the classifier comparison.

Compared to the ML classifiers, RUMs have two primary advantages: *interpretability* and *robustness*. The parameter values for the RUM (presented in Table B.4) present a complete explanation of the scale and direction (positive/negative) of influence of every input variable on the utility for each relevant class. Whilst increasing model complexity by adding more parameters can make it harder to follow (when considering all features simultaneously), it is

always possible to understand the impact changing each feature individually has on the class utilities, no matter how complex the model. This results in a high level of *interpretability*.

Additionally, by conducting significance tests on each parameter, and checking the signs and magnitudes for each one, it is possible to check that the utility specifications are consistent with established behavioural theory. The extra constraints on the RUM model, compared to the LR model (and the other classifiers), ensures that the utility for each class is only calculated from attributes describing that class (e.g. public transport fares only affect public transport utility), and that possible overfitting/endogeneity does not result in unexpected model behaviours (e.g. increasing public transport fairs increasing the utility of public transport). This ensures that the RUMs describe *robust* behavioural models which do not contradict established theory.

These advantages are unmatched by the ML classifiers. Both the LR model and the SVM (when used with a linear kernel) also include linear weights for each variable (via the decision hyper-plane in the SVM). However, in both models, these weights are abstracted in several ways. Firstly, in both models a parameter is specified for every variable for every class (e.g. increased public transport costs affect the utility/choices for walking, cycling, and driving, as well as public transport). Secondly, the SVM uses a *one-vs-one* classification scheme as well as a separate probability calibration model. This makes the relationship between input variables and choice utilities/probabilities non-linear. For both models, the lack of any statistical tests or requirements for signs and scale of the parameters mean that the models do not represent a robust behavioural model and may include insignificant parameters and/or parameters with incorrect sign, which can result in unexpected behaviours in simulation.

By including one or more hidden layers in the FFNN, and using non-linear activation functions, the network is able to model complex, non-linear relationships between input features and output class predictions. This means it is not possible to easily summarise the impact of each feature on the predictions, which can vary dependent on the other feature values. Additionally, the non-linear nature makes it difficult to check for consistency with established behavioural theory.

Individual DTs are highly interpretable, as the prediction is obtained by simply following the binary splits down the tree. However, combining the effect several DTs in an EL algorithm obfuscates the prediction process, and reduces the model's interpretability (in exchange for greater predictive power). Regardless, aggregate analysis of the binary splits in the DTs in an EL classifier can provide useful information about the prediction process.

A measure of feature importance in an EL model can be easily obtained by summing up the information contribution of each feature over all splits in all trees (the feature importances for the GBDT model are presented in Section 7.3.3.2). Additionally, by considering

sequential binary splits, it is possible to investigate the importance of different feature interactions in the model. However, feature and feature interaction importances present only an aggregate measure of importance (considering all classes simultaneously), and do not give any indication of the direction of impact (i.e. making a given class more or less likely).

As well as the feature importances, it is possible to analyse the split values for each feature to extract useful information on non-linear relationships of the features. This is exploited in Section 7.4, where the fitted GBDT model is used to inform the utility specifications in a RUM. As with the FFNN, the highly non-linear nature of the binary splits makes it hard to check for behavioural consistency.

Overall, considering the classifiers compared in this study, there is a trade-off between the higher predictive power and flexibility of the ML algorithms and the behavioural robustness and interpretability of the RUMs.

Fit and predict times Table 7.9 shows the fit and predict times in seconds for the classifiers averaged over the CV folds. The CV is performed on the train data only, so on average the models are fit to 49 289 trips and used to predict 5477 trips. All models are trained using the same Central Processing Unit (CPU) on the same PC (no processing is performed on a Graphics Processing Unit (GPU)). Eight threads of parallelisation are used for fitting all classifiers except the SVM, for which the *LIBSVM* implementation within *scikit-learn* does not support parallelisation. The BIOGEME library only reports durations to the nearest second, and so the predict time for the RUM is known to be under 0.5 s.

Table 7.9 Mean fit and predict times (in seconds) for each classifier over 10 folds of CV. *SVM uses no parallelisation for fitting, all other classifiers use eight threads.

	Fit time	Predict time
RUM	196.50	<0.5
LR	28.71	0.02
FFNN	1.75	0.24
RF	102.60	2.49
ET	95.92	3.87
GBDT	72.22	2.84
SVM*	2.64×10^5	25.25

The SVM is much more computationally intensive to fit than the other classifiers, taking several orders of magnitude longer to fit the model (>72 hours). As discussed in Section 2.3.5, SVM efficiency scales at a minimum with $O(n^2)$ (where n is the number of rows in the data), rising to $O(n^3)$ for high C (regularisation) values. The C value during model optimisation is

limited to 500 to allow for completion of the 100 iterations of SMBO on a 10 % sample to complete in reasonable time. However, the selected C value of 227.5 (see Table A.11) results in very long fit times for the larger dataset. This is exacerbated by the use of five-fold CV for probability calibration, and six binary classifiers for the *one-vs-one* multiclass classification scheme. In practice, this rules out SVM from being used for probabilistic prediction for data of this depth and scale. The SVM also takes the longest to predict the data (over eight times longer than the next slowest model), and so would also be slower to use for simulation once the model is fit.

Of the remaining classifiers, the FFNN has the fastest fit time. This is despite the fact it has many more parameters to fit than the LR and RUM models (5076 for the FFNN, 176 for LR, and 55 for the NL model). This shows the efficiency of the mini-batch gradient descent algorithm used in *tensorflow*. The rest of the classifiers have fit times around the same order of magnitude, though the predict time is longer for the EL models. This is because the EL models have to compute the output for all 1400+ trees in the ensemble to generate each prediction.

7.3.2 Sampling methods for hierarchical data

This section investigates the use of household-wise (grouped) sampling vs trip-wise sampling for mode choice prediction. The six ML classifiers compared in Section 7.3.1 are each optimised again using the same modelling framework, except that trip-wise sampling is used for the CV folds. All other steps/hyper-parameter search spaces are kept identical. The selected hyper-parameter values for the trip-wise sampling models are given in Appendix A.2.2. The trip-wise and household-wise sampled models are compared using k-fold CV, holdout validation, and bootstrapping significance tests.

Table 7.10 shows the CV (folds sampled trip-wise), and holdout validation results for the models optimised using trip-wise sampling. The table shows that the trip-wise CV significantly over-predicts true out-of-sample test performance for models fitted to hierarchical data using trip-wise data, in particular for the EL algorithms and the SVM. This is due to the data leakage from pairs/sets of correlated trips with matching modes in the hierarchical data (see Section 6.4.2). This data leakage allows the models to overfit to particular features which are constant across these correlated trips, such as the exact straight-line distance. The model can then simply repeat the mode observed in the training data with high confidence for the corresponding trips in the test data, therefore overestimating the performance on true out-of-sample data.

As with using in-sample validation to estimate true performance (see Section 7.3.1.4), the amount of positive bias for trip-wise CV is dependent on the model being tested. As such, the

Table 7.10 Trip-wise sampling optimised models - *k*-fold CV (folds sampled trip-wise) and holdout validation (train on 2012/13-13/14, test on 2014/14) GMPCA.

	CV	Test	Relative difference
LR	0.509	0.500	1.017
FFNN	0.507	0.498	1.017
RF	0.580	0.507	1.143
ET	0.585	0.504	1.161
GBDT	0.627	0.482	1.301
SVM	0.560	0.439	1.276

performance rankings using trip-wise CV do not match those using external validation, and relative trip-wise CV performance is not a reliable indicator of true expected performance.

Household-wise sampling does not show this behaviour. The relative difference in performance estimates for household-wise sampling (shown in Table 7.11) are smaller for all models than those for trip-wise sampling. In addition, the relative differences are much more consistent between models, ranging from 0.9 % to 2.2 % (compared to 1.7 f to 30.1 for trip-wise sampling). This shows household-wise sampling to be a more appropriate method for performance estimation.

Table 7.11 Household-wise sampling optimised models - *k*-fold CV (folds sampled household-wise) and holdout validation (train on 2012/13-13/14, test on 2014/14) GMPCA.

	CV	Test	Relative difference
LR	0.507	0.500	1.014
FFNN	0.507	0.503	1.009
RF	0.519	0.508	1.021
ET	0.518	0.507	1.022
GBDT	0.530	0.521	1.017
SVM	0.507	0.501	1.013

As well as significantly overestimating model performance, optimising the models using trip-wise sampling also causes hyper-parameters to be selected which favour overfitting to the data. This is shown by inspecting the optimal hyper-parameters for the trip-wise EL, FFNN and SVM models. The RF model optimised using trip-wise sampling (Table A.14) uses a larger maximum tree depth and significantly smaller minimum leaf size than the household-wise sampling model (Table A.8). This allows each tree in the ensemble to overfit more to the data. Similarly, the trip-wise ET model (Table A.15) uses a larger max depth, and

does not subsample features or bootstrap the observations, causing there to be less variability in the ensemble (and therefore more potential for overfitting). The GBDT model additionally uses a much larger maximum depth, smaller minimum leaf size, and more boosting rounds for the trip-wise optimised model. The trip-wise FFNN model has a much more complex structure, using three hidden layers (rather than one in the household-wise model). These layers add flexibility to the model, allowing it to overfit more easily. Finally, the trip-wise SVM makes use of a non-linear Radial Basis Function (RBF) kernel (as opposed to the linear kernel used in the household-wise SVM), which has higher potential to overfit to the data.

The differences in model hyper-parameters result in models which do not perform as well for true out-of-sample prediction. Figure 7.7 shows the bootstrapping distributions for the household-wise and trip-wise model for each classifier (except SVM for which bootstrapping results are not obtained). The household-wise optimised classifiers all achieve higher average performance than the respective trip-wise optimised models. For all except the LR models, the differences in the distributions are shown to be significant at the 5 % level using a two-sample t-test.

7.3.3 Mode-alternative attributes

This section uses the GBDT algorithm to investigate the impacts of the data generation framework presented in Chapter 6, which adds mode-alternative attributes to the dataset. This investigation is performed by comparing the original *choice-set model* to the *raw-data model* - a GBDT model optimised and fitted using only the socio-economic and demographic profile data from the LTDS (i.e. the subset of the features in the full dataset which are not generated through the data fusion methodology). The raw-data model is optimised using the ML modelling framework, with the same hyper-parameter search space as the original choice-set GBDT model.

Firstly, Section 7.3.3.1 investigates the performance differences between the two models. Section 7.3.3.2 then presents the differences in the relative importances of each feature in the model.

7.3.3.1 Performance differences

Table 7.12 shows the holdout validation results of the optimised raw-data and choice-set models. In all three metrics, the choice-set model outperforms the raw-data model. This signifies that the data fusion process has added significant features to the dataset.

Notably, the performance achieved by the raw-data model is lower than that of any of the ML classifiers trained on the full dataset, for all metrics (see Table 7.3). This demonstrates

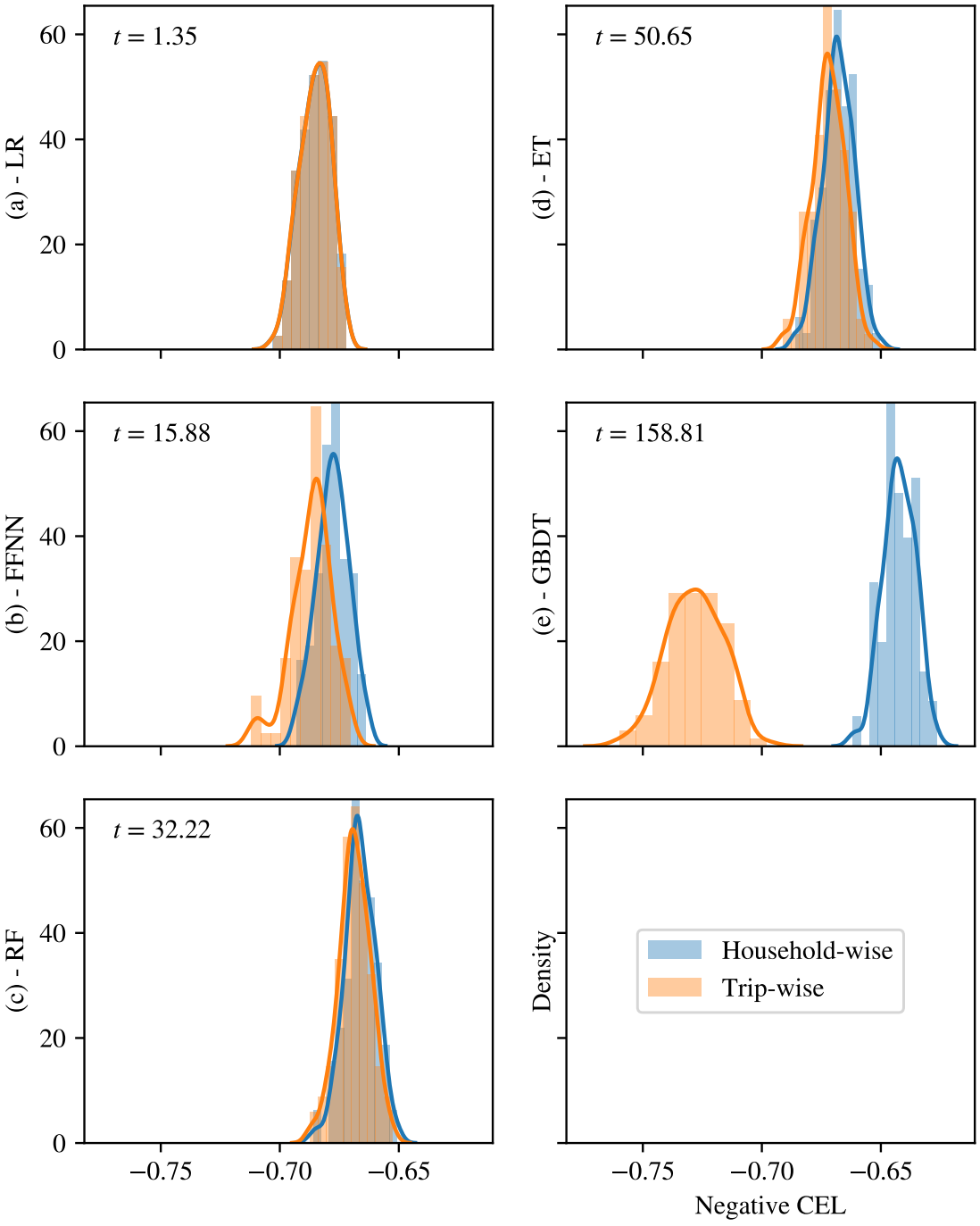


Fig. 7.7 Pairwise kernel density plots and histograms of out-of-sample CEL for 100 iterations of bootstrapping for household-wise and trip-wise optimised models for each classifier.

Table 7.12 Holdout validation results for raw-data and choice-set GBDT models.

	GMPCA	**AMPCA**	**DCA**
Raw-data model	0.4881	0.6078	0.7154
Choice-set model	0.5216	0.6509	0.7485

that adding mode-alternative attributes to the dataset plays a more significant role in the overall performance of the resulting ML classifiers than the choice of model algorithm.

Figure 7.8 shows violin distribution plots of the predicted mode choice probabilities for the selected mode within the holdout-test dataset for each model, i.e. the probability predicted by the model for the mode actually taken by the passenger. The choice-set model distributions are skewed more towards higher probabilities than the raw-data model distributions, both for all trips combined and for each individual transport mode. This shows that the choice-set model tends to predict the correct mode with higher probability than the raw-data model.

Figure 7.8 shows that both models predict highest choice-probabilities for the selected mode for driving and public transport trips, followed by walking trips and finally cycling. This is a result of the mode shares of trips in the dataset, presented in Section 6.4.

Fig. 7.9 shows the histograms of the 100 iterations of out-of-sample bootstrapping results for each model. There is no overlap in the two distributions, with the worst choice-set model score exceeding the best raw-data model score. This results in a high t-statistic (231.2), meaning that, using the power-based approach presented in Section 5.2.6, the performance difference between the models is significant for a test sample of 59 trips or greater at the 5 % significance level.

7.3.3.2 Feature importances

As discussed in Section 2.3.3, each split in a DT is calculated based on the reduction in entropy in the data (i.e. information *gain*). It is therefore possible to calculate the information contribution of each feature in a tree. This can be summed over all trees in an ensemble to calculate the relative *importance* of each feature in a GBDT model.

Figure 7.10 shows the ranked relative feature importances of each feature in the choice-set and raw-data models. The relative importances are calculated from the total ensemble gain of each feature, which is equal to the average gain contributed by a feature at each split across all trees in the ensemble, times the weight (the frequency that feature is used in the ensemble). Classes of features are grouped by category to form compound features, as shown in Figs. 7.10c and 7.10d. This includes the categorical features which are one-hot encoded (`faretype`, `fueltype`, `purpose`), and cyclical data which has the sine and

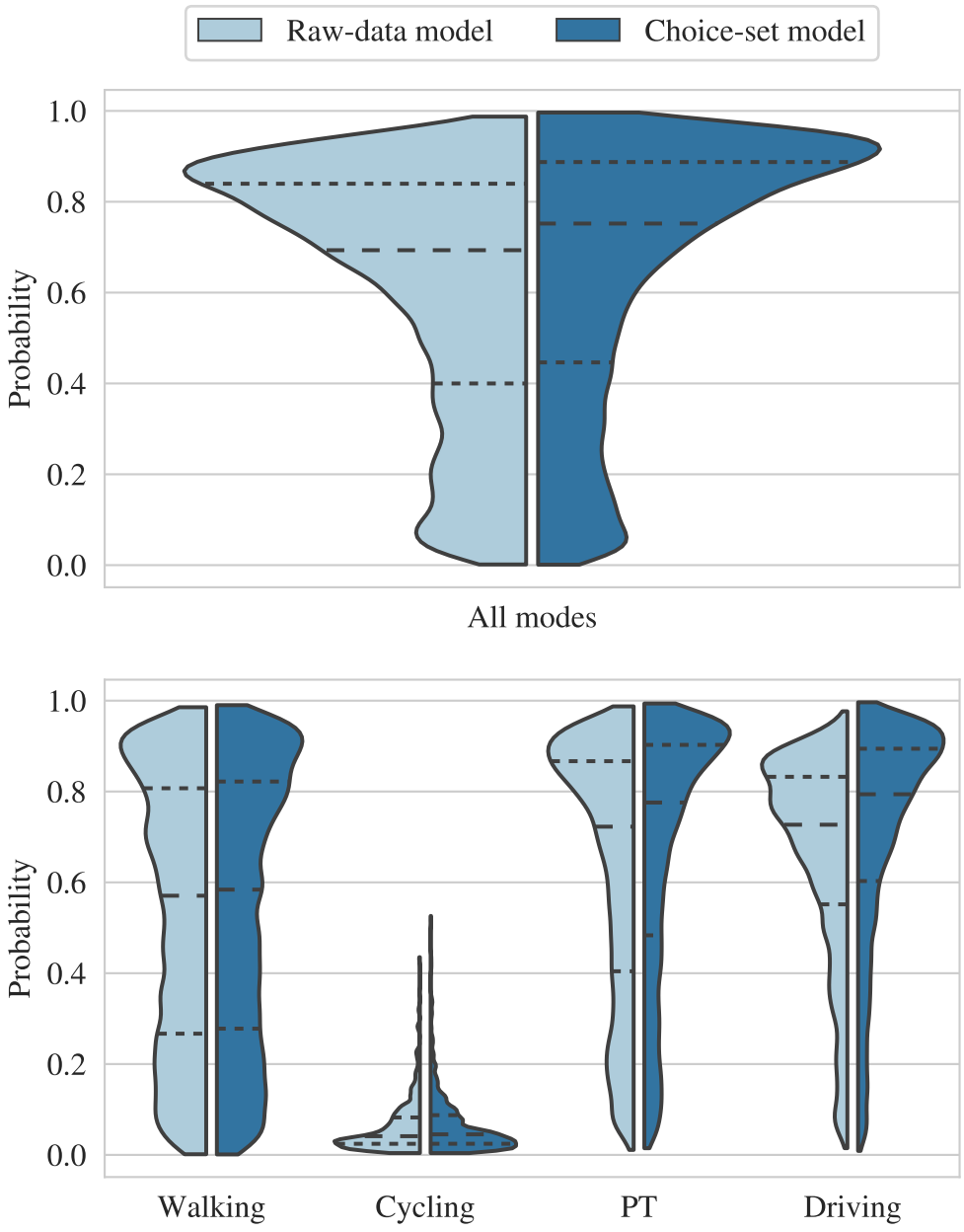


Fig. 7.8 Violin frequency plots of predicted mode choice probabilities for raw-data and choice-set models for all modes combined and grouped by selected transport mode. Dashed lines mark median and interquartile ranges of each distribution.

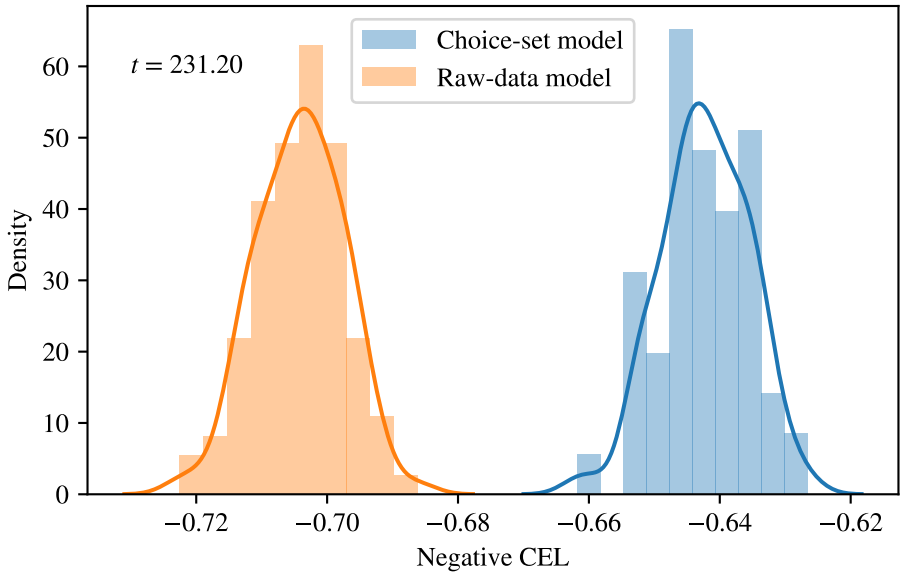


Fig. 7.9 Kernel density plots and histograms of out-of-sample bootstrap CEL for raw-data and choice-set models.

cosine added (`start_time`, `day_of_week`, `travel_month`) for both models, as well as the different subcategories of the public transport duration and driving costs (`dur_pt` and `cost_driving`) in the choice-set model.

In general, the choice-set model has more balanced feature importances than the raw-data model. In the raw-data model, the two features with highest importance (`distance` and `vehicle_ownership`) account for 67 % of the total information in the model. In contrast, the two features with highest importance in the choice-set model (`vehicle_ownership` and `dur_walking`) only account for 29 % of the total information gain. Figure 7.10a shows that the duration features added to the choice-set model from the directions API have high relative importance, as does `traffic_percent`. Within the public transport duration, the on-board bus, on-board rail, and access durations are all of similar importance.

In both models, `sex` and `travel_month` are of low relative importance, suggesting that there are not strong month-by-month or gender variations in the mode-distributions.

7.4 Assisted specification approach

As discussed in Section 7.3.1.6, RUMs have two primary advantages over ML methods: (1) the interpretability of the model parameters, and (2) the robustness of the underlying behavioural model. However, as shown by Section 7.3.1, the flexible ML classifiers, and

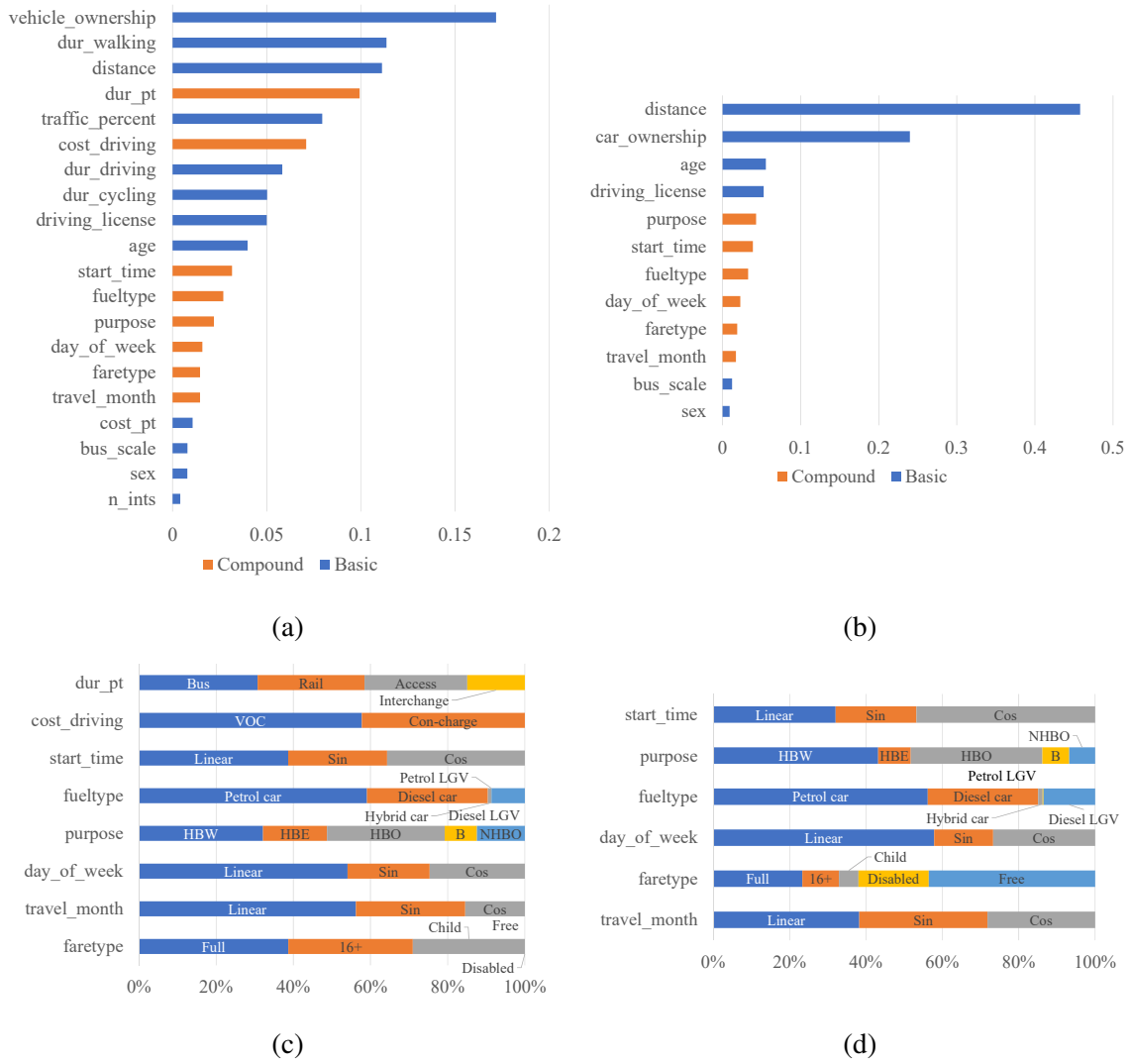


Fig. 7.10 Relative feature importance (ensemble gain) with compound features for (a) choice-set model and (b) raw-data model. Sub-feature labels and proportions for compound features are given in (c) for the choice-set model and (d) for the raw-data model.

in particular the GBDT model, have a significant predictive performance advantage over the RUMs. The lower performance of the RUMs is partly due to the limited complexity of the utility specifications which can be investigated manually in an open-ended search. This limits the RUMs tested in Section 7.2 to first order interactions of the covariates from the LTDS profile directly with the choice utilities.

This section presents the results of the assisted specification approach, which addresses the performance limitations of RUMs by using the feature importances and split points from the GBDT ensemble to inform the utility specification structure in a RUM.

Firstly, Section 7.4.1 investigates the most important attributes in the GBDT model (vehicle ownership, distance, driving licence, and age), in order to identify four successive modifications to the combined MNL presented in Table B.3. This includes a non-linear function of a continuous input variable (*log-distance model*), heuristic binning of a continuous variable (*heuristic age model*), second order feature interactions with the choice probabilities (*vehicle ownership model*), and third order feature interactions (*full assisted specification model*).

Next, Section 7.4.2 presents the results for the four models, and discusses the fit process and parameter values. Finally, Section 7.4.2.1 compares the *full assisted specification* model with the ML classifiers, using holdout validation and bootstrapping results.

The investigations in this section are intended to illustrate some ways in which ML classifiers can be used to inform the utility specifications of a RUM.

7.4.1 Model specifications

From the single feature importances (plotted in Fig. 7.10a), the most important covariates are vehicle ownership, distance, driving licence, and age. Of these, two are categorical: vehicle ownership (no vehicles, less than one vehicle per adult, one or more vehicles per adult) and driving licence (yes, no); and two are continuous: distance and age. The categorical variables are already fully interacted with the ASCs, (using dummy variables). However, distance is included as a trip variable with its own parameter, and age is interacted with the ASCs as a binned variable, using the a-priori bins of child (<18), adult (18-64), and pensioner (65+).

Figure 7.11 shows the distribution of the binary split values for the straight-line trip distance, across all trees in the GBDT. The distribution is heavily skewed towards shorter trips, with a long tail towards longer trips. This shape is characteristic of a log-normal distribution and suggests trip choice probabilities are related to log-distance. This implies that log-distance should be included as an input in the utility specification for a RUM.

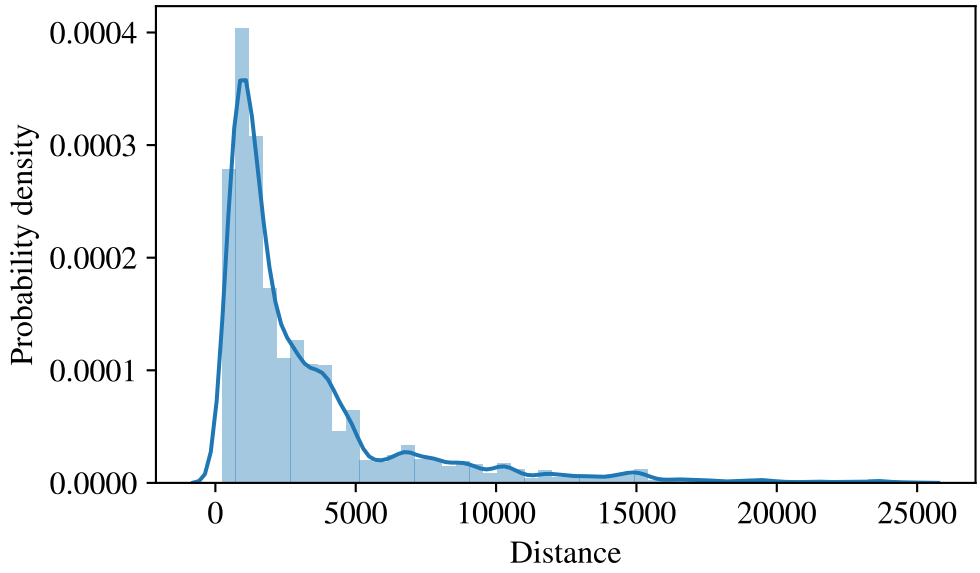


Fig. 7.11 Histogram and Kernel Density Estimation (KDE) plot of split values for straight-line trip distance across all trees in GBDT classifier.

Figure 7.12 shows the corresponding plot for the natural logarithm of the trip distance. The distribution is approximately symmetrical, reinforcing the suggestion that there is a relationship between the log straight-line trip distance and mode choice.

Based on the split distributions in Figs. 7.11 and 7.12, the log-distance is added (alongside the distance) to the utility specifications for each mode for the combined MNL presented in Table B.3. As new parameters are added to the model, the model is once again simplified from the full (complex) specification. This model is referred to as the *log-distance* model in Table 7.13.

Figure 7.13 shows a bar chart of the number of splits at each age value across all trees in the GBDT model. Unlike the distribution for the distance splits, there is not a strong skew in the data. Instead, there are three modal peaks, at 11.5, 31.5, and 66.5 years (age is given as a whole integer number of years, so the split points all occur at half years). These modal split values are used to define four new bins to be interacted with the ASCs in a RUM: (i) child (<12), (ii) young adult (12-31), (iii) mature adult (32-66), and (iv) pensioner (67+). These heuristic bins are used in a new model, in place of the a-priori bins of child (<18), adult (18-64), and pensioner (65+) from the *log-distance* model specification. Again, as new parameters are added, the model is simplified from the full complex model. This is referred to as the *heuristic age* model in Table 7.13.

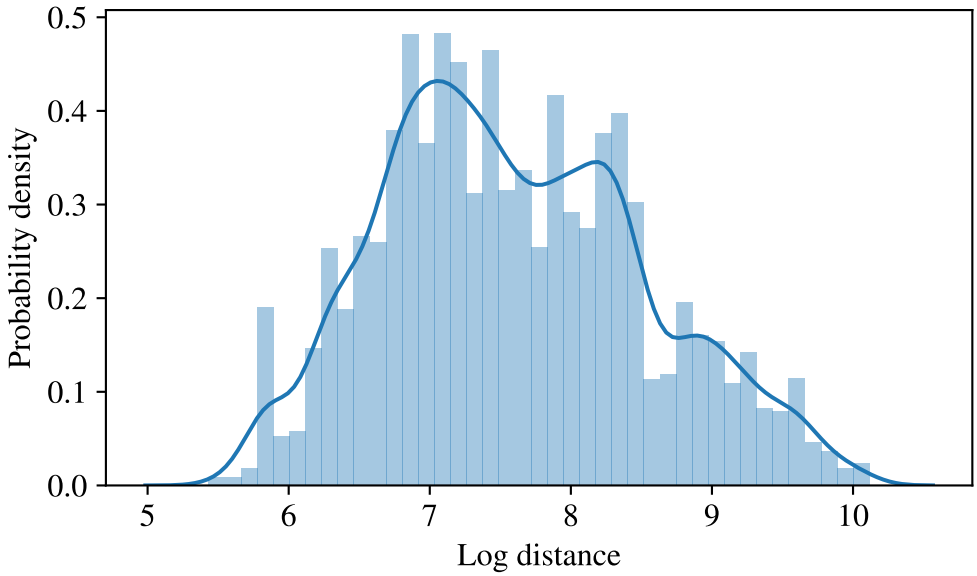


Fig. 7.12 Histogram and KDE plot of split values for natural logarithm of straight-line trip distance across all trees in GBDT classifier.

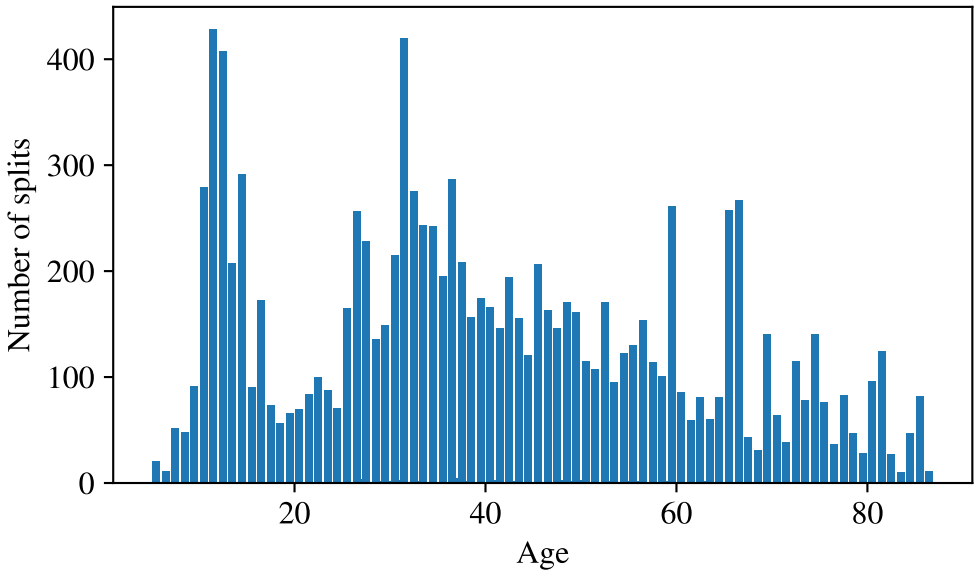


Fig. 7.13 Bar chart of number of splits at each age value across all trees in GBDT classifier.

Of the 10 most important second order feature interactions (i.e. two variables jointly interacting with the choice probabilities), six include vehicle ownership (alongside traffic variability, walking duration, congestion charge, driving duration, straight line distance, and driving licence ownership). This implies that vehicle ownership should be interacted with the other variables in the utility specifications in the RUM.

In order to achieve this, the vehicle ownership covariate is full interacted with all other parameters in the *heuristic age* model, replacing each parameter with three new parameters before simplification (one for each vehicle ownership group). This is referred to as the *vehicle ownership* model Table 7.13.

Finally, the most important third order interaction (i.e. three variables jointly interacting with the choice probabilities) which contains at least two socio-economic covariates is vehicle ownership/driving licence/traffic variability. Vehicle ownership/driving licence is also the most important second order feature interaction between socio-economic covariates. As such, for the final model tested, the driving licence variable is fully interacted with the parameters from the *vehicle ownership* model, so that there each parameter is replaced with two parameters before simplification (one for each of owning and not owning a driving licence). This is referred to as the *full assisted specification* model in Table 7.13.

7.4.2 Assisted specification results

Table 7.13 shows the fit, train, and test results for the four assisted model specifications, alongside the original MNL using manual specification (shown in Table B.3). All the modifications presented in Section 7.4.1 improve the out-of-sample predictive performance (both GMPCA and DCA). This demonstrates that the assisted specification approach, where the structure of the GBDT is used to inform the utility specifications in a RUM, can be used to improve model fit, and that the four different types of modification tested (non-linear function of input feature, heuristic binning of input covariate, second order feature interactions, third order feature interactions) are all valid ways of improving the RUM performance.

Table 7.13 Results for assisted specification approach RUMs.

			Fit		Train		Test	
Model	**Params**	**Fit time**	**LL**	**AIC**	**GMPCA**	**DCA**	**GMPCA**	**DCA**
Original manual specification	54	01:40	-37281.77	74671.53	0.5062	0.7390	0.4954	0.7297
Log-distance	58	02:15	-36513.90	73143.80	0.5134	0.7414	0.5022	0.7334
Heuristic age	60	04:10	-35976.02	72072.04	0.5185	0.7486	0.5073	0.7416
Vehicle ownership	87	13:01	-35403.55	70981.10	0.5239	0.7524	0.5107	0.7428
Full assisted specification	100	42:32	-35082.62	70365.24	0.5270	0.7546	0.5116	0.7434

A total of 165 parameters are tested in the *vehicle ownership* model (three new vehicle ownership parameters for each of the 55 non-vehicle ownership parameters in the *heuristic age* model). Just under half of these parameters are removed during simplification, either through combining related parameters which are not significantly different from each other or removing parameters which are not significantly different from zero. This leaves 87 significant parameters.

Five of the parameters in the *vehicle ownership* model are driving license parameters. As such, there are 164 parameters tested for the *full assisted specification* model (two driving licence parameters for each of the 82 remaining parameters in the *vehicle ownership* model). After simplification, the *full assisted specification* model contains 100 parameters.

Table B.6 shows the parameter estimates for the *full assisted specification* model. The naming convention is the same as in Section 7.2.1, with the addition two optional suffixes. The suffixes for vehicle ownership are: NVO - no vehicle ownership, VO1 - less than one vehicle per adult, VO2 - one or more vehicles per adult, VO - any vehicle ownership (combination of VO1 and VO2), NVO1 - no vehicle or less than one vehicle per adult (combination of VO and VO1); and for driving licence are: DL0 - no driving licence, and DL1 - driving licence. If either suffix is not present, there is no effect of the covariate on the corresponding parameter.

All of the parameters in Table B.6 are significantly different from zero and all other related parameters. All except one parameter have signs which are consistent with established theory. B_COST_FUEL_NVO_DL1 is positive, suggesting that driving licence holders with no vehicles in the household have increased utility for driving for increasing fuel costs. As members of households with no available vehicles, driving trips made by this group must either be made by taxi or as a passenger in another household's car. As such, their utility of driving is not directly affected by fuel cost. It is therefore reasonable to remove this parameter from the model. Doing so results in an increase (lower is better) in the AIC (from 70365.2 to 70392.0), but a smaller relative decrease in the out-of-sample test GMPCA (from 0.5116 to 0.5115) and an improvement in the test accuracy (from 0.7433 to 0.7435).

7.4.2.1 Comparison with ML models

100 iterations of bootstrap validation are performed on the *full assisted specification* model in order to estimate the distribution of the out-of-sample performance and compare it to the ML classifiers from Section 7.3.1. The distributions of negative CEL for all models are shown in Fig. 7.14. Within the figure, the *assisted specification* model is referred to as AS-MNL (assisted-specification MNL), and the baseline RUM is referred to as MS-NL (manual specification NL). From the graph it is clear to see that the modifications made to the

AS-MNL in Section 7.4.1 have significantly improved the performance of the RUM over the MS-NL. The assisted specification MNL outperforms all models except the GBDT model.

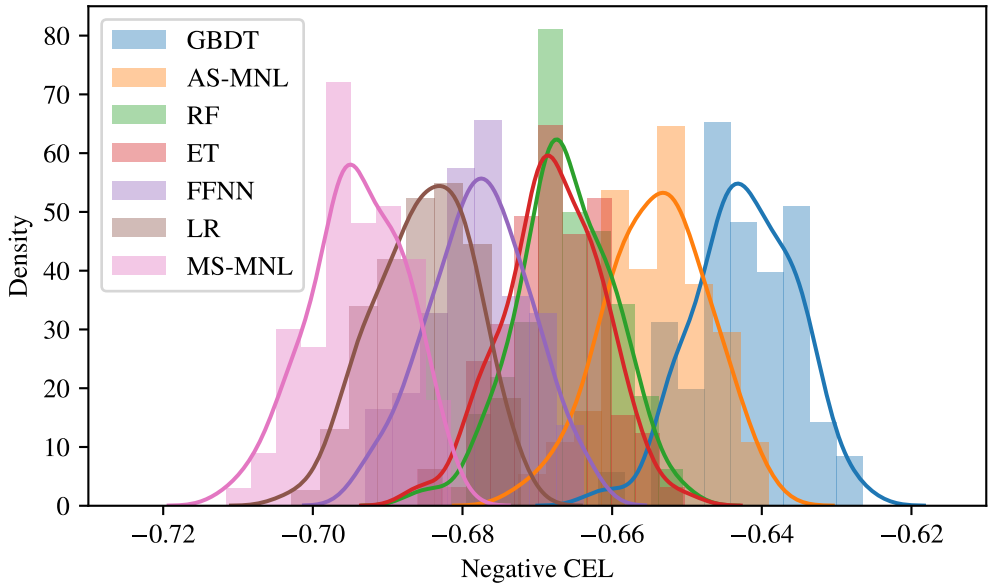


Fig. 7.14 Kernel density plots and histograms of out-of-sample CEL for 100 iterations of bootstrapping for assisted specification MNL (AS-MNL), compared to other classifiers.

Table 7.14 shows pairwise n at which the AS-MNL performance is significantly different from the the model in the column at the 5 % significance level, according to the power-based approach presented in Section 5.2.6.

Table 7.14 Pairwise grid of test size at which *assisted specification* model performance is significantly different from model in column.

GBDT	RF	ET	FFNN	LR	MS-NL
309	300	265	155	117	91

Despite significant improvement in predictive performance, the AS-MNL still maintains the interpretability provided by the linear model (see Table B.6), and the robust behavioural model which is consistent with expected behaviour.

The results in this section are intended to form a first investigation into a assisted specification approach and are not designed to be exhaustive. As such, it is expected that the AS-MNL could be further improved by analysing other features and interactions in the GBDT model and making appropriate modifications to the utility specifications in the RUM. A formal framework for automated specification is suggested as an area for further work.

7.5 Summary

This chapter presents the experimental results for the investigations carried out within the thesis, including an application of state-of-practice RUMs to the new data developed for the thesis, a systematic study of ML classification techniques for mode-choice prediction, and a first investigation into a assisted specification approach.

This summary of the chapter is divided into two sections. Firstly, Section 7.5.1 uses the experimental results to review the methodological framework implemented within the thesis and demonstrate how the results of previous studies may have been affected by the technical limitations identified in Chapter 3. Next, Section 7.5.2 presents the new findings from the experimental results using the robust modelling framework.

7.5.1 Modelling framework

The following sections use the experimental results to review the modelling framework developed for this thesis, focusing in turn on datasets, performance estimation, model optimisation, and model selection.

Datasets

Adding mode-alternatives to the dataset represents a significant contribution to both random utility and ML mode choice prediction. The results in Table 7.2 show that the data generation framework presented in Chapter 6 allows for a RUM to be fitted which performs significantly better than one fitted on only the original socio-economic and trip profile data from the LTDS. The combined data allows for a robust behavioural model to be trained, with estimated parameter signs and values consistent with behavioural theory.

As with the RUMs, Section 7.3.3 finds that the GBDT classifier trained on the choice-set data significantly outperforms a model trained using only the raw input data. Furthermore, the difference in performance between these two models is larger than that between any classifiers trained on the full dataset, signifying that adding mode-alternative attributes to the dataset plays a more significant role in the overall performance of the resulting mode-choice classifiers than the choice of model algorithm.

Performance estimation

The results in Table 7.7 show that in-sample validation is an inconsistent and optimistically biased estimator of predictive performance. Out-of-sample validation is therefore used instead.

Section 7.3.2 shows that trip-wise sampling for model validation with hierarchical data significantly overestimates true out-of-sample predictive performance, particularly for flexible non-linear classifiers (e.g. EL methods). Furthermore, the results indicate that models optimised using incorrect sampling perform worse on out-of-sample data than those optimised using appropriate sampling methods. Household-wise sampling during model optimisation and performance estimation addresses these issues and allows for a more accurate estimate of true out-of-sample predictive performance.

Whilst using household-wise sampling allows for a more accurate performance estimate than trip-wise sampling or in-sample validation, Table 7.7 indicates that cross-validation (using household-wise sampling) still marginally overestimates true predictive performance. The external validation scheme used in this thesis, where the model is trained on previous years' data in order to predict a subsequent future year unseen during model training/ optimisation, closely emulates the intended use-case for a predictive model, i.e. it tests the ability of the model to successfully *extrapolate* the findings from historic trips to future unseen trips. This is in contrast to the validation schemes commonly used in ML practice, where test sets/folds are sampled randomly from the dataset. These validation schemes therefore represent the ability of the model to *interpolate* the data.

Figure 7.5 shows the use of discrete classification results in highly inaccurate mode-share predictions, over-representing the mode-shares for the majority class, and under-representing the minority class. Probabilistic simulation resolves this issue, resulting in predicted mode-shares which are a much closer representation of the ground-truth mode-shares. As such, mode choice classifier performance should be evaluated using metrics based on the outputted choice probabilities.

Figure 7.4 indicates that all of the classifiers used in this thesis are capable of outputting well calibrated choice probabilities.

Model optimisation

Table 7.8 and Fig. 7.6 show that performance is highly dependent on chosen hyper-parameter values, indicating that hyper-parameter selection should be performed during model development.

SMBO is shown to be a rigorous search scheme which automates the model optimisation process and allows for impartial comparison of different classifiers. SMBO has high transparency and repeatability, requiring only a search space and number iterations to be specified.

As discussed, the cross-validation scheme used for model optimisation results in optimistically biased estimates of out-of-sample performance. This is due to *data-leakage*

from the validation data allowing model hyper-parameters to be overfit to noise in the data. By optimising the models on the training data only, the external validation represents true out-of-sample model performance.

Model selection

The approaches proposed in Chapter 5 allow for formal investigation into the significance of differences in probabilistic classifiers. The approaches work for both parametric and non-parametric models, and so allow for RUMs to be directly compared with ML models, unlike traditional parametric significance tests.

7.5.2 Key findings

As shown in Section 7.5.1, the modelling framework developed for this thesis successfully addresses the technical limitations of previous studies and enables a formal comparison of classification techniques for predicting passenger mode-choice. The following sections summarise the key new findings from the experimental results which make use of this robust modelling framework.

The GBDT model is the highest performing classifier for this dataset. Section 7.4 reveals EL models to be the highest performing among the tested classifiers, due to the flexibility and efficiency of DT ensembles for modelling non-linear relations. The GBDT classifier achieves the stand-out highest performance of the EL algorithms, and of all the classifiers investigated in this thesis. There are two primary reasons for this: (1) the DTs in the GBDT ensemble directly evaluate the choice probabilities for each mode and (2) the GBDT ensemble size can be set dynamically to maximise out-of-sample predictive performance.

Relative differences in predictive performance between classifiers are far smaller than has been suggested by previous research. Whilst the tests proposed in Chapter 5 show that the performance differences between classifiers are significant, the results show the differences in relative predictive performance between classifiers are far smaller than has been suggested by previous research. As discussed in Section 7.5.1, adding mode-alternative attributes through the data-generation framework represents larger performance contribution to the mode choice models than the choice of model algorithm.

Crucially, the results in this thesis identify a smaller performance gap between the highest performing ML methods and state-of-practice RUMs, when used with the detailed choice-set dataset developed for this thesis. This is highly relevant to the ML-RUM tradeoff discussed

in Section 7.3.1.6, between the higher predictive power and flexibility of the ML algorithms, and the behavioural robustness and interpretability of the RUMs.

The structure of a fitted EL classifier can be used to improve the performance of a RUM through informing the utility function specification in an assisted specification approach. The assisted specification approach presented in Section 7.4 demonstrates four different ways in which the structure of a fitted EL model can be successfully used to inform the utility specification for a RUM in order to improve model fit:

1. adding a non-linear function of a continuous input variable informed by the distribution of split-points for that variable in the EL model,
2. using heuristic binning of a continuous variable informed by the modal split-point values for that variable in the EL model,
3. adding second order covariate interactions with the choice probabilities informed by feature interaction importances from the EL model,
4. adding third order covariate interactions with the choice probabilities informed by feature interaction importances from the EL model.

These are all applied to the full assisted specification MNL model presented in Table B.6, which achieves significantly better performance than all but the highest performing ML classifier, whilst maintaining a robust, interpretable behavioural model.

Chapter 8

Conclusions and further work

8.1 Overview

This chapter presents the conclusions and further work. Firstly Section 8.2 summarises the work carried out in the thesis. Section 8.2.1 evaluates the work against the general limitations of the previous research identified in Chapter 3, and Section 8.2.2 gives details of the publications fed by the material in this thesis. Finally, Section 8.3 critiques the methodology, and proposes directions for future research.

8.2 Summary of work

This thesis develops a novel approach for urban travel mode choice prediction and applies it to trip records in the Greater London area. The new approach consists of two parts: (1) a data generation framework which combines multiple data-sources to build trip datasets containing the likely mode-alternative options faced by a passenger at the time of travel, and (2) a modelling framework which makes use of these datasets to fit, optimise, validate, and select mode choice classifiers. This approach is used to compare the relative predictive performance of a complete suite of current Machine Learning (ML) classification algorithms, as well as traditional utility-based choice models. Furthermore, a new assisted specification approach, where a fitted ML classifier is used to inform the utility function structure in a Random Utility Model (RUM) is then explored.

The background and motivation for mode choice prediction is outlined in Chapter 2. The chapter evaluates the limitations of the state-of-practice random utility approach for mode choice modelling and establishes the need for techniques which can provide a more detailed

understanding of mode choice at higher spatial and temporal granularity. A number of ML classification algorithms are introduced as possible alternatives to RUMs.

There is a large body of existing research into ML applications to mode choice prediction. However, past studies are affected by several significant methodological limitations. To explore this formally, Chapter 3 conducts a systematic review of ML methodologies for mode choice prediction. The review covers 63 studies selected through an exhaustive search of three online publication databases. Ten technical limitations are identified in the existing methodologies, covering the datasets, performance estimation, model optimisation, and model selection techniques used in each study. Each limitation is present in a sizeable proportion of the studies, and each study is found to be individually affected by at least three of the limitations.

To understand each limitation in the systematic review further, Chapter 4 establishes a theoretical framework to present predictive classification within a unified notation. The theoretical background of each technical limitation is introduced within this context.

The results of the systematic review highlight the need for a comprehensive comparative study of classification techniques for mode choice prediction. Chapter 5 establishes a new modelling framework for this purpose, within which each technical limitation is individually addressed. The chapter also proposes novel statistical approaches for evaluating the differences between two probabilistic classifiers, as well as the assisted specification approach.

Chapter 6 develops a framework for generating datasets of passenger mode choice-sets for the study. The framework combines individual trip records with closely matched trajectories alongside their corresponding mode-alternatives, and precise estimates of public transport fares and car operating costs. The framework is used with trip diary data from the London Travel Demand Survey (LTDS) to create a new trip dataset for London. This represents the most comprehensive and closely tailored travel dataset for estimating travel choices in a major metropolitan area. The framework has been implemented within an automated process, which allows for the fast assembly of similar datasets from historic trip data from any geographical region.

Chapter 7 presents the results for the investigations in the thesis, including the comparative study of probabilistic classification techniques for mode choice prediction. This includes several ML algorithms, including Logistic Regression (LR), Artificial Neural Networks (ANNs), Ensemble Learning (EL), and Support Vector Machines (SVMs), as well as conventional RUMs (Multinomial Logit (MNL) and Nested Logit (NL)).

The Gradient Boosting Decision Trees (GBDT) model is found to be the highest performing classifier for this task, due to (a) its highly flexible structure which allows it to

generalise efficiently to high-order, non-linear, interactions between input features and mode choice; (b) the Decision Trees (DTs) in the GBDT ensemble directly evaluating the choice probabilities for each mode; and (c) the ability to set the GBDT ensemble-size dynamically to maximise out-of-sample predictive performance.

The implications of the limitations identified in the systematic review are assessed experimentally. The results confirm that the limitations do significantly impact the performance estimates and introduce a positive bias towards non-linear classifiers which can easily overfit to input training data. This has resulted in the performance of such classifiers being overstated in previous studies. Whilst the highest performing ML models do outperform the RUMs, the gap in performance is far smaller than has been suggested by previous research.

The results show that adding mode-alternative attributes to the dataset through the data generation framework contributes substantially to the predictive performance of both the resultant RUMs and ML models and plays a more significant role in the overall performance of the resulting mode-choice classifiers than the choice of model algorithm.

The relative merits and suitability of GBDT models and RUMs are assessed in depth. The discussion highlights that whilst GBDT models show greater predictive performance, RUMs have higher interpretability, as the model parameters indicate both the importance and direction of impact (positive/negative) of each individual feature on the utility of each mode.

A key limitation of RUMs identified by the research is the complexity in specifying interactions of socio-economic variables with trip attributes (e.g. duration and cost), and the high dimensionality of the parameters in the resulting models. As such, an assisted specification approach is explored, where the first and second order feature importances and split-points from a EL model are used to infer the key socio-economic covariates and non-linear transformations of input variables in the utility specifications for a RUM.

The results for the assisted specification approach demonstrate that the structure of a fitted EL model can be successfully used to inform the utility specification for a RUM in order to improve model fit. The full assisted specification MNL achieves significantly better performance than all but the highest performing ML classifier (GBDT), whilst maintaining a robust, interpretable behavioural model.

8.2.1 Evaluation against general limitations of existing work

As discussed in detail in Section 5.2, the methodological approach in this thesis addresses each of the *technical* limitations raised in the systematic review in Chapter 3. Furthermore, the following sections briefly describe how the material in this thesis addresses each of the *general* limitations raised.

GL1: Limited number of studies which systematically compare several classifiers on the same task

A comparative study of ML classifiers for mode choice prediction is conducted, which includes six ML classification algorithms with conventional MNL and NL models.

GL2: Relatively low number of investigations into EL algorithms, in particular GBDT

Three EL algorithms are investigated: Random Forests (RFs), Extremely randomised Trees (ET), and GBDT.

GL3: Inconsistent representation of Discrete Choice Models (DCMs) in ML studies

Logit models are included both as a RUM (MNL) and a ML classifier (LR). The utility specifications for the RUMs in the study are optimised using a thorough manual sequential search, exploiting established behavioural theory as well as parametric significance tests.

GL4: Not describing the dataset and modelling process in sufficient detail

The methodology and dataset are both presented in detail in Chapters 5 and 6.

GL5: Shortage of studies using large datasets to investigate mode choice

By processing three years of trip diaries from a substantial household travel survey, a large dataset of 81 086 trips has been created, which is used to investigate passenger mode choice.

GL6: Lack of relevant, openly available datasets including mode-alternative attributes

The dataset, which includes mode-alternative attributes, has been made openly available under the Creative Commons Attribution (CC-BY) license.

GL7: Not presenting specific model hyper-parameters

The hyper-parameter search spaces are given in Appendix A.1, with optimised values for all classifiers given in Appendix A.2.

8.2.2 Publications

The material in this thesis has fed into several papers, including a journal article (Hillel, Elshafie, and Jin 2018) and three conference papers (Hillel, Bierlaire, et al. 2018a; Hillel, Bierlaire, et al. 2018b; Hillel, Guthrie, et al. 2016).

8.3 Limitations and further work

This thesis develops a novel approach for urban mode choice prediction consisting of two parts: (i) a data generation framework which combines multiple data-sources to build trip datasets containing the likely mode-alternative options faced by a passenger at the time of travel, and (ii) a modelling framework which makes use of these datasets to fit, optimise, validate, and select mode choice classifiers. Furthermore, a new assisted specification approach is explored, which uses a fitted ML classifier to inform the utility function structure in a utility-based choice model.

The following sections critique the methodology across these three components and identify three primary areas for further research.

8.3.1 Critique of methodology

The following subsections discuss the methodological limitations associated with the data generation framework, the modelling framework, and the assisted specification approach.

Data generation framework

The data generation framework is shown to substantially benefit the predictive performance of both the RUMs and ML models. There are however some limitations of the data generation framework which, if addressed, could further improve the information contribution.

Firstly, whilst the public transport and driving requests are matched time-of-day and day-of-week to the original trip in order to provide typical conditions at the time of travel, they do not describe the *exact* network conditions as they were at the time of travel (e.g. train delays, road closures, etc). This means the dataset cannot be considered as a precise reflection of the network conditions for the choices made. However, adding this information to the data would require a dataset of historical traffic and public transport conditions with full coverage of the study area and period, which is not currently available. In a reasonably well run transport network, with no widespread or significant day-to-day variations in travel conditions, the impacts of road traffic on mode-choice can still be investigated via the traffic variability attribute. Additionally, by only using the most recent three years of data, the discrepancies arising from long term changes to the transport network (e.g. new roads, stations, bus routes, etc.) are minimised.

Secondly, parking costs and public transport season tickets/daily Oyster usage caps are omitted from the estimation of driving and public transport costs respectively. However, adding this information to the data-generation framework would require much more detailed

trip and person specifications to be identified, and would require this information to be specified for the future trips being predicted. Furthermore, there is high potential for introducing input features which are dependent on output choice (see Section 3.3.3).

Finally, the original LTDS data contains only one-day travel diaries. This means it is not possible to investigate habitual behaviours or changes in behaviour over time at the level of individual travellers. Instead, this would require a panel data format.

Modelling framework

The modelling framework addresses the technical limitations of the previous research. The results show that these limitations affect the modelling results. Crucially, the differences in performance between the classifiers using the corrected modelling framework are far smaller than has been suggested by previous research.

The most important of the limitations identified in the previous work is the use of incorrect sampling for hierarchical data (see Sections 6.4.2 and 7.3.2). Household-wise sampling is shown to resolve this issue by preventing data leakage from grouped trips. However, the modelling framework still uses a trip-based analysis, assuming all trips are independent. Model performance could be improved further by specifically modelling these hierarchical relationships, e.g. by using a tour or activity-based analysis. This is suggested as an area for further work (see Section 8.3.2).

Additionally, the analysis in the modelling framework is used to predict the distance-based main mode for the whole trip. Both the LTDS data and the Google directions Application Programming Interface (API) results contain information about the individual steps/stages in the route. This presents the possibility for a deeper level of analysis, for instance investigating mixed mode trips (e.g. getting a lift by car to a train station).

Assisted specification approach

The assisted specification approach is shown to substantially improve performance of a RUM, whilst maintaining a robust, interpretable behavioural model. However, the approach is still conceptual, and has not yet been formalised into a specific methodological framework (in contrast to the data generation and modelling frameworks, both of which have been fully automated). This means it is not possible to formally compare the performance of the proposed assisted specification approach with an open-ended manual search for specifying the utility functions.

Additionally, the methods used to identify the non-linear relationships in the RUM variables and covariates through investigating the split points from the EL classifier are also

informal and based on visual inspection. There may be examples where the distributions are not as clear, and so the choice of which transformation to apply to the data may not be immediately evident.

8.3.2 Directions for future research

Three primary areas for further research are identified: (i) formalising the assisted specification approach into an automated process, (ii) incorporating the proposed approach into transport simulation models, and (iii) investigating tour and activity-based modelling approaches.

Formalising the assisted specification approach into an automated process

The results for the assisted specification approach are promising, but further work is needed to formalise the approach into a framework which can then be automated. This approach could have wider implications than mode-choice modelling, as a general-purpose methodology to create highly interpretable classifiers for multiple applications.

Incorporating the proposed approach into transport simulation models

The motivation for this research is to provide a deeper understanding of passenger flows in order to be able manage increased travel demand. Whilst the results show that the frameworks implemented for this thesis provide a deeper understanding of individual mode choice, they have not yet been incorporated into city-scale simulation models.

Investigating tour and activity-based modelling approaches

The modelling framework developed in this thesis uses single trips (Origin-Destination (O-D) pairs) as the unit of analysis, treating all trip mode-choices as independent. As the LTDS contains daily activity patterns for each member of the household, there is an opportunity to investigate sequential mode-choice in a tour-based or activity-based approach.

Appendices

A Hyper-parameter search spaces and optimised values

A.1 Hyper-parameter search spaces

Table A.1 Hyper-parameter search space for LR model

Hyper-parameter	Distribution	Range
`penalty`	Uniform choice	L1, L2
`C`	Log-uniform	$1 \times 10^{-5} - 1 \times 10^{5}$
`class_weight`	Uniform choice	None, *balanced*
`solver`	Fixed	*saga*
`multi_class`	Fixed	*multinomial*

Table A.2 Hyper-parameter search space for Feed-Forward Neural Network (FFNN) model

Hyper-parameter	Distribution	Range
`n_hiddenlayers`	Uniform (int)	$1-3$
`optimizer`	Uniform choice	*adadelta*, *adam*, *rmsprop*
`batch_size`	Uniform (int)	$16-256$
`epochs`	Uniform (int)	$1-200$
First hidden layer		
`n_neurons_1`	Uniform (int)	$4-128$
`activation_1`	Uniform choice	*ReLU*, *ELU*, *SELU*, *softplus*, *softsign*, *tanh*, *sigmoid*, *hard sigmoid*, *linear*
`dropout_1`	Uniform	$0-0.75$
Second hidden layer (`n_hiddenlayers` > 1)		
`n_neurons_2`	Uniform (int)	$4-128$
`activation_2`	Uniform choice	*ReLU*, *ELU*, *SELU*, *softplus*, *softsign*, *tanh*, *sigmoid*, *hard sigmoid*, *linear*
`dropout_2`	Uniform	$0-0.75$
Third hidden layer (`n_hiddenlayers` $= 3$)		
`n_neurons_3`	Uniform (int)	$4-128$
`activation_3`	Uniform choice	*ReLU*, *ELU*, *SELU*, *softplus*, *softsign*, *tanh*, *sigmoid*, *hard sigmoid*, *linear*
`dropout_3`	Uniform	$0-0.75$

Table A.3 Hyper-parameter search space for RF and ET models

Hyper-parameter	**Distribution**	**Range**
`n_estimators`	Log-uniform (int)	$10-3000$
`criterion`	Uniform choice	*gini*, *entropy*
`max_depth`	Uniform	$1-14$
`min_samples_leaf`	Uniform choice	1, `msl_int`
`msl_int`	Log-uniform (int)	$2-50$
`max_features`	Probabilistic choice	*sqrt* $(p=0.2)$, *log2* $(p=0.1)$, None $(p=0.1)$, `mf_frac` $(p=0.6)$
`mf_frac`	Uniform	$0-1$
`bootstrap`	Uniform choice	True, False

Table A.4 Hyper-parameter search space for GBDT model

Hyper-parameter	**Distribution**	**Range**
`max_depth`	Uniform	$1-14$
`gamma`	Log-uniform	$0-5$
`min_child_weight`	Log-uniform (int)	$1-100$
`max_delta_step`	Log-uniform (int)	$0-10$
`subsample`	Uniform	$0.5-1$
`colsample_bytree`	Uniform	$0.5-1$
`colsample_bylevel`	Uniform	$0.5-1$
`reg_alpha`	Log-uniform	$0-1$
`reg_lambda`	Log-uniform	$1-4$
`learning_rate`	Fixed	0.01
`n_estimators`	See methodology	$1-6000$
`extra_stopping_rounds`	Fixed	50

Table A.5 Hyper-parameter search space for SVM model. n = number of features in model (43).

Hyper-parameter	**Distribution**	**Range**
`C`	Log-uniform	$1 \times 10^{-4} - 500$
`kernel`	Uniform choice	*linear*, *poly*, *rbf*, *sigmoid*
`degree`	Uniform (*poly* kernel only)	$2 - 6$
`gamma`	Log-uniform (*rbf*, *poly*, *sigmoid* kernels)	$0.001/n - 1000/n$
`coef0`	Probabilistic choice (*poly*, *sigmoid* kernels)	0 ($p = 0.3$), `coef0_float` ($p = 0.7$)
`coef0_float`	Uniform	$0 - 10$ (*poly*), $-10 - 10$ (*sigmoid*)
`shrinking`	Uniform choice	True, False
`class_weight`	Uniform choice	None, *balanced*
`probability`	Fixed	True

A.2 Optimised hyper-parameter values

A.2.1 Grouped (household-wise) sampling

Table A.6 Optimised hyper-parameter values for LR model with grouped (household-wise) sampling

Hyper-parameter	**Value**
`penalty`	L1
`C`	0.7715
`class_weight`	None
`solver`	*saga*
`multi_class`	*multinomial*

Table A.7 Optimised hyper-parameter values for FFNN model with grouped (household-wise) sampling

Hyper-parameter	**Value**
`n_hiddenlayers`	1
`optimizer`	*adam*
`batch_size`	57
`epochs`	150
Hidden layer	
`n_neurons_1`	108
`activation_1`	*relu*
`dropout_1`	0.1462

Table A.8 Optimised hyper-parameter values for RF model with grouped (household-wise) sampling

Hyper-parameter	**Value**
`n_estimators`	1578
`criterion`	*entropy*
`max_depth`	13
`min_samples_leaf`	15
`max_features`	0.4148
`bootstrap`	True

Table A.9 Optimised hyper-parameter values for ET model with grouped (household-wise) sampling

Hyper-parameter	**Value**
`n_estimators`	2609
`criterion`	*entropy*
`max_depth`	13
`min_samples_leaf`	1
`max_features`	0.8179
`bootstrap`	True

Table A.10 Optimised hyper-parameter values for GBDT model with grouped (household-wise) sampling

Hyper-parameter	**Range**
`max_depth`	6
`gamma`	5.439×10^{-3}
`min_child_weight`	36
`max_delta_step`	4
`subsample`	0.65
`colsample_bytree`	0.65
`colsample_bylevel`	0.55
`reg_alpha`	4.823×10^{-4}
`reg_lambda`	2.572
`learning_rate`	0.01
`n_estimators`	1472

Table A.11 Optimised hyper-parameter values for SVM model with grouped (household-wise) sampling

Hyper-parameter	**Value**
`C`	227.499
`kernel`	*linear*
`degree`	NA
`gamma`	NA
`coef0`	NA
`shrinking`	True
`class_weight`	None
`probability`	True

A.2.2 Random (trip-wise) sampling

Table A.12 Optimised hyper-parameter values for LR model with random (trip-wise) sampling

Hyper-parameter	Value
`penalty`	L1
`C`	1.228
`class_weight`	None
`solver`	*saga*
`multi_class`	*multinomial*

Table A.13 Optimised hyper-parameter values for FFNN model with random (trip-wise) sampling

Hyper-parameter	Value
`n_hiddenlayers`	3
`optimizer`	*rmsprop*
`batch_size`	124
`epochs`	115
Hidden layer 1	
`n_neurons_1`	112
`activation_1`	*selu*
`dropout_1`	0.1352
Hidden layer 2	
`n_neurons_1`	58
`activation_1`	*relu*
`dropout_1`	0.0339
Hidden layer 3	
`n_neurons_1`	46
`activation_1`	*relu*
`dropout_1`	0.2062

Table A.14 Optimised hyper-parameter values for RF model with random (trip-wise) sampling

Hyper-parameter	**Value**
`n_estimators`	726
`criterion`	*entropy*
`max_depth`	14
`min_samples_leaf`	1
`max_features`	0.3897
`bootstrap`	True

Table A.15 Optimised hyper-parameter values for ET model with random (trip-wise) sampling

Hyper-parameter	**Value**
`n_estimators`	1426
`criterion`	*entropy*
`max_depth`	14
`min_samples_leaf`	1
`max_features`	None
`bootstrap`	False

Table A.16 Optimised hyper-parameter values for GBDT model with random (trip-wise) sampling

Hyper-parameter	**Value**
`max_depth`	13
`gamma`	0.03855
`min_child_weight`	1
`max_delta_step`	9
`subsample`	0.9
`colsample_bytree`	0.55
`colsample_bylevel`	0.8
`reg_alpha`	0.03022
`reg_lambda`	3.473
`learning_rate`	0.01
`n_estimators`	2547

Table A.17 Optimised hyper-parameter values for SVM model with random (trip-wise) sampling

Hyper-parameter	Value
`C`	11.486
`kernel`	*rbf*
`degree`	NA
`gamma`	0.0772
`coef0`	NA
`shrinking`	True
`class_weight`	*balanced*
`probability`	True

A.2.3 Raw data GBDT model

Table A.18 Optimised hyper-parameter values for *raw-data* GBDT model

Hyper-parameter	Value
`max_depth`	5
`gamma`	3.393×10^{-4}
`min_child_weight`	35
`max_delta_step`	2
`subsample`	0.75
`colsample_bytree`	0.95
`colsample_bylevel`	0.5
`reg_alpha`	0.5894
`reg_lambda`	1.882
`learning_rate`	0.01
`n_estimators`	1359

B Random utility model parameter values

B.1 Initial random utility models

Table B.1 Parameter estimates for RUM 1: mode-alternative attributes only.

Parameter number	Description	Coeff. estimate	Robust Asympt. std. error	*t*-stat	*p*-value
1	ASC_CYCLING	-4.88	0.0631	-77.35	0.00
2	ASC_DRIVING	-1.49	0.0443	-33.67	0.00
3	ASC_PT	-2.39	0.0455	-52.45	0.00
4	B_COST_DRIVING	-0.103	0.00550	-18.69	0.00
5	B_COST_PT	-0.197	0.00974	-20.24	0.00
6	B_TIME_CYCLING	-4.86	0.159	-30.45	0.00
7	B_TIME_DRIVING	-4.12	0.151	-27.23	0.00
8	B_TIME_ACCESS_PT	-4.72	0.136	-34.57	0.00
9	B_TIME_BUS_PT	-1.97	0.0881	-22.31	0.00
10	B_TIME_INTERCHANGEWAIT_PT	-4.87	0.251	-19.45	0.00
11	B_TIME_RAIL_PT	-1.72	0.160	-10.77	0.00
12	B_TIME_WALKING	-8.55	0.134	-63.71	0.00
13	B_TRAFFICVARIABILITY_DRIVING	-3.06	0.0698	-43.75	0.00

Table B.2 Parameter estimates for RUM 2: LTDS socio-economic and trip profile only.

Parameter number	Description	Coeff. estimate	Robust Asympt. std. error	t-stat	p-value
1	ASC_CYCLING	-5.19	0.223	-23.23	0.00
2	ASC_DRIVING	-4.21	0.115	-36.62	0.00
3	ASC_PT	-3.51	0.0813	-43.16	0.00
4	B_AGE_CHILD_DRIVING	0.802	0.0517	15.51	0.00
5	B_AGE_CHILD_PT	0.344	0.0486	7.09	0.00
6	B_AGE_PENSIONER_CYCLING	-0.471	0.132	-3.58	0.00
7	B_AGE_PENSIONER_DRIVING	0.656	0.0500	13.13	0.00
8	B_AGE_PENSIONER_PT	0.954	0.0493	19.34	0.00
9	B_DAY_SAT_CYCLING	-0.364	0.101	-3.62	0.00
10	B_DAY_SAT_DRIVING	-0.132	0.0590	-2.24	0.02
11	B_DAY_SAT_PT	0.332	0.0646	5.15	0.00
12	B_DAY_WEEK_DRIVING	-0.310	0.0467	-6.64	0.00
13	B_DAY_WEEK_PT	0.489	0.0519	9.44	0.00
14	B_DEPARTURE_AMPEAK_CYCLING	0.417	0.195	2.14	0.03
15	B_DEPARTURE_AMPEAK_DRIVING	-0.757	0.0826	-9.16	0.00
16	B_DEPARTURE_INTERPEAK_CYCLING	-0.278	0.0760	-3.65	0.00
17	B_DEPARTURE_INTERPEAK_DRIVING	-0.0780	0.0315	-2.48	0.01
18	B_DEPARTURE_PMPEAK_CYCLING	0.339	0.0690	4.91	0.00
19	B_DEPARTURE_PMPEAK_DRIVING	0.405	0.0359	11.27	0.00
20	B_DEPARTURE_PMPEAK_PT	0.163	0.0366	4.45	0.00
21	B_DISTANCE_CYCLING	1.55	0.0346	44.84	0.00
22	B_DISTANCE_DRIVING	1.67	0.0334	50.16	0.00
23	B_DISTANCE_PT	1.79	0.0334	53.78	0.00
24	B_DRIVINGLICENCE_CYCLING	0.637	0.0665	9.58	0.00
25	B_DRIVINGLICENCE_DRIVING	0.938	0.0431	21.76	0.00
26	B_DRIVINGLICENCE_PT	-0.278	0.0401	-6.95	0.00
27	B_FEMALE_CYCLING	-0.811	0.0628	-12.92	0.00
28	B_FEMALE_DRIVING	0.234	0.0312	7.49	0.00
29	B_FEMALE_PT	0.232	0.0321	7.23	0.00
30	B_PURPOSE_B_CYCLING	1.26	0.143	8.83	0.00
31	B_PURPOSE_B_DRIVING	0.481	0.0846	5.69	0.00
32	B_PURPOSE_B_PT	0.732	0.0898	8.16	0.00
33	B_PURPOSE_HBE_CYCLING	0.337	0.133	2.53	0.01
34	B_PURPOSE_HBE_PT	0.347	0.0447	7.77	0.00
35	B_PURPOSE_HBO_CYCLING	0.571	0.101	5.64	0.00
36	B_PURPOSE_HBO_DRIVING	0.454	0.0359	12.66	0.00
37	B_PURPOSE_HBO_PT	0.266	0.0422	6.30	0.00
38	B_PURPOSE_HBW_CYCLING	1.07	0.118	9.08	0.00
39	B_PURPOSE_HBW_DRIVING	-0.409	0.0658	-6.22	0.00
40	B_PURPOSE_HBW_PT	0.365	0.0676	5.40	0.00
41	B_VEHICLEOWNERSHIP_1_DRIVING	2.29	0.0412	55.57	0.00
42	B_VEHICLEOWNERSHIP_1_PT	-0.369	0.0350	-10.54	0.00
43	B_VEHICLEOWNERSHIP_2_DRIVING	2.76	0.0469	58.91	0.00
44	B_VEHICLEOWNERSHIP_2_PT	-0.585	0.0455	-12.87	0.00
45	B_WINTER_CYCLING	-0.321	0.0806	-3.98	0.00
46	B_WINTER_DRIVING	0.110	0.0288	3.83	0.00

Table B.3 Parameter estimates for RUM 3: combined mode-alternative attributes and LTDS socio-economic/trip profile.

Parameter number	Description	Coeff. estimate	Robust Asympt. std. error	*t*-stat	*p*-value
1	ASC_CYCLING	-4.95	0.117	-42.15	0.00
2	ASC_DRIVING	-4.24	0.0838	-50.53	0.00
3	ASC_PT	-3.03	0.0793	-38.24	0.00
4	B_AGE_CHILD_DRIVING	0.774	0.0548	14.12	0.00
5	B_AGE_CHILD_PT	0.245	0.0531	4.61	0.00
6	B_AGE_PENSIONER_CYCLING	-0.447	0.132	-3.37	0.00
7	B_AGE_PENSIONER_DRIVING	0.541	0.0513	10.54	0.00
8	B_AGE_PENSIONER_PT	0.834	0.0537	15.54	0.00
9	B_COST_DRIVE	-0.118	0.00628	-18.79	0.00
10	B_COST_PT	-0.0925	0.0146	-6.33	0.00
11	B_DAY_SAT_CYCLING	-0.338	0.0990	-3.42	0.00
12	B_DAY_SAT_PT	0.204	0.0494	4.13	0.00
13	B_DAY_WEEK_DRIVING	-0.163	0.0370	-4.41	0.00
14	B_DAY_WEEK_PT	0.346	0.0470	7.37	0.00
15	B_DEPARTURE_INTERPEAK_CYCLING	-0.221	0.0751	-2.95	0.00
16	B_DEPARTURE_INTERPEAK_DRIVING	-0.127	0.0340	-3.74	0.00
17	B_DEPARTURE_PMPEAK_CYCLING	0.347	0.0701	4.95	0.00
18	B_DEPARTURE_PMPEAK_DRIVING	0.431	0.0375	11.48	0.00
19	B_DEPARTURE_PMPEAK_PT	0.162	0.0375	4.31	0.00
20	B_DISTANCE_CYCLING	0.405	0.107	3.80	0.00
21	B_DISTANCE_DRIVING	0.654	0.0971	6.74	0.00
22	B_DISTANCE_PT	0.656	0.0974	6.74	0.00
23	B_DRIVINGLICENCE_CYCLING	0.674	0.0702	9.60	0.00
24	B_DRIVINGLICENCE_DRIVING	1.06	0.0451	23.57	0.00
25	B_DRIVINGLICENCE_PT	-0.298	0.0407	-7.31	0.00
26	B_FEMALE_CYCLING	-0.810	0.0636	-12.73	0.00
27	B_FEMALE_DRIVING	0.191	0.0321	5.96	0.00
28	B_FEMALE_PT	0.217	0.0325	6.67	0.00
29	B_PURPOSE_B_CYCLING	1.09	0.131	8.36	0.00
30	B_PURPOSE_B_DRIVING	0.418	0.0860	4.85	0.00
31	B_PURPOSE_B_PT	0.778	0.0916	8.49	0.00
32	B_PURPOSE_HBE_DRIVING	-0.553	0.0493	-11.21	0.00
33	B_PURPOSE_HBE_PT	0.372	0.0541	6.88	0.00
34	B_PURPOSE_HBO_CYCLING	0.400	0.0805	4.97	0.00
35	B_PURPOSE_HBO_PT	0.265	0.0370	7.17	0.00
36	B_PURPOSE_HBW_CYCLING	0.765	0.100	7.63	0.00
37	B_PURPOSE_HBW_DRIVING	-0.686	0.0634	-10.81	0.00
38	B_PURPOSE_HBW_PT	0.279	0.0680	4.11	0.00
39	B_TIME_CYCLING	-2.45	0.612	-4.00	0.00
40	B_TIME_DRIVING	-4.32	0.200	-21.56	0.00
41	B_TIME_ACCESS_PT	-4.41	0.160	-27.62	0.00
42	B_TIME_BUS_PT	-1.92	0.117	-16.50	0.00
43	B_TIME_INTERCHANGEWAIT_PT	-5.02	0.317	-15.83	0.00
44	B_TIME_INTERCHANGEWALK_PT	-2.89	1.01	-2.85	0.00
45	B_TIME_RAIL_PT	-1.51	0.220	-6.88	0.00
46	B_TIME_WALKING	-5.97	0.383	-15.58	0.00
47	B_TRAFFICVARIABILITY_DRIVING	-2.56	0.0846	-30.25	0.00
48	B_VEHICLEOWNERSHIP_1_DRIVING	2.17	0.0453	47.93	0.00
49	B_VEHICLEOWNERSHIP_1_PT	-0.413	0.0380	-10.86	0.00
50	B_VEHICLEOWNERSHIP_2_DRIVING	2.57	0.0509	50.57	0.00
51	B_VEHICLEOWNERSHIP_2_PT	-0.615	0.0485	-12.67	0.00
52	B_VEHICLEOWNERSHIP_CYCLING	-0.138	0.0657	-2.10	0.04
53	B_WINTER_CYCLING	-0.329	0.0817	-4.02	0.00
54	B_WINTER_DRIVING	0.123	0.0315	3.91	0.00

B.2 Nested models

Table B.4 Parameter estimates for NL model - flexible modes.

Parameter number	Description	Coeff. estimate	Robust Asympt. std. error	*t*-stat	*p*-value
1	ASC_CYCLING	-4.58	0.112	-40.78	0.00
2	ASC_DRIVING	-3.85	0.0787	-48.90	0.00
3	ASC_PT	-2.29	0.0863	-26.52	0.00
4	B_AGE_CHILD_DRIVING	0.737	0.0501	14.73	0.00
5	B_AGE_CHILD_PT	0.173	0.0435	3.99	0.00
6	B_AGE_PENSIONER_CYCLING	-0.569	0.130	-4.37	0.00
7	B_AGE_PENSIONER_DRIVING	0.429	0.0466	9.21	0.00
8	B_AGE_PENSIONER_PT	0.648	0.0452	14.36	0.00
9	B_COST_DRIVE	-0.120	0.00619	-19.36	0.00
10	B_COST_PT	-0.0998	0.0138	-7.24	0.00
11	B_DAY_SAT_CYCLING	-0.322	0.0976	-3.30	0.00
12	B_DAY_SAT_PT	0.188	0.0434	4.34	0.00
13	B_DAY_WEEK_DRIVING	-0.189	0.0338	-5.58	0.00
14	B_DAY_WEEK_PT	0.295	0.0400	7.37	0.00
15	B_DEPARTURE_INTERPEAK_CYCLING	-0.216	0.0746	-2.90	0.00
16	B_DEPARTURE_INTERPEAK_DRIVING	-0.128	0.0332	-3.87	0.00
17	B_DEPARTURE_PMPEAK_CYCLING	0.310	0.0678	4.58	0.00
18	B_DEPARTURE_PMPEAK_DRIVING	0.394	0.0336	11.74	0.00
19	B_DEPARTURE_PMPEAK_PT	0.109	0.0302	3.60	0.00
20	B_DISTANCE_CYCLING	0.222	0.0875	2.53	0.01
21	B_DISTANCE_DRIVING	0.472	0.0758	6.22	0.00
22	B_DISTANCE_PT	0.471	0.0758	6.21	0.00
23	B_DRIVINGLICENCE_CYCLING	0.703	0.0679	10.36	0.00
24	B_DRIVINGLICENCE_DRIVING	1.07	0.0405	26.34	0.00
25	B_DRIVINGLICENCE_PT	-0.264	0.0322	-8.21	0.00
26	B_FEMALE_CYCLING	-0.834	0.0616	-13.55	0.00
27	B_FEMALE_DRIVING	0.167	0.0286	5.83	0.00
28	B_FEMALE_PT	0.180	0.0259	6.92	0.00
29	B_PURPOSE_B_CYCLING	1.00	0.124	8.12	0.00
30	B_PURPOSE_B_DRIVING	0.309	0.0746	4.14	0.00
31	B_PURPOSE_B_PT	0.593	0.0761	7.79	0.00
32	B_PURPOSE_HBE_DRIVING	-0.570	0.0450	-12.68	0.00
33	B_PURPOSE_HBE_PT	0.311	0.0447	6.97	0.00
34	B_PURPOSE_HBO_CYCLING	0.405	0.0799	5.07	0.00
35	B_PURPOSE_HBO_PT	0.224	0.0321	6.99	0.00
36	B_PURPOSE_HBW_CYCLING	0.761	0.0948	8.03	0.00
37	B_PURPOSE_HBW_DRIVING	-0.700	0.0541	-12.95	0.00
38	B_PURPOSE_HBW_PT	0.189	0.0556	3.40	0.00
39	B_TIME_CYCLING	-2.17	0.606	-3.58	0.00
40	B_TIME_DRIVING	-4.06	0.195	-20.86	0.00
41	B_TIME_ACCESS_PT	-4.60	0.148	-31.15	0.00
42	B_TIME_BUS_PT	-2.13	0.113	-18.77	0.00
43	B_TIME_INTERCHANGEWAIT_PT	-5.06	0.305	-16.60	0.00
44	B_TIME_INTERCHANGEWALK_PT	-2.67	0.982	-2.72	0.01
45	B_TIME_RAIL_PT	-1.49	0.211	-7.06	0.00
46	B_TIME_WALKING	-5.16	0.320	-16.16	0.00
47	B_TRAFFICVARIABILITY_DRIVING	-2.45	0.0824	-29.72	0.00
48	B_VEHICLEOWNERSHIP_1_DRIVING	2.18	0.0412	52.92	0.00
49	B_VEHICLEOWNERSHIP_1_PT	-0.377	0.0299	-12.57	0.00
50	B_VEHICLEOWNERSHIP_2_DRIVING	2.59	0.0463	55.98	0.00
51	B_VEHICLEOWNERSHIP_2_PT	-0.558	0.0399	-13.97	0.00
52	B_VEHICLEOWNERSHIP_CYCLING	-0.136	0.0628	-2.17	0.03
53	B_WINTER_CYCLING	-0.329	0.0813	-4.05	0.00
54	B_WINTER_DRIVING	0.117	0.0308	3.80	0.00
55	MU_FLEXIBLE	1.40	0.0432	9.26*	0.00

Table B.5 Parameter estimates for NL model - powered modes.

Parameter number	Description	Coeff. estimate	Robust Asympt. std. error	t-stat	p-value
1	ASC_CYCLING	-4.97	0.112	-44.18	0.00
2	ASC_DRIVING	-3.87	0.0959	-40.35	0.00
3	ASC_PT	-3.00	0.0744	-40.30	0.00
4	B_AGE_CHILD_DRIVING	0.685	0.0532	12.88	0.00
5	B_AGE_CHILD_PT	0.269	0.0504	5.34	0.00
6	B_AGE_PENSIONER_CYCLING	-0.446	0.132	-3.37	0.00
7	B_AGE_PENSIONER_DRIVING	0.599	0.0500	11.98	0.00
8	B_AGE_PENSIONER_PT	0.826	0.0509	16.25	0.00
9	B_COST_DRIVE	-0.0920	0.00635	-14.49	0.00
10	B_COST_PT	-0.0730	0.0124	-5.91	0.00
11	B_DAY_SAT_CYCLING	-0.345	0.0979	-3.53	0.00
12	B_DAY_SAT_PT	0.159	0.0412	3.86	0.00
13	B_DAY_WEEK_DRIVING	-0.137	0.0356	-3.85	0.00
14	B_DAY_WEEK_PT	0.270	0.0442	6.12	0.00
15	B_DEPARTURE_INTERPEAK_CYCLING	-0.211	0.0741	-2.84	0.00
16	B_DEPARTURE_INTERPEAK_DRIVING	-0.102	0.0283	-3.61	0.00
17	B_DEPARTURE_PMPEAK_CYCLING	0.349	0.0698	5.00	0.00
18	B_DEPARTURE_PMPEAK_DRIVING	0.401	0.0364	11.02	0.00
19	B_DEPARTURE_PMPEAK_PT	0.189	0.0362	5.21	0.00
20	B_DISTANCE_CYCLING	0.402	0.105	3.82	0.00
21	B_DISTANCE_DRIVING	0.646	0.0962	6.72	0.00
22	B_DISTANCE_PT	0.653	0.0963	6.78	0.00
23	B_DRIVINGLICENCE_CYCLING	0.660	0.0657	10.04	0.00
24	B_DRIVINGLICENCE_DRIVING	0.887	0.0497	17.85	0.00
25	B_DRIVING_LICENCE_PT	-0.185	0.0428	-4.32	0.00
26	B_FEMALE_CYCLING	-0.808	0.0633	-12.77	0.00
27	B_FEMALE_DRIVING	0.195	0.0310	6.27	0.00
28	B_FEMALE_PT	0.217	0.0312	6.95	0.00
29	B_PURPOSE_B_CYCLING	1.07	0.130	8.24	0.00
30	B_PURPOSE_B_DRIVING	0.389	0.0837	4.64	0.00
31	B_PURPOSE_B_PT	0.688	0.0884	7.78	0.00
32	B_PURPOSE_HBE_DRIVING	-0.486	0.0482	-10.09	0.00
33	B_PURPOSE_HBE_PT	0.284	0.0523	5.43	0.00
34	B_PURPOSE_HBO_CYCLING	0.376	0.0798	4.71	0.00
35	B_PURPOSE_HBO_PT	0.225	0.0318	7.08	0.00
36	B_PURPOSE_HBW_CYCLING	0.767	0.0997	7.69	0.00
37	B_PURPOSE_HBW_DRIVING	-0.616	0.0631	-9.77	0.00
38	B_PURPOSE_HBW_PT	0.168	0.0668	2.52	0.01
39	B_TIME_CYCLING	-1.75	0.601	-2.90	0.00
40	B_TIME_DRIVING	-3.32	0.217	-15.27	0.00
41	B_TIME_ACCESS_PT	-3.52	0.192	-18.34	0.00
42	B_TIME_BUS_PT	-1.56	0.109	-14.37	0.00
43	B_TIME_INTERCHANGEWAIT_PT	-3.93	0.302	-13.01	0.00
44	B_TIME_INTERCHANGEWALK_PT	-2.39	0.810	-2.95	0.00
45	B_TIME_RAIL_PT	-1.28	0.180	-7.12	0.00
46	B_TIME_WALKING	-5.67	0.380	-14.93	0.00
47	B_TRAFFICVARIABILITY_DRIVING	-2.20	0.0911	-24.13	0.00
48	B_VEHICLEOWNERSHIP_1_DRIVING	1.87	0.0619	30.17	0.00
49	B_VEHICLEOWNERSHIP_1_PT	-0.151	0.0474	-3.18	0.00
50	B_VEHICLEOWNERSHIP_2_DRIVING	2.26	0.0675	33.46	0.00
51	B_VEHICLEOWNERSHIP_2_PT	-0.236	0.0661	-3.58	0.00
52	B_WINTER_CYCLING	-0.332	0.0809	-4.10	0.00
53	B_WINTER_DRIVING	0.101	0.0264	3.81	0.00
54	MU_POWERED	1.29	0.0526	5.51*	0.00

B.3 Hybrid approach

Table B.6 Parameter estimates for full hybrid model.

Parameter number	Description	Coeff. estimate	Robust Asympt. std. error	t-stat	p-value
1	ASC_CYCLING	-13.5	0.662	-20.37	0.00
2	ASC_DRIVING_DL0	-16.9	0.626	-26.92	0.00
3	ASC_DRIVING_DL1	-14.0	0.515	-27.12	0.00
4	ASC_PT_CO	-23.3	0.581	-40.10	0.00
5	ASC_PT_NVO	-19.4	0.580	-33.42	0.00
6	B_AGE_CHILD_CYCLING_NVO_DL0	-1.21	0.365	-3.32	0.00
7	B_AGE_CHILD_DRIVING_NVO_DL0	0.706	0.129	5.48	0.00
8	B_AGE_CHILD_DRIVING_VO1_DL0	1.70	0.0947	17.91	0.00
9	B_AGE_CHILD_DRIVING_VO2_DL0	2.18	0.0918	23.79	0.00
10	B_AGE_CHILD_PT_VO1_DL0	-0.344	0.107	-3.21	0.00
11	B_AGE_MATUREADULT_CYCLING_DL1	1.10	0.104	10.53	0.00
12	B_AGE_MATUREADULT_DRIVING_NVO_DL0	-1.75	0.122	-14.29	0.00
13	B_AGE_MATUREADULT_DRIVING_NVO_DL1	-0.722	0.210	-3.44	0.00
14	B_AGE_MATUREADULT_DRIVING_VO2_DL1	0.520	0.0601	8.65	0.00
15	B_AGE_MATUREADULT_PT_NVO	-0.918	0.0938	-9.80	0.00
16	B_AGE_MATUREADULT_PT_VO1_DL1	-0.374	0.0583	-6.42	0.00
17	B_AGE_YOUNGADULT_CYCLING_DL0	0.452	0.107	4.24	0.00
18	B_AGE_YOUNGADULT_CYCLING_DL1	-0.363	0.0817	-4.45	0.00
19	B_AGE_YOUNGADULT_DRIVING_NVO2_DL1	-0.137	0.0624	-2.19	0.03
20	B_AGE_YOUNGADULT_DRIVING_VO1_DL0	-0.287	0.0642	-4.47	0.00
21	B_AGE_YOUNGADULT_DRIVING_VO1_DL1	-0.534	0.0509	-10.48	0.00
22	B_AGE_YOUNGADULT_PT_NVO_DL0	-0.203	0.0599	-3.40	0.00
23	B_AGE_YOUNGADULT_PT_NVO_DL1	-0.364	0.0781	-4.67	0.00
24	B_COST_CONCHARGE_CO	-0.118	0.00641	-18.35	0.00
25	B_COST_FUEL_NVO_DL1	0.851	0.158	5.39	0.00
26	B_COST_FUEL_VO2_DL1	-0.268	0.105	-2.56	0.01
27	B_COST_PT_NVO	-0.271	0.0266	-10.20	0.00
28	B_DAY_SAT_CYCLING_NVO_DL1	-0.816	0.224	-3.65	0.00
29	B_DAY_SAT_PT_CO	0.313	0.0655	4.77	0.00
30	B_DAY_WEEK_DRIVING_NVO1_DL0	-0.396	0.0519	-7.62	0.00
31	B_DAY_WEEK_DRIVING_NVO1_DL1	-0.192	0.0454	-4.21	0.00
32	B_DAY_WEEK_PT_CO	0.510	0.0566	9.01	0.00
33	B_DEPARTURE_INTERPEAK_DRIVING_VO2_DL0	-0.264	0.0872	-3.03	0.00
34	B_DEPARTURE_INTERPEAK_DRIVING_VO2_DL1	-0.124	0.0550	-2.25	0.02
35	B_DEPARTURE_PMPEAK_CYCLING_NVO1	0.288	0.0650	4.43	0.00
36	B_DEPARTURE_PMPEAK_DRIVING_DL0	0.809	0.0513	15.75	0.00
37	B_DEPARTURE_PMPEAK_DRIVING_DL1	0.360	0.0417	8.62	0.00
38	B_DEPARTURE_PMPEAK_PT_CO	0.257	0.0460	5.60	0.00
39	B_DISTANCE_CYCLING_CO_DL0	-0.896	0.139	-6.44	0.00
40	B_DISTANCE_CYCLING_DL1	-1.47	0.142	-10.33	0.00
41	B_DISTANCE_CYCLING_NVO_DL0	-0.819	0.136	-6.03	0.00
42	B_DISTANCE_DRIVING_CO_DL0	-0.671	0.139	-4.84	0.00
43	B_DISTANCE_DRIVING_DL1	-1.27	0.141	-9.01	0.00
44	B_DISTANCE_DRIVING_NVO_DL0	-0.517	0.137	-3.77	0.00
45	B_DISTANCE_PT_CO_DL0	-0.751	0.138	-5.45	0.00

Continued on next page

Table B.6 – continued from previous page

Parameter number	Description	Coeff. estimate	Robust Asympt. std. error	t-stat	p-value
46	B_DISTANCE_PT_DL1	-1.34	0.141	-9.46	0.00
47	B_DISTANCE_PT_NVO_DL0	-0.658	0.135	-4.88	0.00
48	B_FEMALE_CYCLING	-0.853	0.0593	-14.38	0.00
49	B_FEMALE_DRIVING_DL0	0.458	0.0541	8.46	0.00
50	B_FEMALE_PT_DL0	0.261	0.0466	5.60	0.00
51	B_FEMALE_PT_DL1	0.140	0.0335	4.17	0.00
52	B_LOGDISTANCE_CYCLING	1.44	0.0995	14.45	0.00
53	B_LOGDISTANCE_DRIVING_CO_DL0	2.33	0.0961	24.26	0.00
54	B_LOGDISTANCE_DRIVING_CO_DL1	2.21	0.0818	26.99	0.00
55	B_LOGDISTANCE_DRIVING_NVO_DL0	2.33	0.0977	23.87	0.00
56	B_LOGDISTANCE_DRIVING_NVO_DL1	1.86	0.0863	21.59	0.00
57	B_LOGDISTANCE_PT_CO	3.12	0.0893	34.91	0.00
58	B_LOGDISTANCE_PT_NVO	2.88	0.0909	31.71	0.00
59	B_PURPOSE_B_CYCLING	0.871	0.101	8.65	0.00
60	B_PURPOSE_B_DRIVING_VO2	0.729	0.109	6.69	0.00
61	B_PURPOSE_B_PT_CO_DL0	0.995	0.144	6.89	0.00
62	B_PURPOSE_B_PT_CO_DL1	0.523	0.0783	6.68	0.00
63	B_PURPOSE_HBE_DRIVING_CO_DL0	-0.737	0.0696	-10.58	0.00
64	B_PURPOSE_HBE_DRIVING_NVO	-1.23	0.166	-7.41	0.00
65	B_PURPOSE_HBE_DRIVING_VO1_DL1	-0.299	0.0900	-3.32	0.00
66	B_PURPOSE_HBE_PT_DL0	0.355	0.0594	5.97	0.00
67	B_PURPOSE_HBE_PT_VO2_DL1	-0.492	0.174	-2.83	0.00
68	B_PURPOSE_HBO_CYCLING_VO2	0.611	0.0970	6.30	0.00
69	B_PURPOSE_HBO_DRIVING_CO_DL1	-0.401	0.0419	-9.56	0.00
70	B_PURPOSE_HBO_PT_NVO_DL0	0.196	0.0578	3.38	0.00
71	B_PURPOSE_HBW_CYCLING_NVO1	0.723	0.0728	9.94	0.00
72	B_PURPOSE_HBW_CYCLING_VO2_DL1	1.55	0.144	10.72	0.00
73	B_PURPOSE_HBW_DRIVING_NVO	-1.60	0.145	-11.01	0.00
74	B_PURPOSE_HBW_DRIVING_VO1_DL0	-1.32	0.114	-11.53	0.00
75	B_PURPOSE_HBW_DRIVING_VO1_DL1	-0.842	0.0612	-13.76	0.00
76	B_PURPOSE_HBW_PT_VO2_DL0	1.33	0.322	4.14	0.00
77	B_PURPOSE_HBW_PT_VO2_DL1	0.315	0.0858	3.67	0.00
78	B_TIME_ACCESS_PT_CO	-5.57	0.197	-28.25	0.00
79	B_TIME_ACCESS_PT_NVO_DL0	-6.56	0.359	-18.26	0.00
80	B_TIME_ACCESS_PT_NVO_DL1	-6.55	0.449	-14.57	0.00
81	B_TIME_BUS_PT_NVO1_DL0	-2.12	0.160	-13.25	0.00
82	B_TIME_BUS_PT_NVO1_DL1	-2.94	0.147	-19.97	0.00
83	B_TIME_BUS_PT_VO2_DL0	-2.80	0.283	-9.89	0.00
84	B_TIME_BUS_PT_VO2_DL1	-4.00	0.203	-19.67	0.00
85	B_TIME_DRIVING_NVO	-5.37	0.504	-10.66	0.00
86	B_TIME_DRIVING_VO1_DL0	-3.86	0.431	-8.96	0.00
87	B_TIME_DRIVING_VO1_DL1	-4.10	0.262	-15.63	0.00
88	B_TIME_DRIVING_VO2_DL0	-3.98	0.494	-8.05	0.00
89	B_TIME_DRIVING_VO2_DL1	-2.63	0.353	-7.45	0.00
90	B_TIME_INTERCHANGEWAIT_PT_CO	-6.36	0.336	-18.92	0.00
91	B_TIME_INTERCHANGEWAIT_PT_NVO	-4.06	0.618	-6.57	0.00
92	B_TIME_INTERCHANGEWALK_PT_VO1	-4.69	1.25	-3.77	0.00
93	B_TIME_RAIL	-1.86	0.209	-8.87	0.00

Continued on next page

Table B.6 – continued from previous page

Parameter number	Description	Coeff. estimate	Robust Asympt. std. error	*t*-stat	*p*-value
94	B_TIME_WALKING_DL0	-4.99	0.525	-9.50	0.00
95	B_TIME_WALKING_DL1	-7.14	0.567	-12.60	0.00
96	B_TRAFFICVARIABILITY_DL0	-1.63	0.151	-10.78	0.00
97	B_TRAFFICVARIABILITY_NVO1_DL1	-2.36	0.125	-18.94	0.00
98	B_TRAFFICVARIABILITY_VO2_DL1	-3.24	0.156	-20.75	0.00
99	B_WINTER_CYCLING_CO_DL1	-0.417	0.121	-3.44	0.00
100	B_WINTER_DRIVING_CO	0.163	0.0355	4.59	0.00

References

Akaike, H. (Dec. 1974). "A New Look at the Statistical Model Identification". In: *IEEE Transactions on Automatic Control* 19.6, pp. 716–723.

Andrade, Katia, Kenetsu Uchida, and Seiichi Kagaya (2006). "Development of Transport Mode Choice Model by Using Adaptive Neuro-Fuzzy Inference System". In: *Transportation Research Record* 1977, pp. 8–16.

Assi, Khaled J. et al. (2018). "Mode Choice Behavior of High School Goers: Evaluating Logistic Regression and MLP Neural Networks". In: *Case Studies on Transport Policy* 6.2, pp. 225–230.

Barff, Richard, David Mackay, and Richard W. Olshavsky (Mar. 1, 1982). "A Selective Review of Travel-Mode Choice Models". In: *Journal of Consumer Research* 8.4, pp. 370–380.

Ben-Akiva, Moshe E. and Steven R. Lerman (1985). *Discrete Choice Analysis: Theory and Application to Travel Demand*. MIT Press. 424 pp.

Bergstra, J., D. Yamins, and D. D. Cox (2013). "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. Atlanta, GA, USA: JMLR.org, pp. I-115–I-123.

Bergstra, James S., Rémi Bardenet, et al. (2011). "Algorithms for Hyper-Parameter Optimization". In: *Advances in Neural Information Processing Systems*, pp. 2546–2554.

Bergstra, James and Yoshua Bengio (2012). "Random Search for Hyper-Parameter Optimization". In: *Journal of Machine Learning Research* 13 (Feb), pp. 281–305.

Bergstra, James, Brent Komer, et al. (2014). "Preliminary Evaluation of Hyperopt Algorithms on HPOLib". In: *ICML Workshop on AutoML*.

Bhat, C. R., N. Eluru, and R. B. Copperman (2008). "Flexible Model Structures for Discrete Choice Analysis". In: *Handbook of Transport Modelling,* 5, pp. 75–104.

Biagioni, James P. et al. (2009). "Tour-Based Mode Choice Modeling: Using an Ensemble of Conditional and Unconditional Data Mining Classifiers". In: *Transportation Research Board 88th Annual Meeting*. Vol. 312. Washington DC, USA: Transportation Research Board, pp. 1–15.

Bierlaire, Michel (2016). *PythonBiogeme: A Short Introduction*. TRANSP-OR 160706. Switzerland: Transport and Mobility Laboratory, School of Architecture, Civil and Environmental Engineering, Ecole Polytechnique Fédérale de Lausanne, p. 19.

— (2018). *Biogeme Examples - Swissmetro*. URL: http://biogeme.epfl.ch/examples_swissmetro.html (visited on 08/26/2018).

Bierlaire, Michel, Kay Axhausen, and Georg Abay (2001). "The Acceptance of Modal Innovation: The Case of Swissmetro". In: *Swiss Transport Research Conference*.

Bottou, Léon and Chih-Jen Lin (2007). "Support Vector Machine Solvers". In: *Large scale kernel machines* 3.1, pp. 301–320.

Breiman, Leo (Oct. 1, 2001). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32.
— (Oct. 19, 2017). *Classification and Regression Trees*. Routledge.
Brown, Iain and Christophe Mues (Feb. 15, 2012). "An Experimental Comparison of Classification Algorithms for Imbalanced Credit Scoring Data Sets". In: *Expert Systems with Applications* 39.3, pp. 3446–3453.
Bureau of Transportation Statistics (2018). *The 1995 American Travel Survey (ATS) - Household Trip Characteristics*. URL: https://catalog.data.gov/dataset/the-1995-american-travel-survey-ats-household-trips (visited on 08/26/2018).
Cantarella, Giulio Erberto and Stefano de Luca (2003). "Modeling Transportation Mode Choice through Artificial Neural Networks". In: *Fourth International Symposium on Uncertainty Modeling and Analysis, 2003. ISUMA 2003.* College Park, MD, USA: IEEE, pp. 84–90.
— (2005). "Multilayer Feedforward Networks for Transportation Mode Choice Analysis: An Analysis and a Comparison with Random Utility Models". In: *Transportation Research Part C: Emerging Technologies*. Handling Uncertainty in the Analysis of Traffic and Transportation Systems (Bari, Italy, June 10–13 2002) 13.2, pp. 121–155.
Caruana, Rich and Alexandru Niculescu-Mizil (2006). "An Empirical Comparison of Supervised Learning Algorithms". In: *Proceedings of the 23rd International Conference on Machine Learning*. ACM, pp. 161–168.
Chalumuri, Ravi Sekhar et al. (2009). "Applications of Neural Networks in Mode Choice Modelling for Second Order Metropolitan Cities of India". In: *Proceedings of the Eastern Asia Society for Transportation Studies*. Vol. 7. Surabaya, Indonesia: Eastern Asia Society for Transportation Studies, pp. 1–16.
Chang, Chih-Chung and Chih-Jen Lin (2011). "LIBSVM: A Library for Support Vector Machines". In: *ACM Transactions on Intelligent Systems and Technology* 2.3, 27:1–27:27.
Chapelle, Olivier and Yi Chang (2011). "Yahoo! Learning to Rank Challenge Overview". In: *Proceedings of the Learning to Rank Challenge*, pp. 1–24.
Chen, Tianqi and Carlos Guestrin (2016). "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 785–794.
Cheng, Long et al. (2014). "Modeling Mode Choice Behavior Incorporating Household and Individual Sociodemographics and Travel Attributes Based on Rough Sets Theory". In: *Computational Intelligence and Neuroscience* 2014, pp. 1–9.
Chicago Metropolitan Agency for Planning (2018a). *CATS Household Travel Survey, 1990*. URL: https://datahub.cmap.illinois.gov/dataset/travel-survey-1990 (visited on 08/26/2018).
— (2018b). *Travel Tracker Survey, 2007 - 2008: Public Data*. URL: https://datahub.cmap.illinois.gov/dataset/traveltracker0708 (visited on 08/26/2018).
Chollet, François et al. (2015). "Keras". In:
Cortes, Corinna and Vladimir Vapnik (Sept. 1, 1995). "Support-Vector Networks". In: *Machine Learning* 20.3, pp. 273–297.
Croissant, Yves (Dec. 16, 2016). *Ecdat: Data Sets for Econometrics*. URL: https://CRAN.R-project.org/package=Ecdat (visited on 08/26/2018).
Defazio, Aaron, Francis Bach, and Simon Lacoste-Julien (July 1, 2014). "SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives". In:
Delaware Valley Regional Planning Commission (2018). *2012 Household Travel Survey*. URL: https://www.dvrpc.org/transportation/Modeling/Data/ (visited on 08/26/2018).

Dell'Orco, Mauro and Michele Ottomanelli (2012). "Simulation of Users Decision in Transport Mode Choice Using Neuro-Fuzzy Approach". In: *International Conference on Computational Science and Its Applications (ICCSA 2012)*. Salvador de Bahia, Brazil: Springer, pp. 44–53.

Department for Transport (2014). *Transport Analysis Guidance (TAG) Unit A1. 3 - User and Provider Impacts*.

Edara, Praveen Kumar, Dušan Teodorović, and Hojong Baik (2007). "Using Neural Networks to Model Intercity Mode Choice". In: *Smart Systems Engineering: Computational Intelligence in Architecting Complex Engineering Systems*. Vol. 17. Artificial Neural Networks in Engineering Conference (ANNIE 2007). St Louis, Missouri, USA: ASME Press, pp. 143–148.

Efron, B. (Jan. 1979). "Bootstrap Methods: Another Look at the Jackknife". In: *The Annals of Statistics* 7.1, pp. 1–26.

Efron, Bradley (1983). "Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation". In: *Journal of the American Statistical Association* 78.382, pp. 316–331.

Efron, Bradley and Robert Tibshirani (June 1, 1997). "Improvements on Cross-Validation: The 632+ Bootstrap Method". In: *Journal of the American Statistical Association* 92.438, pp. 548–560.

Ermagun, Alireza, Taha Hossein Rashidi, and Zahra Ansari Lari (2015). "Mode Choice for School Trips: Long-Term Planning and Impact of Modal Specification on Policy Assessments". In: *Transportation Research Record* 2513, pp. 97–105.

Errampalli, Madhu, Masashi Okushima, and Takamasa Akiyama (2007). "Combined Fuzzy Logic Based Mode Choice and Microscopic Simulation Model for Transport Policy Evaluation". In: *11th World Conference on Transport Research*. Berkley CA, USA: Transportation Research Board.

Federal Highway Administration (2018). *National Household Travel Survey*. URL: https://nhts.ornl.gov/ (visited on 08/26/2018).

Friedman, Jerome H. (2001). "Greedy Function Approximation: A Gradient Boosting Machine". In: *The Annals of Statistics* 29.5, pp. 1189–1232.

Gao, Jian et al. (2013). "Impact of Transit Network Layout on Resident Mode Choice". In: *Mathematical Problems in Engineering* 2013, pp. 1–8.

Gazder, Uneb and Nedal T. Ratrout (2015). "A New Logit-Artificial Neural Network Ensemble for Mode Choice Modeling: A Case Study for Border Transport". In: *Journal of Advanced Transportation* 49.8, pp. 855–866.

Geurts, Pierre, Damien Ernst, and Louis Wehenkel (Apr. 2006). "Extremely Randomized Trees". In: *Machine Learning* 63.1, pp. 3–42.

Gneiting, Tilmann and Adrian E Raftery (Mar. 2007). "Strictly Proper Scoring Rules, Prediction, and Estimation". In: *Journal of the American Statistical Association* 102.477, pp. 359–378.

Golshani, Nima et al. (2018). "Modeling Travel Mode and Timing Decisions: Comparison of Artificial Neural Networks and Copula-Based Joint Model". In: *Travel Behaviour and Society* 10, pp. 21–32.

Google (2018). *About Waze*. URL: https://support.google.com/waze/answer/6071177?hl=en&ref_topic=6262616&vid=1-635759896114079353-1109811941&rd=1 (visited on 08/24/2018).

Google Developers (2018). *Map Coverage Details*. URL: https://developers.google.com/maps/coverage (visited on 08/23/2018).

Greene, William H. (Nov. 21, 2011). *Econometric Analysis*. Pearson Higher Ed. 1230 pp.

Hagenauer, Julian and Marco Helbich (2017). "A Comparative Study of Machine Learning Classifiers for Modeling Travel Mode Choice". In: *Expert Systems with Applications* 78, pp. 273–282.

Harrell, Frank E., Kerry L. Lee, and Daniel B. Mark (1996). "Mutlivariable Prognostic Models: Issues in Developing Models, Evaluating Assumptions and Adequacy, and Measuring and Reducing Errors". In: *Statistics in Medicine* 15.4, pp. 361–387.

Hartigan, J. A. and M. A. Wong (1979). "Algorithm AS 136: A K-Means Clustering Algorithm". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28.1, pp. 100–108.

Hastie, Trevor, Jerome Friedman, and Robert Tibshirani (2008). *The Elements of Statistical Learning*. 2nd ed. Springer Series in Statistics. New York: Springer. 745 pp.

Hensher, David A. and Lester W. Johnson (Aug. 1, 1983). "Alternative Modelling Procedures in Studies of Travel Mode Choice: A Review and Appraisal". In: *Transportation Planning and Technology* 8.3, pp. 203–216.

Hensher, David A. and Tu T. Ton (2000). "A Comparison of the Predictive Potential of Artificial Neural Networks and Nested Logit Models for Commuter Mode Choice". In: *Transportation Research Part E: Logistics and Transportation Review* 36.3, pp. 155–172.

Hillel, T., P. Guthrie, et al. (2016). "Assessing the Discrepancies between Recorded and Commonly Assumed Journey Times in London". In: *Transforming the Future of Infrastructure through Smarter Information: Proceedings of the International Conference on Smart Infrastructure and Construction (ICSIC 2016)*. Cambridge Centre for Smart Infrastructure & Construction. ICE Publishing, pp. 759–764.

Hillel, Tim, Michel Bierlaire, et al. (2018a). "A New Framework for Assessing Classification Algorithms for Mode Choice Prediction". In: 7th Symposium of the European Association for Research in Transportation (hEART 2018). Athens, Greece.

— (2018b). "Validation of Probabilistic Classifiers". In: Swiss Transport Research Conference (STRC 2018). Monte Verità, Ascona, Switzerland.

Hillel, Tim, Mohammed Z E B Elshafie, and Ying Jin (2018). "Recreating Passenger Mode Choice-Sets for Transport Simulation: A Case Study of London, UK". In: *Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction* 171.1, pp. 29–42.

Hoos, Holger et al. (2014). "An Efficient Approach for Assessing Hyperparameter Importance". In: *International Conference on Machine Learning*, pp. 754–762.

Hornik, Kurt (Jan. 1, 1991). "Approximation Capabilities of Multilayer Feedforward Networks". In: *Neural Networks* 4.2, pp. 251–257.

Hossein Rashidi, Taha and Hironobu Hasegawa (2014). "An Innovative Simultaneous System of Disaggregate Models for Trip Generation, Mode, and Destination Choice". In: *Transportation Research Board 93rd Annual Meeting*. Washington DC, USA: Transportation Research Board.

Hussain, H. D. et al. (2017). "Analysis of Transportation Mode Choice Using a Comparison of Artificial Neural Network and Multinomial Logit Models". In: *ARPN Journal of Engineering and Applied Sciences* 12.5, pp. 1483–1493.

Jia, Hongfei, Xiongjiu Cao, and Kaihua Yang (2015). "Residents' Travel Mode Choice Model". In: *Traffic Engineering & Control* 56.1, pp. 169–174.

Jin, Ying, I. N. Williams, and M. Shahkarami (Sept. 2002). "A New Land Use and Transport Interaction Model for London and Its Surrounding Regions". In: *Proceedings of the AET European Transport Conference*.

Jing, Peng et al. (Apr. 14, 2018). "Travel Mode and Travel Route Choice Behavior Based on Random Regret Minimization: A Systematic Review". In: *Sustainability* 10.4, p. 1185.

Juremalani, Jayesh (2017). "Comparison of Different Mode Choice Models for Work Trips Using Data Mining Process". In: *Indian Journal of Science and Technology* 10.17, pp. 1–3.

Karlaftis, Matthew G. (2004). "Predicting Mode Choice through Multivariate Recursive Partitioning". In: *Journal of Transportation Engineering* 130.2, pp. 245–250.

Kedia, Ashu Shivkumar, Krishna Bhuneshwar Saw, and Bhimaji Krishnaji Katti (2015). "Fuzzy Logic Approach in Mode Choice Modelling for Education Trips: A Case Study of Indian Metropolitan City". In: *Transport* 30.3, pp. 286–293.

Kitchenham, Barbara and Stuart Charters (2007). *Guidelines for Performing Systematic Literature Reviews in Software Engineering*. EBSE Technical Report 2007-01. EBSE.

Komer, Brent, James Bergstra, and Chris Eliasmith (2014). "Hyperopt-Sklearn: Automatic Hyperparameter Configuration for Scikit-Learn". In: *Proceedings of the 13th Python in Science Conference*, pp. 34–40.

Kostenko, Boris (Dec. 22, 2018). *XGBoost Feature Interactions Reshaped*. URL: https://github.com/limexp/xgbfir (visited on 12/23/2018).

Kruger, J. (1991). "Review of Research on Urban Area Mode Choice Modelling". In: *13th CAITR Conference, December 12-13, 1991, Cromwell College, The University of Queensland*. Conference of Australian Institutes of Transport Research (CAITR), 13th, 1991, Brisbane, Queensland.

Kumar, Mukesh, Pradip Sarkar, and Errampalli Madhu (2013). "Development of Fuzzy Logic Based Mode Choice Model Considering Various Public Transport Policy Options". In: *International Journal for Traffic and Transport Engineering* 3.4, pp. 408–425.

Lee, Dongwoo, Sybil Derrible, and Francisco Camara Pereira (2018). "Comparison of Four Types of Artificial Neural Networks and a Multinomial Logit Model for Travel Mode Choice Modeling". In: *Transportation Research Board 97th Annual Meeting*. Washington DC, USA: Transportation Research Board.

Li, Juan et al. (2016). "Cluster-Based Logistic Regression Model for Holiday Travel Mode Choice". In: *Procedia Engineering*. Vol. 137. 6th International Conference on Green Intelligent Transportation System and Safety (GITSS 2015). Beijing, China: Elsevier, pp. 729–737.

Liang, LeiLei et al. (2018). "Travel Mode Choice Analysis Based on Household Mobility Survey Data in Milan: Comparison of the Multinomial Logit Model and Random Forest Approach". In: *Transportation Research Board 97th Annual Meeting*. Washington DC, USA: Transportation Research Board.

Lindner, Anabele, Cira Souza Pitombo, and André Luiz Cunha (2017). "Estimating Motorized Travel Mode Choice Using Classifiers: An Application for High-Dimensional Multicollinear Data". In: *Travel Behaviour and Society* 6, pp. 100–109.

Lu, Yandan and Kazuya Kawamura (2010). "Data-Mining Approach to Work Trip Mode Choice Analysis in Chicago, Illinois, Area". In: *Transportation Research Record* 2156, pp. 73–80.

Luxembourg Institute of Socio-Economic Research (2018). *Socio-Economic Panel of Liewen Zu Lëtzebuerg III (PSELL3)*. URL: http://dataservice.liser.lu/en_US/dataservice/db=23 (visited on 08/26/2018).

Ma, Tai-Yu (2015). "Bayesian Networks for Multimodal Mode Choice Behavior Modelling: A Case Study for the Cross Border Workers of Luxembourg". In: *Transportation Research*

Procedia. Vol. 10. 18th Euro Working Group on Transportation, EWGT 2015, 14-16 July 2015, Delft, The Netherlands, pp. 870–880.

Ma, Tai-Yu, Joseph Y. J. Chow, and Jia Xu (2017). "Causal Structure Learning for Travel Mode Choice Using Structural Restrictions and Model Averaging Algorithm". In: *Transportmetrica A: Transport Science* 13.4, pp. 299–325.

Madrigal, A. C. (Sept. 6, 2012). "How Google Builds Its Maps—and What It Means for the Future of Everything". In: *The Atlantic*.

McFadden, Daniel (1981). "Econometric Models of Probabilistic Choice". In: *Structural Analysis of Discrete Data with Econometric Applications*. Ed. by Charles F. Manski and Daniel McFadden. MIT Press, pp. 198–272.

Meixell, Mary J. and Mario Norbis (Aug. 15, 2008). "A Review of the Transportation Mode Choice and Carrier Selection Literature". In: *The International Journal of Logistics Management* 19.2, pp. 183–211.

Metropolitan Transportation Commission (2018a). *1990 Bay Area Travel Surveys*. URL: http://www.surveyarchive.org/ (visited on 08/26/2018).

— (2018b). *San Francisco Bay Area Travel Survey 2000*. URL: http://www.surveyarchive.org/ (visited on 08/26/2018).

Miller, G. (Dec. 8, 2014). *The Huge, Unseen Operation Behind the Accuracy of Google Maps*. URL: http://www.wired.com/2014/12/google-maps-ground-truth/ (visited on 08/24/2015).

Minal and Ch. Ravi Sekhar (Sept. 2014). "Mode Choice Analysis: The Data, the Models and Future Ahead". In: *International Journal for Traffic and Transport Engineering* 4.3, pp. 269–285.

Moher, David et al. (2009). "Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement". In: *PLoS Medicine* 6.7, p. 6.

Moons, Elke, Geert Wets, and Marc Aerts (2007). "Nonlinear Models for Determining Mode Choice". In: *Progress in Artificial Intelligence*. 13th Portuguese Conference on Aritficial Intelligence (EPIA 2007). Lecture Notes in Computer Science. Guimarães, Portugal: Springer, pp. 183–194.

Nam, Daisik et al. (2017). "A Model Based on Deep Learning for Predicting Travel Mode Choice". In: *Transportation Research Board 96th Annual Meeting*. Washington DC, USA: Transportation Research Board, pp. 8–12.

Nerhagen, L. (2000). "Mode Choice Behaviour, Travel Mode Choice Models and Value of Time Estiamtion. A Literature Review". In: *CTEK working paper*.

Niculescu-Mizil, Alexandru and Rich Caruana (2005). "Predicting Good Probabilities with Supervised Learning". In: *Proceedings of the 22nd International Conference on Machine Learning*. ACM, pp. 625–632.

Omrani, Hichem (2015). "Predicting Travel Mode of Individuals by Machine Learning". In: *Transportation Research Procedia*. Vol. 10. 18th Euro Working Group on Transportation, EWGT 2015, 14-16 July 2015, Delft, The Netherlands, pp. 840–849.

Omrani, Hichem et al. (2013). "Prediction of Individual Travel Mode with Evidential Neural Network Model". In: *Transportation Research Record* 2399, pp. 1–8.

Papaioannou, Dimitrios and Luis Miguel Martinez (2015). "The Role of Accessibility and Connectivity in Mode Choice. A Structural Equation Modeling Approach". In: *Transportation Research Procedia*. Vol. 10. 18th Euro Working Group on Transportation, EWGT 2015, 14-16 July 2015, Delft, The Netherlands, pp. 831–839.

Pedregosa, Fabian et al. (2011). "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (Oct), pp. 2825–2830.

Pirra, Miriam and Marco Diana (2017). "Tour-Based Mode Choice Study Through Support Vector Machine Classifiers". In: *Transportation Research Board 96th Annual Meeting*. Washington DC, USA: Transportation Research Board.

Pitombo, Cira Souza et al. (2015). "A Two-Step Method for Mode Choice Estimation with Socioeconomic and Spatial Information". In: *Spatial Statistics* 11, pp. 45–64.

Platt, John C. (1999). "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods". In: *Advances in Large Margin Classifiers*. MIT Press, pp. 61–74.

Pulugurta, Sarada, Ashutosh Arun, and Madhu Errampalli (2013). "Use of Artificial Intelligence for Mode Choice Analysis and Comparison with Traditional Multinomial Logit Model". In: *Procedia - Social and Behavioral Sciences*. Vol. 104. 2nd Conference of Transportation Research Group of India (CTRG 2013). Agra, India: Elsevier, pp. 583–592.

Raju, K A, P K Sikdar, and S L Dhingra (1996). "Modelling Mode Choice by Means of an Artificial Neural Network". In: *Environment and Planning B: Planning and Design* 23.6, pp. 677–683.

Ramanuj, P. S. and P. J. Gundaliya (2013). "Disaggregated Modeling of Mode Choice by ANN-a Case Study of Ahmedabad City in Gujarat State". In: *Journal of the Indian Roads Congress* 74.1, pp. 3–12.

Rasmussen, Carl Edward (2004). "Gaussian Processes in Machine Learning". In: *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*. Ed. by Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 63–71.

Rasouli, Soora and Harry J.P. Timmermans (2014). "Using Ensembles of Decision Trees to Predict Transport Mode Choice Decisions: Effects on Predictive Success and Uncertainty Estimates". In: *European Journal of Transport and Infrastructure Research* 14.4, pp. 412–424.

Ratrout, Nedal T., Uneb Gazder, and Hashim M.N. Al-Madani (Jan. 1, 2014). "A Review of Mode Choice Modelling Techniques for Intra-City and Border Transport". In: *World Review of Intermodal Transportation Research* 5.1, pp. 39–58.

Saeb, Sohrab et al. (May 1, 2017). "The Need to Approximate the Use-Case in Clinical Machine Learning". In: *GigaScience* 6.5, pp. 1–9.

Seetharaman, Padma et al. (2009). "Comparative Evaluation of Mode Choice Modelling by Logit and Fuzzy Logic". In: *Proceedings of the Eastern Asia Society for Transportation Studies*. Vol. 7. Surabaya, Indonesia: Eastern Asia Society for Transportation Studies, pp. 1–16.

Sekhar, Ch. Ravi, Minal, and E. Madhu (2016). "Mode Choice Analysis Using Random Forrest Decision Trees". In: *Transportation Research Procedia*. Vol. 17. 11th Transportation Planning and Implementation Methodologies for Developing Countries, TPMDC 2014, 10-12 December 2014, Mumbai, India, pp. 644–652.

Semanjski, Ivana, Angel Lopez, and Sidharta Gautama (2016). "Forecasting Transport Mode Use with Support Vector Machines Based Approach". In: *Transactions on Maritime Science* 5.2, pp. 111–120.

Shafahi, Yusof and Sobhaan Nazari (2006). "Disaggregate Mode Choice Analysis for Work Trips Using Genetic-Fuzzy and Neuro-Fuzzy Systems." In: *Proceedings of the 10th IASTED International Conference on Artificial Intelligence and Soft Computing (ASC 2006)*. Palma de Mallorca. Spain: ACTA Press, pp. 250–255.

Shmueli, Galit (Aug. 2010). "To Explain or to Predict?" In: *Statistical Science* 25.3, pp. 289–310.

Shukla, Nagesh et al. (2013). "Data-Driven Modeling and Analysis of Household Travel Mode Choice". In: *20th International Congress on Modelling and Simulation (MODSIM 2013)*. Adelaide, Australia: The Modelling and Simulation Society of Australia and New Zealand Inc., pp. 92–98.

Slevin, Roger and Matthew Griffin (2016). *Background Information on NaPTAN*. Department for Transport.

Snoek, Jasper, Hugo Larochelle, and Ryan P. Adams (2012). "Practical Bayesian Optimization of Machine Learning Algorithms". In: *Advances in Neural Information Processing Systems*, pp. 2951–2959.

Srivastava, Nitish et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *The Journal of Machine Learning Research* 15.1, pp. 1929–1958.

Subba Rao, P. V. et al. (1998). "Another Insight into Artificial Neural Networks through Behavioural Analysis of Access Mode Choice". In: *Computers, Environment and Urban Systems* 22.5, pp. 485–496.

Svozil, Daniel, Vladimir Kvasnicka, and Jiri Pospichal (Nov. 1, 1997). "Introduction to Multi-Layer Feed-Forward Neural Networks". In: *Chemometrics and Intelligent Laboratory Systems* 39.1, pp. 43–62.

Tang, Dounan, Min Yang, and Mei Hui Zhang (2012). "Travel Mode Choice Modeling: A Comparison of Bayesian Networks and Neural Networks". In: *Applied Mechanics and Materials* 209-211, pp. 717–723.

Tang, Liang, Chenfeng Xiong, and Lei Zhang (2015). "Decision Tree Method for Modeling Travel Mode Switching in a Dynamic Behavioral Process". In: *Transportation Planning and Technology* 38.8, pp. 833–850.

Train, K. E. (2009). *Discrete Choice Methods with Simulation*. Cambridge university press.

Transport for London (2015). *London Travel Demand Survey (LTDS) - Summary Report*.

— (2016). *Data Feeds*. URL: https://www.tfl.gov.uk/info-for/open-data-users/data-feeds (visited on 09/03/2016).

Transport for NSW (2018). *Household Travel Survey 2005/06 – 2016/17*. URL: https://opendata.transport.nsw.gov.au/dataset/household-travel-survey-200506-%E2%80%93-201617 (visited on 08/26/2018).

Transport for Victoria (2018). *VISTA Data and Publications*. URL: https://transport.vic.gov.au/data-and-research/vista/vista-data-and-publications/ (visited on 08/26/2018).

Van Middelkoop, Manon, Aloys Borgers, and Harry Timmermans (2003). "Inducing Heuristic Principles of Tourist Choice of Travel Mode: A Rule-Based Approach". In: *Journal of Travel Research* 42.1, pp. 75–83.

Varma, Sudhir and Richard Simon (2006). "Bias in Error Estimation When Using Cross-Validation for Model Selection". In: *BMC Bioinformatics*, p. 8.

Wang, Fangru and Catherine L. Ross (2018). "Machine Learning Travel Mode Choices: Comparing the Performance of an Extreme Gradient Boosting Model with a Multinomial Logit Model". In: *Transportation Research Record* Advanced online publication, pp. 1–11.

Wang, Weijie and Moon Namgung (2007). "Knowledge Discovery from the Data of Long Distance Travel Mode Choices Based on Rough Set Theory". In: *International Journal of Multimedia and Ubiquitous Engineering* 2.3, pp. 81–90.

Wu, Ting-Fan, Chih-Jen Lin, and Ruby C. Weng (2004). "Probability Estimates for Multi-Class Classification by Pairwise Coupling". In: *Journal of Machine Learning Research* 5 (Aug), pp. 975–1005.

Xian-Yu, Jian-Chuan (2011). "Travel Mode Choice Analysis Using Support Vector Machines". In: *11th International Conference of Chinese Transportation Professionals (ICCTP 2011)*. Nanjing, China: American Society of Civil Engineers, pp. 360–371.

Xie, Chi, Jinyang Lu, and Emily Parkany (2003). "Work Travel Mode Choice Modeling with Data Mining: Decision Trees and Neural Networks". In: *Transportation Research Record* 1854, pp. 50–61.

Yin, Huanhuan and Hongzhi Guan (2011). "Traffic Mode Choice Model Based on BP Neural Network". In: *Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE)*. Changchun, China: IEEE, pp. 1441–1444.

Zenina, Nadezda and Arkady Borisov (2011). "Transportation Mode Choice Analysis Based on Classification Methods". In: *Scientific Journal of Riga Technical University. Computer Sciences* 45.1, pp. 49–53.

Zhang, Chongsheng, Changchang Liu, et al. (Oct. 1, 2017). "An Up-to-Date Comparison of State-of-the-Art Classification Algorithms". In: *Expert Systems with Applications* 82, pp. 128–150.

Zhang, Yunlong and Yuanchang Xie (2008). "Travel Mode Choice Modeling with Support Vector Machines". In: *Transportation Research Record* 2076, pp. 141–150.

Zhao, Dan et al. (2010). "Travel Mode Choice Modeling Based on Improved Probabilistic Neural Network". In: *Traffic and Transportation Studies 2010 (ICTTS 2010)*. Vol. 383. Kunming, China: ASCE, pp. 685–695.

Zhou, Miaomiao and Jian Lu (2011). "Research on Prediction of Traffic Mode Choice of Urban Residents". In: *11th International Conference of Chinese Transportation Professionals (ICCTP 2011)*. Nanjing, China: American Society of Civil Engineers, pp. 449–460.

Zhu, Zheng et al. (2017). "A Mixed Bayesian Network for Two-Dimensional Decision Modeling of Departure Time and Mode Choice". In: *Transportation* Advanced online publication, pp. 1–24.