Modelling blue-light ambulance mobility in the London metropolitan area

Modelling Blue-Light Ambulance Mobility in the London Metropolitan Area

Marcus J. Poulton

2019

A thesis presented for the degree of PhD Computer Science and Information Systems

Department of Computer Science and Information Systems

Birkbeck, University of London

Acknowledgements

A special thank you goes to my supervisors, Professor George Roussos and Dr David Weston, for keeping me on the right track. Their advice and guidance was invaluable in carrying out the research and producing this document.

This research would have never happened without the significant quantity of data kindly provided by London Ambulance Service (LAS). Particular thanks go to Susannah Money and Leanne Smith of the LAS for their help and support. The determination and dedication of the LAS control room staff and ambulance crew is unparalleled in the field of emergency care. The sheer volume of data that the IM&T team at LAS collect and analyse on a daily basis gives credence to their professionalism. It has also helped me immensely in my analysis of ambulance mobility. I would also like to thank Dr Tassos Noulas for his contributions and enthusiasm.

I dedicate this work to my mother and father.

Abstract

Actions taken immediately following a life-threatening incident are critical for the survival of the patient. In particular, the timely arrival of ambulance crew often makes the difference between life and death. As a consequence, ambulance services are under persistent pressure to achieve rapid emergency response. Meeting stringent performance requirements poses special challenges in metropolitan areas where the higher population density results in high rates of life-threatening incident occurrence, compounded by lower response speeds due to traffic congestion. A key ingredient of data-driven approaches to address these challenges is the effective modelling of ambulance movement thus enabling the accurate prediction of the expected arrival time of a crew at the site of an incident. Ambulance mobility patterns however are distinct and in particular differ from civilian traffic: crews travelling with flashing blue lights and sirens are by law exempt from certain traffic regulations; and moreover, ambulance journeys are triggered by emergency incidents that are generated following distinct spatial and temporal patterns.

We use a large historical dataset of incidents and ambulance location traces to model route selection and arrival times. Working on a road routing network modified to reflect the differences between emergency and regular vehicle traffic, we develop a methodology for matching ambulances Global Positioning System (GPS) coordinates to road segments, allowing the reconstruction of ambulance routes with precise speed data. We demonstrate how a road speed model that exploits this information achieves best predictive performance by implicitly capturing route-specific patterns in changing traffic conditions. We then present a hybrid model that achieves a high route similarity score while minimising journey duration error. This hybrid model outperforms alternative mobility models. To the best of our knowledge, this study represents the first attempt to apply data-driven methodologies to route selection and estimation of arrival times of ambulances travelling with blue lights and sirens.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Summary

A well-established clinical outcome is that shorter ambulance arrival times play a critical role in the case of emergency patients involved in incidents of high severity [1]. For example, in the UK the National Health Service (NHS) Ambulance services treat approximately 30,000 Out-of-Hospital Cardiac Arrest (OHCA) patients every year with a varied resuscitation outcome of between 13%-27%. Survival is low with between 2%-12% of patients reaching hospital discharge [2]. This trend may be changing, however,

as recent research in Japan shows ever improving outcomes for OHCA patients where Emergency Medical Services (EMS) personnel witness the arrest [3].

Across the world EMS performance targets are generally based on time-to-scene of the first responding resource. The UK is no different and has a system performance target to reach at least 75% of its Category A (immediately life threatening) patients within 8 minutes [4]. This target is low in comparison to many other countries where the target is usually 8 minutes for 90% of immediately life threatening cases [5]. Indeed, the Ontario Pre-hospital Advanced Life Support (OPALS) study [6] calls for a maximum 5 minute arrival time to 90% of immediately life threatening cases.

The use of top-down performance targets based on time of arrival leads EMS to employ a tactic of sending the nearest available ambulance to the patient [7]. In this respect, 'nearest' can either be interpreted as a Euclidean distance or a prediction of arrival time based on the road network and vehicle type. Different techniques could be applied to estimating arrival time by taking into consideration of spatio-temporal factors, vehicle type, road types, and average speeds. This is a challenging task as the mobility characteristics of ambulances in their various forms differ from normal civilian traffic, for example, ambulances travelling under Blue Lights and Sirens (BLS) is by law [8] exempt from traffic regulations that would otherwise impede progress to a patient. For example, ambulances responding to a call on BLS are permitted to treat red traffic lights as a give way sign, are able to pass the wrong side of a keep left bollard and disobey the speed limit.

The London Ambulance Service (LAS) recognise that efficiency of dispatch must be improved [9] as the need to have an Emergency Medical Responder (EMR) on scene quickly is clear. However, LAS arrival time performance for the most serious medical emergencies has been a long way off the 75% required of it, being somewhere between 58% and 65% for the first four months of 2016 [10].

Recent trends in ambulance operational research have included event-driven simulation [11, 12, 13] to emulate the urban environment and thus provide a platform

for researching aspects such as novel dispatch algorithms and on strategic or tactical placement of emergency vehicles. Similar tools are in use at LAS for the same purpose. However, results produced by these simulations can be significantly affected by the quality of the ambulance mobility models [14]. Improved models could, therefore, contribute towards the generation of more effective response strategies and tactics.

1.2 Objectives

In this thesis we address the challenge of accurate routing and journey time estimation of an ambulance in response to a medical emergency, using London as our specific case study. By understanding the spatio-temporal dynamics of emergency events and road network characteristics our main goal is to model and predict ambulance mobility with a view to enable considerable improvements to the operational efficiency of the emergency service. To address the challenges of accurate modelling of ambulance response we set the following objectives:

1. To explore and understand blue-light ambulance mobility, using London as our case study.

2. To identify if any spatio-temporal relationships exist in ambulance mobility in London.

3. To develop models that can be used to accurately predict ambulance routes and journey times whilst travelling with blue lights and sirens.

4. To establish the challenges involved and potential benefits in modelling blue-light ambulance mobility.

1.3 Contributions

The following contributions from our research into modelling ambulance mobility in the London Metropolitan area are:

1. **Spatio-temporal analysis of ambulance datasets:** Our analysis of ambulance traces reveals strong temporal and spatial variations in the speed of different types emergency vehicle. Clusters of activity are also present in larger satellite towns.

2. **Creation of the Blue Light Road Network (BLRN):** We create road routing network model specifically for emergency vehicles travelling under BLS conditions in London. The model represents the road network as perceived by the LAS crew, who follow different road regulations under BLS conditions.

3. **Accurate ambulance speed estimation from low frequency GPS data:** We use a data-driven approach to mine large datasets of emergency vehicle telemetry and emergency event data, constructing road speed models that predict travel times for routes previously undertaken by LAS emergency vehicles travelling with BLS in London. Furthermore, our best speed model can generate optimal routes that are quicker, differing from routes currently chosen by their crew.

4. **Development of Blue Light Route Choice Model (BLRCM):** We produce a road speed model, BLRCM, designed to predict routes that emergency vehicles take whilst travelling under BLS in London.

5. **Development of a data-driven predictive ambulance mobility model:** By developing a Hybrid Routing Model (HRM) that combines emergency vehicle route choice and accurate speed estimation we demonstrate a routing model capable of better estimating both route and arrival time for emergency vehicles travelling with blue lights and sirens in London.

1.4 Related Work

We have published two related peer-review publication as follows:

1. M. Poulton and G. Roussos, *Towards Smarter Metropolitan Emergency Response* in 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2013, pp. 25762580. [14]

   This publication was a pre-cursor to our research work presented in this thesis, specifically in the development of accurate speed models used in simulation. Taking the London Ambulance Service as our case study we developed a simulation framework and introduce an enhanced routing and dispatch method that combines concurrent assignment and redeployment of units in a single algorithm. We provide evidence that our unified proactive relocation and dispatch model produces significant improvements in measured performance in terms of meeting citizen needs.

2. M. Poulton, A. Noulas, D. Weston, and G. Roussos, *Modeling Metropolitan-Area Ambulance Mobility Under Blue Light Conditions* IEEE Access, vol. 7, pp. 13901403, 2019. [15]

   This publication provides a summary of the findings of this thesis. We summarised the data collection, processing and analysis of GPS traces and provide descriptions of the journey time estimation models we developed. Working on the London road network graph modified to reflect the differences between emergency and civilian vehicle traffic, we develop a methodology for the precise estimation of expected ambulance speed at the individual road segment level. We demonstrate how a model that exploits this information achieves best predictive performance by implicitly capturing route-specific persistent patterns in changing traffic conditions. We then present a predictive method that achieves a high route similarity score while minimising journey duration error.

1.5 Research Approach

Our main asset was the large volume of historic data kindly supplied by the LAS, so a predominantly quantitative data-driven approach was employed. By characterising and cleansing the data available to us, and then through experimentation, we provide the exploratory precursor prior to formulating our models of emergency vehicle mobility. Although most of the supplied data relates to non-emergencies, we focus specifically on the subset of data that relates to the most serious emergencies, as these events stand to benefit the most from shorter arrival times. Our approach was divided into five distinct phases; data load and cleanse, route reconstruction, speed model generation, model evaluation, and development of a new hybrid routing model.

1.5.1 Data Load and Cleanse

The digital road network is a key component in our research and forms the basis of many of the experiments and models. With the knowledge that emergency vehicles travelling on BLS have different rules of the road to civilian traffic we modify an industry-standard road network to produce a BLRN more suited to our needs. This modified network model was used extensively throughout the research.

Operational data, which included emergency vehicle telemetry, dispatch and emergency event data were loaded, cleansed and initial analysis carried out to characterise their key features. A standard baseline road speed model was formulated by snapping historic GPS data to nearest road links. This method gave us insight into the GPS data quality specific to the LAS and the coverage of the road network.

1.5.2 Route Reconstruction

During the data cleansing process a key discovery was that snapping of GPS observations to their nearest road segment is insufficient to build a suitable road speed model due to the sparse nature of the data. We overcome this limitation by reassembling emergency

vehicle trajectories from raw GPS data in a process known as map-matching.

1.5.3 Speed Model Generation

We introduce five experimental road speed models. One of these models, Metric II, replicates as far as possible the behaviour of the LAS routing engine which proves useful in base-lining the predictive capabilities of the routing engine. We used a quantitative approach to build Metrics $III - V$ in order to capture underlying spatial-temporal trends.

1.5.4 Model Evaluation

The evaluation of our speed models is targeted at identifying how accurately and precisely they can predict both ETA and the journey path taken by emergency vehicles travelling under BLS conditions. Different aspects of the accuracy of routing results such as spatial, temporal, route and ETA variance are examined.

1.5.5 Hybrid Routing Model

During speed model evaluation we discover that some models are better than others at estimating the route that a responder will take. We employ optimisation methods to the best candidate speed model to refine the route prediction. This approach produces a BLRCM that can be used for accurate route prediction. The Hybrid Routing Model is a modification to the routing engine that uses the BLRCM to predict the route and performs a second pass through the estimated route using a different speed model (Metric V) to calculate the ETA. This use of two speed models improves accuracy and precision of routing calculations.

1.6 Thesis Structure

A literature review is presented in Chapter 2 covering subjects such as route planning, route reconstruction techniques, travel time distributions and route choice modelling within the emergency services domain. In Chapter 3 we present a more detailed view of operations at LAS and the performance targets they are obliged to achieve. We also describe the different operational vehicles along with a description of the on-board computing capabilities and routing technology in use.

The data received from LAS is reviewed in Chapter 4 with details on how it was loaded and cleansed. The data is characterised where necessary in relation to emergency vehicle mobility. We also explain how the BLRN is created. A baseline road speed model is formulated by the commonly used technique of snapping historic GPS data to nearest road links. This method gives us insight into the GPS data quality specific to the LAS and to coverage of the road network.

Chapter 5 introduces two map-matching techniques used to assemble route trajectories from historic emergency vehicle GPS data. Assembled routes provide us with additional road speed information that can be used to develop enhanced road speed models.

In Chapter 6 we describe our implementation of a road routing engine specifically tailored to emergency vehicles and introduce five experimental road speed models built using output from the map-matching process. This is followed by a description of how the models are evaluated.

Chapter 7 presents the results of the road speed model evaluation. We explore different aspects of the accuracy of routing results such as spatial, temporal, route and ETA variance. As a consequence of our findings during model evaluation we also describe in Chapter 8 a Hybrid Routing Model for accurate journey time and route prediction.

2

Literature Review

2.1 State of the Art in Ambulance Mobility Modelling

The study of computational techniques for the effective and efficient management of EMS resources has a relatively long history [16]. The main focus of research in this domain has been on the development of models for the placement of EMS facilities such as ambulance stations, and on strategies for resource relocation so that specific performance metrics are maximised [17, 18]. These methods typically adopt a static view of the EMS based on highly aggregate estimates of key performance indicators

such as station utilisation and makes certain restrictive assumptions for example placing significant limitations on the ability of the EMS to preform ambulance re-assignments on-the-fly. Moreover, it pays limited attention to the influence of dynamic patterns of incident generation and changing traffic conditions. This is justified by the main focus in this domain on the development of resource allocation strategies rather than on minimising ambulance response time. The following surveys [19, 20] provides a chronological perspective on the development of research in this area.

Research into the full range of EMS systems has not been surveyed in detail [21]. However, recent research has included data-driven approaches, incorporating empirical spatio-temporal factors into wider ambulance performance issues such as incident generation [22], road traffic [23, 24] and real-time vehicle coverage [25, 26]. The focus on improved clinical outcomes rather than resource allocation optimization is now preferred due to the wider availability of EMS data [27]. These data-driven approaches highlight the limitations of historical dispatch strategies, for example, the common practice of sending the nearest available ambulance to the incident is far from optimal [28, 29]. Indeed, recent work suggests that patient survivability performance can be achieved through the implementation of sophisticated strategies which take into consideration spatial and temporal factors [29].

Recent trends in ambulatory operational research have incorporated complex synthetic models employed in event-driven simulation [30, 31, 32, 33] that support a degree of adaptability to road traffic and incident generation patterns. Nevertheless, when considering the vehicle routing problem specifically, current literature typically adopts a traditional optimisation approach [34, 35, 36, 37, 38] such as linear programming with constraints using tree-based search for shortest path calculations. However, results produced by these simulations can be significantly affected by the quality of the ambulance mobility models [14].

2.2 Travelling with Blue Lights and Sirens

Travelling with BLS can be a hazardous occupation and several studies have set out to quantify the risks and benefits of this practice. In a North Carolina study of 50 journeys, each undertaken simultaneously by two vehicles, one under BLS conditions and one without, Hunt [39] found that 76% of journeys were faster with BLS, and faster by only approximately 10% overall. In a similar study the London Air Ambulance (LAA) [40] claim vehicles on BLS are more than 50% faster from a cohort of 40 journeys and recognises that this comes with the price of an increased risk in traffic accidents. Significant time savings using BLS have been demonstrated in rural areas, with benefits in urban conditions less clearly defined [41]. A Denmark study [42] including over 400 journeys found a linear correlation between GPS-estimated transport time and actual transport time, with ambulances arrival time estimated at $0.42 \times GPSestimate - 0.92$ minutes. However a recent contrasting review (cf. Murray and Key, 2017 [43]) suggests that there are minimal time savings and therefore queries the potential benefits to patients.

2.3 Route Planning

In the last decade we have seen the introduction of routing engines that can use the road network as a basis for calculating vehicle routes, such as the Northgate XC routing product and Geotracker [44] system in use at the LAS. Both engines are able to calculate routes and journey time isochrones for emergency vehicles. An isochrone is a line drawn on a map connecting points to which an ambulance can arrive after travelling for a certain time from a specific point of departure, for example, a boundary can be drawn around an ambulance describing how far it could travel in 8 minutes from the location where it was first assigned to an emergency event.

Vehicle routing is a classic graph traversal problem that is typically solved using a single-source shortest path algorithm on a directed graph consisting of edges and

nodes. Many algorithms exist [45], such as the well-known Dijkstra's algorithm [46] and its subsequent variations [47]. The Floyd-Warshall algorithm [48] is also able to compute the shortest path time but in its original form does not return the actual path. The A* algorithm [49] is a widely used fast graph-traversal algorithm. A* performance comes from an heuristic that expands potentially fewer nodes than Dijkstra. Its use in real road networks has been studied in relation to ambulances [35] and found to be faster than Dijkstra [50] as it is able to take advantage of the spatial nature of the road network. This algorithm is not able to calculate journey time isochrones as it requires a destination location and so cannot be used in ambulance systems that require this functionality.

A continuing area of research is in the field of dynamic vehicle routing algorithms that attempt to solve real-world issues that allow for changes in the road network condition after the original path has been computed. Dynamic A*, known as D* [51], was originated by Anthony Stentz and is an incremental real-time replanning algorithm known to be faster that repeatedly running A*. Several optimised variants now exist such as Focussed D* [52] and LPA* [53]. D* Lite [54] is based on LPA*, is easier to implement than the original D* and has found many uses in robotics and automatic vehicle navigation. In an evaluation of static and dynamic routing algorithms through simulation *Wang et al.* [55] found that Dijkstra's algorithm is the best choice for centralised Intelligent Transportation System (ITS) and short trips, whereas static A* is a good choice for long trips and D* where the route needs to be constantly recalculated.

2.4 Real-Time Estimation

Estimating travel times using real-time traffic data has gained considerable interest in the last few years. Musolino *et al.* [56] estimates travel routes and times for emergency vehicles leveraging real-time information gathered from ITS to continually adjust an internal model of the network cost assignments. The system was designed primarily

for dynamic assignment of routes in an emergency evacuation scenario. Additionally, novel methods of traffic avoidance have been investigated, for example the use of crowd-sourced data has attracted considerable interest[57] [58].

Several online vehicle routing engines exist, such as GraphHopper [59][60], Google [61], Waze [62], Open Source Routing Machine [63], and TomTom [64]. Online systems are generally viewed as not suitable for live emergency services operations, mainly due to resilience of internet connectivity, but do have a use in offline analytics. However, estimating travel times using civilian travel time data leads to overestimation of emergency vehicle arrival times [65].

2.5 Route Reconstruction

Map-matching (MM) is the process of route reconstruction from a raw GPS trajectory where the route is represented as a sequence of connected segments on a digital road network [66]. Position observations captured from the GPS receiver are matched to candidate road segments and the route is reconstructed by inferring intermediate segments where necessary. In an urban environment both GPS reporting frequency, position errors and quality of road network digitisation contribute to MM performance [67, 68]. The MM process can be performed in real-time [69, 70, 71, 72] or offline mode by post-processing entire trajectory, also sometimes referred to as global methods. Real-time methods are best suited to high frequency GPS position fixes where the challenge is to correctly identify the current road segment and then project the GPS observation onto it. Real-time methods are a form of Local algorithm that attempt to extend the known path whereas offline methods are often Global, acting on the entire GPS trajectory to produce the most likely continuous sequence of road segments. Depending on the frequency of GPS observation a continuously navigable set of road segments is not guaranteed in real-time MM and so offline MM is generally more accurate [73].

There are at least three main classes of MM algorithm; geometric, topological and

advanced.

2.5.1 Geometric Map-Matching

Geometric MM algorithms attempt to match GPS observations to the road network geometry. According to a recent review of MM techniques [74] the earliest use related to car navigation was a Geometric approach dating back to 1996 [75]. This commonly used method is a simple search algorithm. In *point-to-point* geometric MM each observation is matched or 'snapped' to the closest end point of a road segment [76]. This method has the advantage of being fast, however, accuracy is reduced in urban environments due to GPS error and the density of the road network [77][78]. In *point-to-curve* methods the observations are instead matched to the shape of a road segment [76]. Another approach is *curve-to-curve* where the direction and position of the observation is matched to the shape of a road segment [76][79]. Additionally, distance measures such as the Frechet Distance [73] are often employed in order to compute the most likely continuous route.

2.5.2 Topological Map-Matching

Topological MM is a natural extension to Geometric curve-to-curve methods in some respects utilising more information about the road network such as road geometry, road interconnection information, vehicle heading and speed to find the most plausible navigable route from a set of candidate road segments [80, 81, 82]. Various optimisations have been proposed such as leapfrogging GPS observations to reduce the computational burden [78]. Information Fusion [83] has been used to improve MM results by utilising historic speed information, adjusting transition probabilities based on the surrounding speed and other spatial-temporal measures. Matching a GPS position fix to a candidate road segment on the road network can be performed using just latitude and longitude, with matching quality improved further using additional sensor information such as heading [84], GPS precision and elevation information [69].

2.5.3 Advanced Map-Matching

Advanced MM techniques extend topological MM by employing probabilistic methods [74]. We give examples of three advanced map-matching algorithms relevant to our research.

Kalman Filters

Kalman Filters (KF) are often used to estimate the state of a dynamic system based on input from noisy sensors. Specifically, a Kalman filter is a form of Bayesian filtering algorithm that maintains an estimated state of a linear dynamic system together with the uncertainty of that state. The estimated state is continuously updated using observations from sensors and a state transition model implemented as a covariance matrix.

A nonlinear extension of the KF algorithm, known as the Extended Kalman Filter (EKF) [85, 86] uses differentiable functions in place of linear functions for state transition and observation models. Extended Kalman Filters have been used widely for real-time map-matching, especially when integrating GPS and dead-reckoning systems that make use of sensors such as odometers, gyroscopes and wheel sensors [87].

Despite these enhancements, KFs and EKFs only support one state belief at a time as estimated state uncertainty is represented as a unimodal distribution. For the urban road map-matching problem this restriction implies that observations need to be frequent and uncertainty needs to be minimal. As a result, multiple sensors are often employed to reduce error in dead reckoning together with a one second GPS observation frequency [82, 88]. Successful urban offline map matching has been achieved in central London using recorded GPS and sensor information from gyroscopes and odometer with GPS signals recorded every second. The system was able to estimate vehicle position using dead-reckoning for 100 seconds without GPS during the transit of a tunnel [68]. Similar approaches using map-matching and an EKF to provide sensor fusion have also

been used for real-time map-matching [89, 90], incorporating complex sensors such as laser range finders [91].

The Unscented Kalman Filter (UKF) attempts to overcome some of the limitations of the EKF by representing distributions using (deterministic) samples of points [92]. Nonetheless, it is thought that the use of Kalman Filters in systems that require nonlinear state transitions are difficult to implement, tune and suffer from numerical instability issues [93].

Our review of Kalman Filters in the map-matching domain suggests that the use of Kalman filters in a dense urban environment with infrequent GPS reports, and an absence of additional dead-reckoning sensor information, are unsuitable for offline map-matching. The deficiency of sensor information implies a restricted motion model that is limited in its ability to deal with multiple road junctions that exist in a complex urban road network. Furthermore, it has been noted that on-line methods applied on an offline basis generally show poor results [94].

Hidden Markov Model with Viterbi algorithm

A commonly used approach to offline map-matching is the use of a Hidden Markov Model with Viterbi algorithm (HMM/V)[95, 66, 96, 97, 70, 98, 99, 72, 71, 100]. Given a set of GPS observations, HMM/V attempts to determine the most likely path of the vehicle, constrained to the network. The algorithm is divided into two stages. In the first stage, a HMM trellis, shown in Figure 2.1, is constructed by generating several candidate road positions, with associated *emission probabilities* for each GPS observation in a track. Additionally, *transition probabilities* are calculated for every link between candidate road position in the trellis. The second stage uses the Viterbi algorithm [101] to calculate the most likely sequence of candidate road segments through the trellis.

Emission probabilities govern the distribution of the observed variable at a particular time given the state of the hidden variable at that time. More specifically in our context

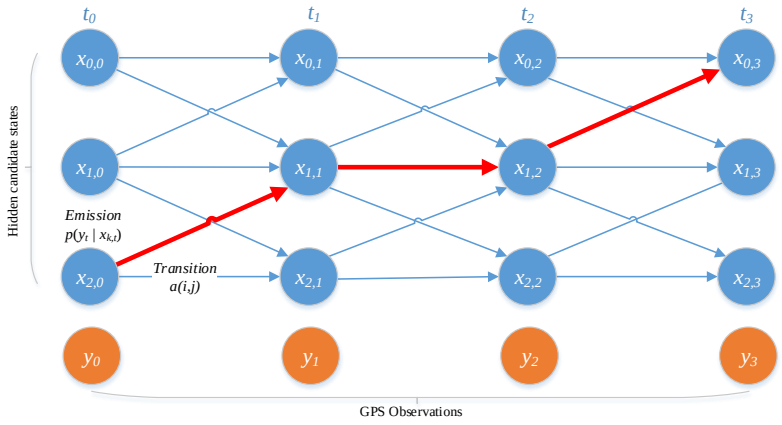


Figure 2.1: Viterbi algorithm, finding the best path through candidate hidden states by using emission and transition probabilities
Several candidates, $x_{k,i}$, are selected for each GPS observation, y_i, and their respective emission probabilities calculated. A transition function determines the probability of moving from state i to state j.

the emission PDF estimates the probability density of a GPS position fix, y, occurring a given distance from the actual location of the GPS receiver, x. The Emission PDF is used in many map-matching applications and is discussed further in Section 2.5.4.

Transition probabilities describe the probability of transiting from a hidden state, x_t, given the previous state, x_{t-1}. In GPS map-matching the transition probabilities attempt to estimate the probability of a vehicle transiting between two sequential candidate road positions. Transition functions discussed further in Section 2.5.5.

A summary HMM/V of the algorithm is as follows:

1. Obtain observations, $y_1, \ldots, y_T$ in the form of GPS fixes.

2. For each observation, generate a set of candidates, $x_{k,i}$, constrained by the road network.

3. Calculate the emission probability, $\mathrm{P}\big(y_i|x_{k,i}\big)$, of the GPS observation y_i for each candidate $x_{k,i}$.

4. Calculate the transition probability between each candidate $x_{k,i}$ and $x_{k,i+1}$ to

form the trellis.

5. Use the Viterbi algorithm to determine the most likely path through the HMM.

The Viterbi Algorithm can be used to find the most likely sequence of hidden states through the Hidden Markov Model, and in the context of map-matching, the most likely road route that the vehicle travelled along whilst emitting observed GPS position fixes. This if often referred to as the Viterbi Path.

Formally, the Viterbi path is the most likely state sequence $x_1, \ldots, x_T$, given the observations, $y_1, \ldots, y_T$, initial probabilities π_i of being in state i and transition probabilities $a_{i,j}$ of transitioning between state i and j. The Viterbi algorithm is defined by the recurrence relations

$$\begin{aligned} V_{1,k} &= \mathrm{P}\big(y_1 \mid k\big) \cdot \pi_{k_1} \\ V_{t,k} &= \max_{x \in S} \left(\mathrm{P}\big(y_t \mid x\big) \cdot a_{x,k} \cdot V_{t-1,x}\right). \end{aligned}$$

Where $V_{t,k}$ is the probability of the most probable state sequence through the first t observations, $\mathrm{P}\big(x_1, \ldots, x_t, y_1, \ldots, y_t\big)$, that have k as its final state.

Newson and Krumm [95] employed an HMM/V algorithm known to work well with noisy data, achieving a low 0.11% Route Mismatch Fraction for a 30 second sample rate, which is favourable compared to the GPS transmission interval of LAS emergency vehicles. Wei *et. al.* [97] achieved 97.0% at 64 second and 98.89% for 10 second sample rate using both Newson and Lou(09) [66] HMM/V techniques, adjusting strategies between each depending on the GPS interval. This approach attained the highest accuracy in the GIS Cup 2012 [99] amongst all competing algorithms. The Lou(09) implementation is very similar to Newson except for the addition of temporal weights on the transition probabilities, derived from speed limit information on the network. Jagadeesh *et. al.* [71] evaluated three similar implementations of HMM/V with an accuracy between 90% and 95% accuracy for 60 second intervals, with lower intervals

expected to produce better results. Finally, Lou *et. al.* [100] use HMM/V to achieve between 95%-100% accuracy on 10 second intervals from GPS-equipped mobile devices using a simple transition function based on ratios related to road segment connectivity and an emission function derived from the road network topology.

Particle Filters

The Particle Filter (PF), a term first coined in 1996 by Del Moral [102], is a Monte Carlo sampling method for performing inference in an evolving state-space model. In general PFs fall under the category of incremental algorithms. Originating from the field of robotics, the filter attempts to sequentially estimate the posterior distribution of internal hidden states given noisy or partial observations using many particles [103].

A generic particle filter, Figure 2.2, consists of an initialisation stage followed by two stages: a prediction stage and an update stage, that are repeated for each incoming observation.

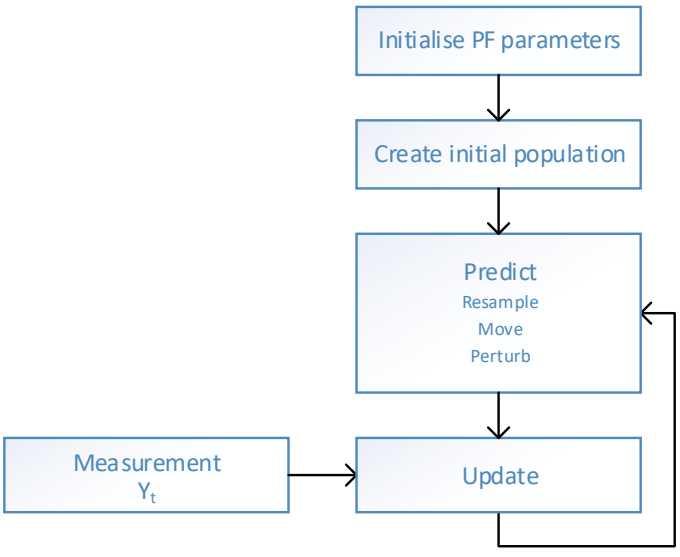


Figure 2.2: General flow of a Particle Filter

Prediction Stage - The particle filter predicts the next state using the particles from the previous step, which collectively form the posterior distribution, and a provided motion model. The prediction stage consists of these steps:

1. **Resample** - Samples are drawn from the particles from the previous step such that the weights can be normalised. This is done by drawing more samples from heavily weighted particles. This step is only necessary when there are a small number of highly weighted samples. This is discussed fully in section 5.2.4.

2. **Move** - A motion model is applied to each particle. This is a dead-reckoning form of movement without knowledge from any sensors. The selected motion model depends on the application of the particle filter.

3. **Diffuse** - The particles are then perturbed in order to maintain diversity in the state representation.

Update Stage - The update stage takes new observation data and updates the weights of current particles based on an emission function.

Figure 2.3 depicts how the state model evolves, $\{X_t : t \in \mathbb{T}\}$, through time given a sequence of observations $\{Y_t : t \in \mathbb{T}\}$. Subsequent states, x_t, are derived from new observation data, y_t, and the previous state information, x_{t-1}.

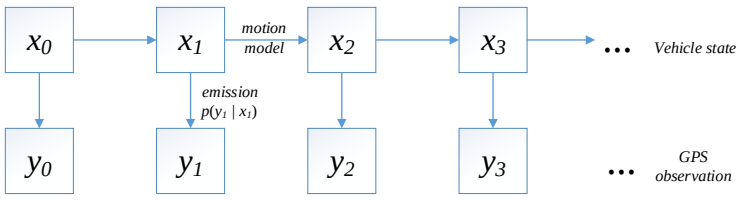


Figure 2.3: Sequential hidden and observed states of a hidden Markov Process

Consider a generic state space model x_t at time t. As the state model is unobserved, we model the initial state of the system at time t_0 by a probability distribution $p(x_0)$ represented by a set of particles. Each particle contains a state and probabilities of being

in that state. Because the states are assumed to be an unobserved Markov process and therefore conditionally independent of historic states, we can simplify the probability of being in state x_t as

$$p(x_t|x_{t-1}, x_{t-2}, ..., x_0) = p(x_t|x_{t-1}). \tag{2.1}$$

Similarly, the probability of an observation is dependent only on the current state, and is conditionally independent of previous states such that

$$p(y_t|x_t, x_{t-1}, ..., x_0) = p(y_t|x_t). \tag{2.2}$$

Our aim is to determine the probability of x_t, where $t > 0$, given the previous observations, $y_{1:t}$. This can be formulated using the recursive Bayesian equation

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}. \tag{2.3}$$

Because of our Markov assumption that y_t is independent of $y_{1:t-1}$, the denominator is constant relative to x. In practice this can be replaced with a constant, α, such that the recursive Bayesian equation becomes

$$p(x_t|y_{1:t}) = \alpha\ p(y_t|x_t)p(x_t|y_{1:t-1}). \tag{2.4}$$

The probability of an observation given a known state, $p(y_t|x_t)$, is known as the emission probability. In our case, the distribution of emission probabilities corresponds to the likelihood of an observed GPS position fix, y_t, occurring a given distance away from the actual GPS receiver, x_t. Research has shown that this distribution is non-Gaussian, with closer fits to Rayleigh and Gamma distributions [104].

Assuming we know the posterior distribution of states from the previous step, $p(x_{t-1}|y_{1:t-1})$, we can derive the current prior distribution using the equation defined by Chapman-Kolmogorov.

$$p(x_t|y_{t-1}) = \int p(x_t|x_{t-1})\ p(x_{t-1}|y_{1:t-1})dx_{k-1}. \tag{2.5}$$

This equation requires the posterior distribution from the previous state and a transition probability, $p(x_t|x_{t-1})$, which is the probability of transitioning from the previous state to the current state. The motion model provides a mechanism for determining these transition probabilities.

Particle filters are useful when the distribution of unobserved states cannot easily be represented and require non-linear methods. The computational requirements of a large number of particles can be significantly alleviated using the parallel computing capabilities of GPUs. Research on a system with up to 128K particles experience speedups up to 75 times over conventional methods [105].

Several implementations of PFs have been used with some success to perform real-time MM, and are commonly used to refine position information from GPS and other sensors constrained to digital maps [106]. Early experiments [107] were run at 2 Hz on a laptop using wheel sensors in the Automatic Braking System (ABS) to provide estimated speed and yaw rates as input to a PF, delivering accurate vehicle position constrained to a digital road map. The system proved at times to be more accurate than GPS, especially when GPS signals were poor or when the vehicle was turning. Map-matching performance can be affected by digitisation errors in the digital road network, and uncertainty of the vehicle position can result from low resolution road maps that only encode road link centreline geometry [108]. Although a large number of particles are preferred, a small number of particles (200) were successfully used to perform MM in a city environment using GPS and Microelectromechanical System (MEMS) input, generating output solutions at 1 Hz [108]. Particle filter techniques can be used without map constraints, however, Toro *et al.* [109] found that using a PF aligned with a digital map improved location estimation accuracy.

The PF filter can be used for successful MM with high frequency GPS even when

signals are lost for a short period of time, especially when heading information is still present [84]. GPS sampling rates of longer than 20 seconds are unfavourable to most PF implementations [110].

2.5.4 Emission PDF

The causes of GPS positional errors are well known and include items such as atmospheric effects due to the ionosphere and atmospheric pressure, multipath effects (e.g. from urban canyons, terrain), and inaccurate satellite clocks [111]. The problem of determining GPS position errors and their distribution from a known point is well studied [112]. The characterisation of the distribution of errors varies from study to study with most pointing to either Gaussian, Log-normal, Gamma [104] or Rayleigh distributions [113, 70, 96].

If we observe a GPS position fix, y, for the location of a GPS receiver, x, then the conditional probability density at y is based on the distance between x and y and is denoted by $p(Y = y|X = x)$. With reference to the central limit theorem, Diggelen's [114] work on Global Navigation Satellite System (GNSS) accuracy found that GPS errors are approximately Gaussian in nature. It is therefore unsurprising that Newson [113], Lou [66], Jagadeesh [71] and others use a Gaussian distribution for the emission PDF on this basis. The Euclidean distance between x and y can be represented as $||y - x||$. The Gaussian conditional emission PDF is

$$\frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(||y-x||)^2}{2\sigma^2}} \tag{2.6}$$

where σ is the standard deviation. Our two assumptions for the emission PDF are that a) the PDF is based purely on the distance between observed and actual GPS positions, and b) the distance based PDF does not vary with the location of x.

Jagadeesh uses a more computationally expensive Great Circle distance, using spher-

ical geometry, such that the distance between the points is measured as

$$distance = ||y - x||_{great\,circle} \tag{2.7}$$

rather than a Euclidean distance. This is unnecessary because at latitudes of 51°north the difference between Euclidean and Great Circle distance is less than 2cm for points approximately 50m apart. Other formulations for the emission function exist, with the normal distribution appearing to be the most popular.

2.5.5 Transition Functions

Transition functions attempt to estimate the probability of a vehicle transiting between two sequential candidate road positions. For example, the probability would be low if the speed required to make the transition was excessively low or high [115]. Assuming that a driver will take the shortest path for any given route, the direction of travel will be toward the destination [66][70]. In general this means that the vehicle will generally be travelling in one direction and that the difference between Euclidean and road route distance will be small.

There have been several attempts to characterise and model transition data in order to form robust probability functions, see Table 2.1. Earlier work by Krumm [115] assumes a Gaussian distribution based on the difference between expected route time and time difference between GPS fixes as their transition function. This was developed further by Jagadeesh and Srikanthan [70] and Lou *et al.* [66] to directly use the difference in distance between GPS fixes and the shortest path between them. Several researchers instead use an inverse exponential distribution function based on the difference between the GPS position fix distance [116] and the road route distance [95][96][98][72], each employing different techniques to solve issues with low probability values experienced in urban environments or low sampling rates.

Each of these methods have implementation drawbacks. For example, accurate time

Table 2.1: Different transition functions used in recent map-matching algorithms

Reference	Transition function, $a(x_{i-1}, x_i)$
Lou *et al.*[66], Jagadeesh and Srikanthan [70]	$\frac{d_y}{d_r}$
Newson *et al.*[95]	$e^{-\frac{1}{\beta}\lvert d_r - d_y \rvert}$
Raymond *et al.*[96]	$e^{-\beta \lvert d_x \rvert}$
Koller *et al.*[98] v1	$e^{-\frac{1}{\beta}\lvert d_r - d_y \rvert}$
Koller *et al.*[98] v2	$e^{-\beta \frac{d_x}{d_y}}$
Mohamed *et al.*[72]	$e^{-\beta \lvert d_y - d_x \rvert}$
Krumm *et al.*[115]	$\frac{exp[-0.5(\delta t_i^2)]}{\sum_j^X exp[-0.5(\delta t_i, j^2)}$

1. $x_i \in X_i$ is a set of candidate road positions at time i.
2. d_x is the distance between candidate road positions $||x_{i-1} - x_i||$.
3. d_r is shortest path distance between subsequent candidates $||x_{i-1}, x_i||_{route}$.
4. d_y is the distance between subsequent GPS fixes, $||y_{i-1} - y_i||$.
5. δt_i is the temporal error between actual and estimated traversal between fixes.

stamps are not always available, ruling out the use of the Krumm *et al.*[115] method. Distances between fixes can also be large depending on speed or frequency of reporting, leading to very low probability values as with Raymond *et al.*[96].

2.6 Travel Time Distribution

Travel time of a known route can be approximated as the sum of the average transit times for the traversed road segments [117]. In reality transit times for individual road segments vary, each having a distinct probability density function. The nature of these density functions varies according the literature, for example Tas *et al.* and others [118, 119] assume a Gamma distribution whereas a Gaussian distribution [120, 121, 122] and Log-normal distributions have also been found to be a good fit [123, 124]. Whilst characterisation of travel time distributions using Gaussian, Gamma or Log-normal

distributions might be mathematically convenient, they are not truly representative of urban travel times. Big data techniques were used by Wang *et al.* [125], employing an empirical approach to compute baseline travel time estimates between journeys with similar origin and destinations using a large volume of historic taxi trip data.

In reality the travel time distributions experienced on the road segments of a journey will vary. Assuming that the road segment travel times are independent random variables, Kim [126] proposed that the distribution of transit times for an entire journey could be estimated using convolution of random variables. An interval travel time (ITT) was also defined by Gros [127] that attempts to provide planners a best and worst-case travel time, thus introducing route reliability aspects into travel time prediction models. Gros proposes that detailed distributions are not required, an important consideration when accurate empirical distribution of travel times cannot be obtained due to sparse input data.

Increasing mobility and congestion leads to an increase in travel time variability and a decrease in reliability. Thus, reliability becomes an important performance measure for transportation facilities. One hypothesis is that the type of distribution of arrival times changes depending on the current road conditions. Guessous et al. [128] spanned theoretical and empirical methods, using point speed values obtained from loop detector data, to match appropriate theoretical distributions to different levels of services (i.e. normal, congestion etc.). In their work they considered six distributions; the Log-normal, Gamma, Burr (extended by Singh-Maddala [129]), Weibull, a mixture of two Normal distributions and two Gamma distributions. However, they admit that the technique might not be transferable as traffic conditions change considerably from one year to the next. This highlights that attempting to fit a distribution might only be of use in the short term analysis of current traffic conditions.

In a study by Westgate *et al.* [124] ambulance data was used to predict ambulance travel times at the trip level instead of the link level. This attempts to overcome the assumption that estimated travel times on individual links are independent and not

affected by previous links in a trip. Although travel time distributions calculated at the trip level are useful, these techniques have limited use in operational environments as are they not fully equipped to deal with scenarios involving the detail of road networks, such as large numbers of road blocks and area exclusions due to public events.

3

The London Ambulance Service

The LAS are responsible for providing emergency medical care in London, the capital city of the UK and the European Union's largest city. Operationally, the LAS cover the Greater London area, which is approximately 1,572 km^2 in size. The area was home to 8.174 million in 2011, growing to 8.788 million in 2016, contributing 12.5% of the entire UK population. London is also the most densely populated city in the United Kingdom with 5,235 individuals per square kilometre.

For those in London who suffered an out-of-hospital cardiac arrest during the year 2014/2015, LAS attended 10,211 arrests and attempted to resuscitate 4,665 of these

patients. Only 9.0% of patients survived to leave hospital, compared with 10.4% in USA [130]. A key principle for ambulance services is that shorter response time improves clinical outcome for patients, especially those suffering from heart attacks [130]. Current research shows that survival rates reduce by 10% for every minute between collapse and commencement of emergency life support, and that after 10 minutes very few patients survive [131]. Since 2006 overall survival rates for out of hospital cardiac arrest have been improving [130], although the exact causes of this improvement are unknown.

It is clear to see that responding quickly and appropriately to cardiac and other life-threatening incidents is critical for patient survival [132]. However, in London as of July 2017, it took up to 13.1 minutes to reach 95% of patients in imminent danger of death, and only 72.4% where reached in under 8 minutes [133].

3.1 Emergency Units

The LAS use several unit types, loosely based on the class of vehicle in use. The entire operational fleet consists of nearly 400 AEU, over 200 FRU and a smaller collection of bicycles and motorcycles. Below are listed the main types of unit that are used in the UK. The range and diversity of vehicle types are indicative of the different response and treatment requirements. Response units are used and dispatched under different circumstances, with a combination of unit types often being dispatched to a single incident.

3.1.1 Ambulance Emergency Unit (AEU)

Ambulance vehicles, Figure 3.1, are the most common vehicle type for most emergency services. For London, these are generally large modified trucks specially equipped to deal with a wide variety of medical conditions. The key feature of the ambulance, as the name suggests, is the ability to transport a patient to hospital and be able to treat them en route if necessary. Ambulances carry a great deal of emergency medical equipment

including sophisticated 12-lead Electrocardiogram (ECG), defibrillator, drugs, oxygen etc. In addition to medical equipment, ambulances are equipped with on board computing and communications capabilities, providing the necessary infrastructure for digital communications between crew and dispatch. These capabilities are described further in Section 3.8.

Figure 3.1: LAS Ambulance Emergency Unit (AEU)
Ambulances are generally crewed by two emergency medical technicians or paramedics and carry an array of medical equipment suitable for emergency care.

3.1.2 Fast Response Unit (FRU)

Fast Response Units, Figure 3.2, are operational cars fitted with medical equipment, usually carrying a single paramedic. The FRU fleet rarely carry passengers so their workflow does not include trips to hospitals. These vehicles carry a subset of equipment to ambulances and are able to carry out resuscitation, cannulate, provide drugs and assist in other tasks a paramedic might be called upon to undertake.

3.1.3 Motorcycle Response Unit (MRU)

Motorcycle Response Unit, Figure 3.3, are trained paramedics that are equipped with motorcycles loaded with medical equipment. These response units are typically used in

Figure 3.2: LAS Fast Response Unit (FRU)
The FRU is general crewed by one paramedic and carries only limited equipment. The vehicle is not suited to carrying patients so this vehicle is usually deployed where more advanced on-scene assistance is required.

specific areas where a motorcycle would be faster, such as city centres where traffic by motorised vehicle is slow due to congestion. The LAS Motorcycle Response Unit (MRU) vehicles suffer from dispatch drawbacks as GPS devices and communication computers are not fitted. Additionally, these vehicles can only carry limited medical equipment.

Figure 3.3: LAS Motorcycle Response Unit (MRU)
These vehicles are used in limited scenarios where assistance is required quickly.

3.1.4 Cycle Response Unit (CRU)

Cycle responders are trained paramedics that are equipped with bicycles carrying limited medical equipment. These response units are used in city centres where traffic by motorised vehicle cannot reach or would be slow in responding, such as parks and large pedestrian areas. These units, like motorbikes, suffer from drawbacks:

1. They can only carry limited equipment and are unable to carry heavy items such as a 12-lead ECG.

2. Without GPS , their position is not accurately unknown and cannot be dispatched automatically.

3. Accurate routing calculations for cycles are more challenging to perform than road vehicles as these can travel off-road.

Figure 3.4: LAS Cycle Response Unit (CRU)

3.1.5 Helicopters

The LAA is a charity organisation that have a fleet of two helicopters and access to several LAS fast response vehicles. The LAA attend some of the more critical incidents in and around London, and treated 1864 patients in the year 2016 [134]. The majority of these events were traffic accidents (606), stabbings and shootings (500), and falls from height (417). The aim of the helicopter unit is to deliver highly trained doctors and paramedics to critically ill patients in the shortest time possible, and treat the patient on site. Where necessary the helicopter can also transport a patient to hospital.

Figure 3.5: London Air Ambulance Helicopter

3.1.6 Community First Responder (CFR)

These are volunteer responders organised by many ambulance services. Although not necessarily trained to the levels of paramedics, the community first responders are able to deal with minor medical complaints and are able to assist paramedics if the need arises. In general the CFR has his or her own vehicle and is trained to deliver Cardiac Pulmonary Resuscitation (CPR), give oxygen, and use a defibrillator.

3.1.7 Patient Transfer Unit (PTU)

Patient Transfer Unit (PTU) are designed to take non-critical patients to and from care centres. The crew are not necessarily medically trained.

Figure 3.6: LAS Patient Transfer Unit (PTU)

3.2 Standby Points

In London there are approximately 80 Standby Points that are specific locations where vehicles and their crew will wait for work. These locations have been selected because they provide good coverage of London and also for practical reasons such as crew safety and the ability for crew to obtain refreshments. Under certain conditions considerable friction is observed between the need to meet strategic targets and positioning tactics.

3.3 Emergency Crew

There are several levels of ambulance crew depending on skill and experience. At the entry level, the new role of Emergency Ambulance Crew (EAC) has been introduced that can provide basic life support and support other crew members.

At the next level, Emergency Medical Technicians and Paramedics are able to deal with life threatening injuries and illness.

3.3.1 Emergency Medical Technician

An Emergency Medical Technician (EMT) is a healthcare provider trained to respond quickly to out-of-hospital medical emergencies. In the UK there are four grades of EMT depending on experience, with the most experienced being able to deliver treatments similar to paramedics.

All EMTs are trained to high standards and are able to deliver a wide range of treatments. Crew are able to treat patients suffering from cardiac arrest, trauma to minor injuries. EMTs are able to deliver drugs, perform cannulation, and provide immediate life support such as defibrillation and airway management.

3.3.2 Paramedic

Paramedics differ from EMTs in that they are also trained in invasive procedures such as intubation and needle chest decompressions, to treat tension pneumothorax for example.

3.4 Rosters

A 12-hour shift is typical for EMT and paramedics who work at the LAS. These shifts normally start at 07:00 and finish at 19:00, relieved by another shift that starts at 19:00 and finishes at 07:00. A single rest break is given during a 'rest break window' that starts some hours after the shift has started. The length of the break varies according to the length of the shift and is 30 minutes for a shift between 6 and less than 10 hours and 45 minutes for a shift 10 hours or longer.

3.5 Performance Targets

Performance targets change according to the politics of the day and vary across the world. In April 2011 ambulance trusts in the UK were given different targets by the Department of Health (DoH). The targets were changed such that 75% of immediately life-threatening incidents must have a first responder arrive within 8 minutes. In addition 95% of patients must be reached within 19 minutes [4]. Other non-life threatening incidents were to have locally agreed targets, and in general, must have an ambulance or first responder arrive within 20 or 30 minutes depending on the severity.

As summarised in Table 3.1, life threatening incidents, known as Category A incidents, are divided into two subcategories, Red 1 and Red 2. Non life threatening incidents,known as Category C, are divided into four Green subcategories. Category B incidents no longer exist under the UK reporting system.

Table 3.1: Ambulance Response Targets

Ambulance Response Targets from 2011 until July 2017 [4]

Category	**Response**	**Notes**
Red 1	Response in 8 minutes	Cardiac arrest
Red 2	Response in 8 minutes	Other life threatening emergencies
Green 1	Response in 20 minutes	Blue lights and sirens
Green 2	Response in 30 minutes	Blue lights and sirens
Green 3	Assessment within 20 minutes	Response within one hour
Green 4	Assessment within 60 minutes	

In the last few years the LAS have experienced ever increasing volumes of category A calls, making up some 40% of all calls. This is a national phenomenon, causing an increased burden for all ambulance services in the UK. A consequence, in part, is that this has led to a decrease in the number of ambulances arriving at category A calls within 8 minutes. Figure 3.7 produced from data supplied by QualityWatch [10], an independent research programme, shows how the trend of ambulance arrival times has been steadily declining since 2013. In this challenging environment, the LAS allow Category C performance to degrade in order to maintain Category A performance

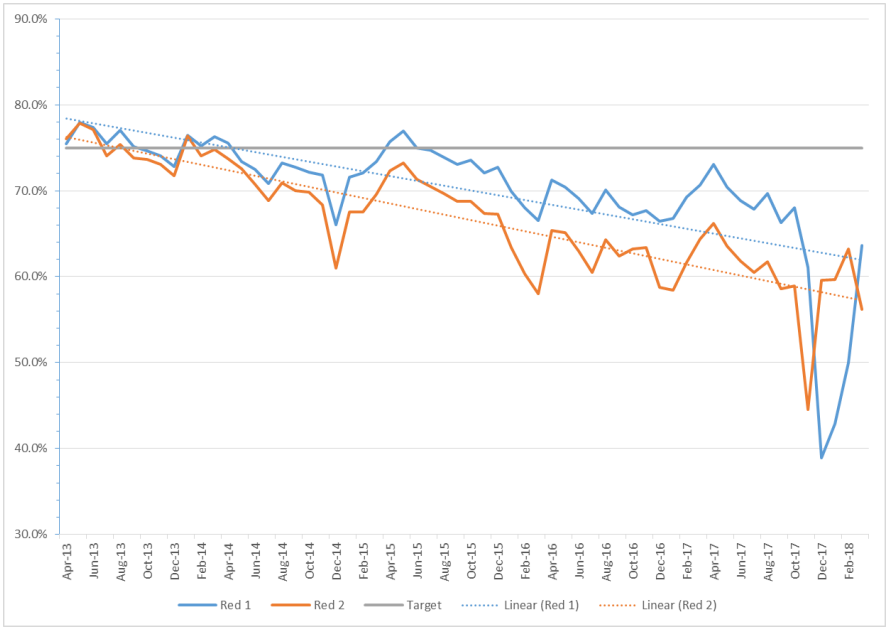


Figure 3.7: Performance of arrival times for the UK between 2011 and 2015

targets. Evidently, the LAS cannot ignore Category C performance entirely so they are given some weight in relation to category A calls. In reality, the two performance figures and targets need to be handled together.

Historically the targets set out by the DoH were designed to take into consideration the amount of data that can be practically collected by Management Information teams in ambulance trusts. As with other work cited in the literature review, it would be possible, although not considered in this thesis, to take into consideration alternative performance objectives such as:

1. Minimise arrival time to category A. This is what the dispatchers attempt, rather than just get there within 8 minutes.

2. Minimise arrival time to category C priority incidents where this does not impact category A performance. This objective has a lesser priority than category A arrival times as there is currently no financial penalty for not achieving local targets.

3. Minimise overall mileage. This increases crew welfare as well as reduce fuel and vehicle maintenance costs.

4. Minimise number of resources. Significant cost savings can be made by reducing the number of EMTs on the books.

5. Maximising crew welfare by balancing workloads and ensuring that rest breaks are taken.

3.5.1 Determining Severity

As part of the initial diagnosis process the call-taker attempts to allocate a Chief Complaint code, which broadly describes what is wrong, e.g. cardiac arrest, fall from height etc. The call-taker then questions the patients' condition to obtain a more specific determinant code, using a triage protocol known as the Advanced Medical Priority Dispatch System (AMPDS). The AMPDS code specifies the nature of the problem, for example, the code 2E1 has the description "Allergic reaction, sting or bite with Ineffective Breathing". Associated with each AMPDS code is a pre-determined attendance plan that specifies how many and which type of resources to deploy. For example, a heart attack or stroke patient will require at least one AEU to treat the patient and transport them to hospital, and one or more FRUs to assist the AEU crew. Other less urgent determinants could require either an AEU or FRU. The Computer-Aided Dispatch (CAD) system invokes the dispatch plan by selecting appropriate vehicles and sending dispatch messages. When responders arrive at the scene they assess and provide any treatment necessary. Approximately 75% of patients attended to are then transported to hospital for further assessment and treatment. Once the patient has been handed over to the hospital staff, the crew are then made available for further assignments. In many cases the crew are repositioned to a location where there is a higher chance of an incident occurring within a short distance.

3.6 Ambulance Workflow

In the UK, callers dialling 999 or 112 are initially connected to one of the national BT call centres. In turn, medical emergency calls are transferred to one of two call centres operated by the LAS, each covering a different area of London.

The procedure for taking the call and determining the location and severity of the call varies is often reviewed and amended. In general, as outlined in Figure 3.8 , the first stage in the call-taking process is to determine the location of the patient. This is sometimes aided by a system called Enhanced Information Service for Emergency Calls (EISEC) [135] that is able to pass the address or location of the caller to the ambulance control room. The next stage is to determine the condition of the patient so that an appropriate response can be made. This is obtained in two stages. Using a standard set of questions the call taker determines Chief Complaint code. This is further refined to an AMPDS code to define the patient condition more precisely. At this point the dispatch system can decide which resources (ambulances etc.), if any, should be dispatched.

The above procedure is being revised in 2018 so that before the location is identified the caller is asked some initial triage questions such as 'is the patient breathing?', 'is the patient conscious?'. This pre-triage process attempts to identify seriously ill patients early on in the process. The intention is that these Category A patients have an ambulance quickly dispatched, as the Chief Complaint and Determinant are not required to be fully assessed before dispatch is initiated.

Dispatched resources make their way to the incident, and on arrival, treat the patient. In most cases, attending paramedics and medical technicians will transport the patient to an appropriate hospital. On arrival, the crew will liaise with hospital staff and hand over the patient into hospital care. After a short period of time to complete paperwork the crew will then become available for further work. If transport is not required then the resources are made available for another event.

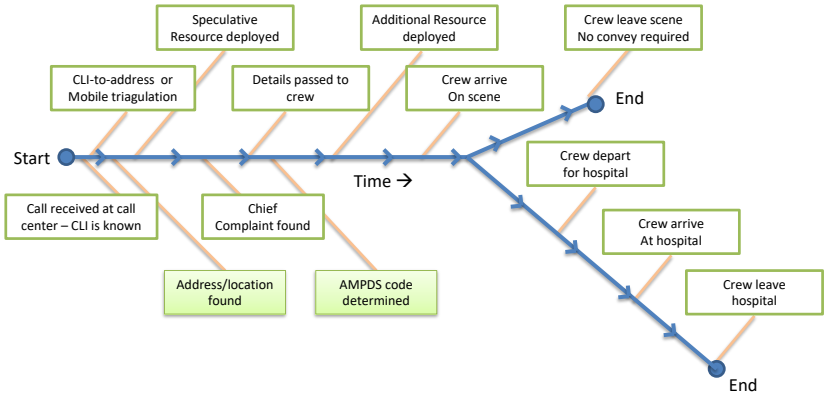


Figure 3.8: Workflow Diagram for an Emergency Incident

3.6.1 Emergency Vehicle State Transitions

The vast majority of emergency dispatches involve AEU and FRU vehicles. This research does not take into consideration placement of defibrillators or any type of responder other than AEU or FRU, such as Cycle Response Unit (CRU), MRU or helicopters as these are relatively few in number. AEU and FRU unit types adhere to a given state transition model shown in Figure 3.9 and have similar state transition models. Initially a vehicle is in a *Waiting* state. This is true when the vehicle first comes on shift or when it has completed, or been cancelled from the previous job. When a vehicle is sent a job from the central dispatch system the vehicle is automatically switched from *Waiting* to the *Dispatched* state. Crew acknowledge receipt of a job assignment which puts the vehicle into *Enroute* status. Eventually the vehicle will arrive *OnScene*, or be cancelled from the job. This results in state changes to *OnScene* and *Waiting* respectively. When vehicles are not busy, i.e. *Waiting*, they normally return to a standby point. There is no specific state for travelling to a standby point or being on it.

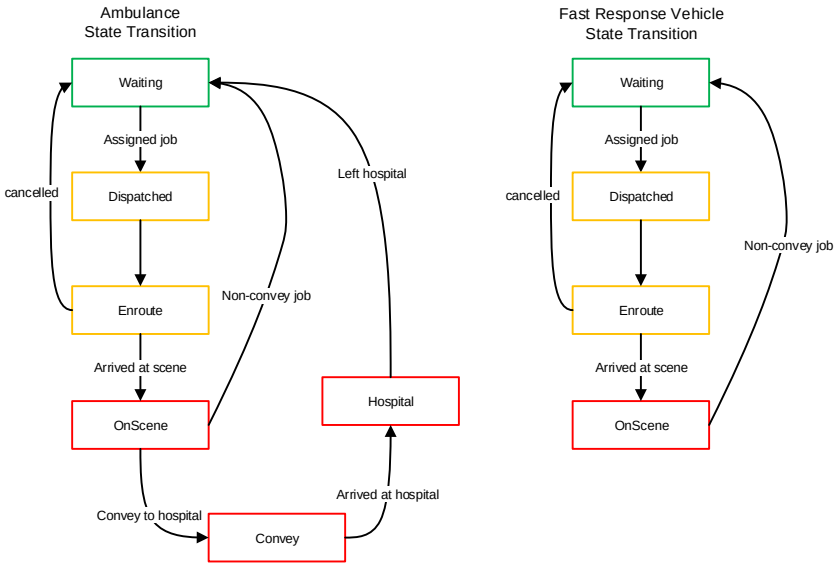


Figure 3.9: State transition diagrams for AEU and FRU

AEU have a more complex workflow as, unlike FRUs, they carry patients to hospital.

3.7 Measuring Response Times

Its worth clarifying what is meant by the response time and how it is measured. The public perception is generally that a response time is measured from the time a call is received by the emergency service until the time a paramedic arrives on scene. We use this metric within our simulations. Ambulance services in the UK measure response times differently according to the severity.

3.7.1 Category A - Red 1 Calls

For category A Red 1 incidents the time starts when the call arrives at the telephony switch at the ambulance call handling centre. For category A incidents delivered electronically, the time starts when the call is delivered to the LAS call centre. The clock stops when the first emergency responder arrives on scene. Specifically, the first response unit must be either an ambulance emergency unit (AEU), a fast response unit (FRU) or an approved responder equipped with a defibrillator accountable to that ambulance

service.

3.7.2 Category A - Red 2 Calls

The rules are slightly different for category A Red 2 immediately life threatening incidents that are not of a cardiac arrest nature. The clock starts with the earliest of:

1. the chief complaint information is obtained or;
2. the first vehicle is assigned or;
3. 240 seconds after the call has been connected to the ambulance service.

This relaxation in clock start gives the call-taker more time to correctly determine the patients condition.

In January 2015, the UK Secretary of State for Health and Social Care informed the Government that the Red 2 calls clock start time would be further relaxed for a trial period in London Ambulance and South West Ambulance services [136]. This changes the Red 2 clock start time such that the call takers have an additional 120 seconds allowing up to 3 minutes grace period to dispatch or obtain the chief complaint before the clock starts. This was increased to 4 minutes after a review had been conducted a year later. The Health Secretary understood from research carried out by the Director for Acute Care to NHS England that the tight eight minute deadlines were only being achieved through creative practice [137]. This is a summary of Directors' observations of the practices that ambulance trusts followed:

1. Dispatching resources to a 999 call, under BLS conditions, before it has been determined what the problem is, and whether an ambulance is actually required.
2. Dispatching multiple ambulance vehicles to the same patient, under BLS conditions, and then standing down the vehicles least likely to arrive first.

3. Diverting ambulance vehicles from one call to another repeatedly, so that ambulance clinicians are chasing time standards rather than focussing on patients care needs.
4. Using a "fast response unit" (car, motorbike, etc.) to "stop the clock", when this unit may provide little clinical value to the patient (e.g. stroke victim), who then has to wait a long time for a conveying ambulance to arrive.
5. Very long waits for lower priority ("green") calls that nevertheless need assessment and conveyance to hospital, and some of which have time dependent problems.

Some interesting dispatch decisions are stated, such as multiple sending of units to a single incident, whether the patient requires it or not. Dispatch protocols also change regularly as management teams grapple with the conflicts of patient welfare, crew welfare, increasing demand and performance requirements. As of July 2017 the ambulance targets changed to address these issues under the Ambulance Response Programme (ARP) [138], replacing Category A and C with Categories 1-4 where Category 1 is approximately equivalent to Red 1, Category 2 to Red 2 etc. As of 2018 the system is still being refined.

3.8 Onboard Computing Equipment

Operational vehicles carry on-board computing capabilities that facilitate navigation and communication between crew and the operations centres. Each vehicle carries extensive instrumentation that monitors its location and state including equipment temperature, handbrake position, door open, blue lights, siren, batteries, fuel level and so forth. The on-board computer, known as the Mobile Data Terminal (MDT), has a small touch sensitive display accessible to the crew.

This information is periodically relayed to the systems located at LAS headquarters over multiple wireless pathways including at least two 2G mobile telephony operators

to ensure resilience and extended coverage as well as IEEE 802.11 when an ambulance is near a station. Approximately 95% of all data traffic is received within 1 second of transmission. Retransmissions generally account for less than 1% of the total data traffic.

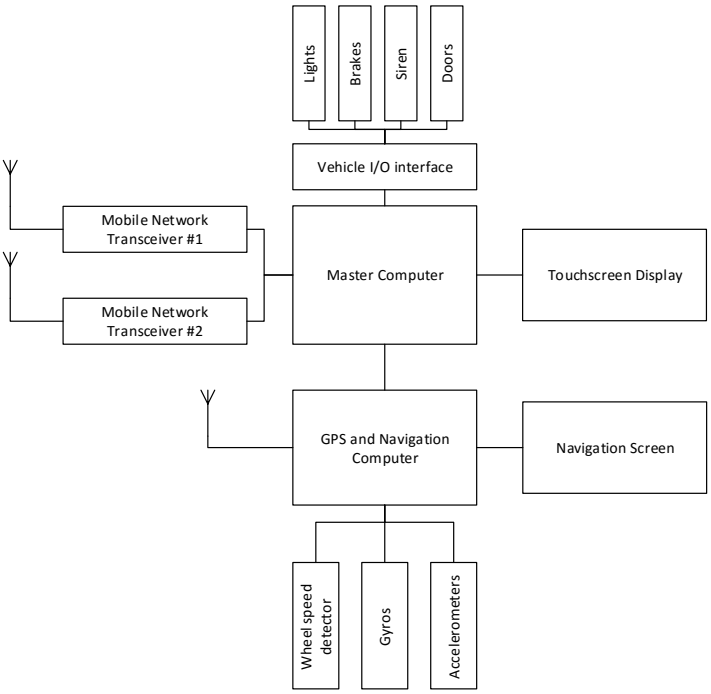


Figure 3.10: On-board computing equipment for AEU and FRU vehicles
The design employs two mobile network transceivers for fault tolerance and a separate GPS navigation system that is independent of the master computer

Emergency vehicle also carry a Siemens Navigation unit with embedded GPS receivers, gyroscopes and accelerometers augmented with wheel sensors that measure speed. The GPS system is capable of reporting accurate position data and provides navigational assistance even when satellite signals are weak; an essential feature in built-up environments.

Information sent from headquarters over the data network is used in conjunction with GPS data by the on-board computer to provide the crew with map-based navigation, search facilities and details about the patient and the incident.

All UK emergency services utilise Terrestrial Trunked Radio (TETRA) two-way transceivers, available as hand-held devices or fitted to vehicles. The devices, capable of carrying encrypted voice and data, are used by crew to communicate directly with the LAS Command and Control centre. The transceivers also contain GPS receivers and transmit location data back to headquarters.

A Siemens navigation system provides routing capabilities and geographic positioning information to the MDT. Positioning information is sent via serial link to the MDT whenever the vehicle is en route to a location such as an incident, hospital, fuel stop or standby point. The position and status payloads include motion vector and operational status information. These are transmitted together, and comprise of a Coordinated Universal Time (UTC) time-stamp, latitude, longitude, speed, heading and operational status of the vehicle. The format and coding of data packets was designed many years ago when packets sizes needed to be as small as possible. In order to reduce data volumes, the speed and heading information were encoded and some precision lost. In addition, position updates only occur when a vehicle starts to move or every 15 seconds when moving. Speed data is encoded in a 4-bit nibble representing 5-mph increments. Similarly, heading is also encoded and is transmitted in degrees in 15 degree increments.

Engineering messages are sent from emergency vehicles as and when the need arises and provide vital information to support staff on the "health" of the ambulance. In addition, software updates can be pushed to the MDT. Dispatch, status and other command messages are also sent from HQ to ambulances. These provide the core method by which the control room dispatch vehicles.

3.9 Computer Aided Dispatch

The LAS operate a Computer Aided Dispatch (CAD) system that facilitates coordination of emergency pre-hospital care across London. A key feature of the CAD system is its ability to identify and dispatch suitably skilled paramedics to patients. In many

cases this is a single ambulance, however, the proper provision of care can demand that multiple resources be dispatched to a single patient. For example, the most serious of emergency events, such as cardiac arrest, often require an ambulance and a fast-response unit to attend.

When operating in automatic dispatch mode the CAD will dispatch a vehicle that it estimates will reach the patient in the shortest period of time. It achieves this task by passing a list of available vehicles to a routing engine which returns an ordered list of vehicles in ascending order of arrival time. A comparison of the results returned from the routing engine used operationally by LAS from approximately 2008 and a Euclidean routing engine, used prior to 2008, revealed that in approximately 1 in 8 routing queries the two systems would return a different vehicle as being the quickest. Although a detailed analysis was not carried out it is suspected that certain features, such as the bisection of London by the River Thames, influence the routing results.

For the majority of emergency incidents an expeditious arrival time is not clinically significant for the patient, as the severity of the patients' condition in not expected to worsen over the course of a few minutes or even hours. However, the LAS deal with several hundred emergencies a day that are serious or immediately life threatening. These cases are expected to worsen significantly over a short period of time so fast arrival times are important. It is no exaggeration to state that making correct dispatch decisions can be a matter of life or death.

3.10 Travel Time Estimation at LAS

There are several systems at LAS that are used for travel time estimation. The first to mention is the satellite navigation unit within the vehicles. Each vehicle is equipped with a Siemens navigation unit and controlling computer that is capable of sending estimates of journey times back to HQ whilst en route to an emergency event. The Estimated Time of Arrival (ETA) is visible to control room staff and can be used

to monitor progress of dispatched vehicles. Note that this mechanism is not used to determine *which* vehicles are dispatched.

The second source of travel time estimation is the CAD vehicle routing engine based at HQ. As previously mentioned, the CAD uses the routing engine to obtain estimates of en route times between two locations using the road network. Internally the engine employs Dijkstra's shortest path algorithm using weighted edges based on vehicle type and road type. The CAD system uses the services of the routing engine when it needs to allocate or suggest a vehicle for dispatch. The routing engine has tunable weights for particular road types and does not take into consideration spatial or temporal variables.

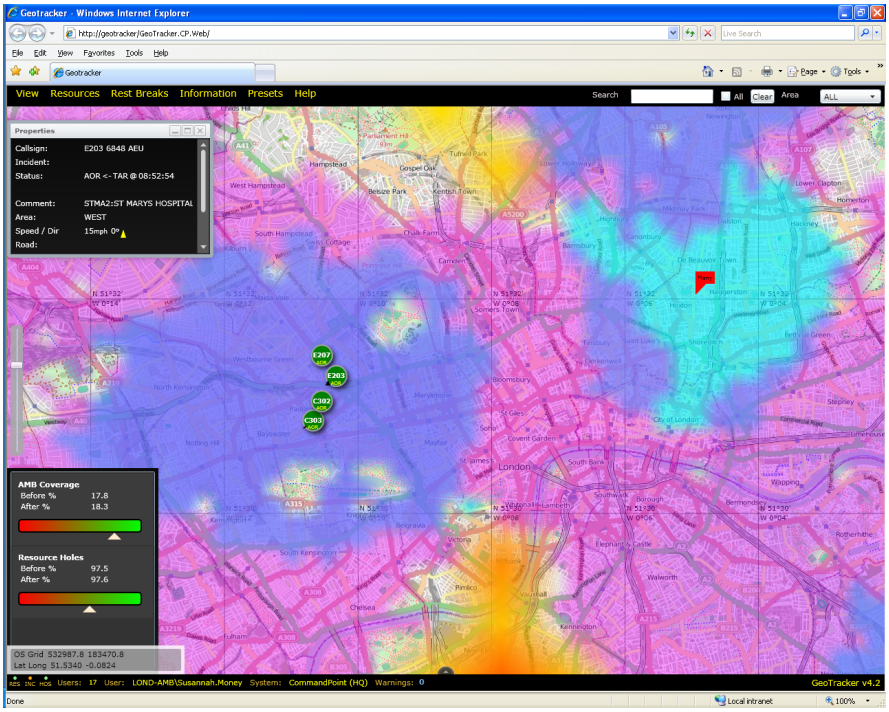


Figure 3.11: Screen shot of Geotracker coverage system at LAS
The system displays incident and vehicle heat-maps in real-time, giving controllers the ability to detect areas of low AEU and FRU coverage. In this image the yellow areas are predicted to have a high volume of emergency incidents and have no emergency vehicle coverage. Available ambulances are shown as green circles.

A third routing engine is used operationally within a Graphical Information System (GIS) called Geotracker (refer to Figure 3.11) that plots AEU and FRU coverage for

London in real-time. The engine continually calculates an eight minute drive time isochrone for each available vehicle and plots a heat-map on a map of London showing overall coverage. This map is especially important for tactical placement of vehicles throughout the day. In addition, Geotracker calculates vulnerable areas of London where medical emergencies are likely to occur and have no vehicle in the vicinity. The Geotracker routing engine is also used for various ad hoc simulations that the service wishes to carry out from time to time.

Some simulation models use Discrete Event Simulator (DES), such as one used at LAS, to emulate the behaviour of ambulance systems. Simulations are vital, for example, as a strategic tool for estimating expected behaviour of new dispatch policies and must be as accurate as possible if these models are to be relied on. To this end, emulated vehicles within the DES should move as realistically as possible, travelling along a similar route and speed that real ambulances would take. It is also important that the distribution of arrival times for a given journey are as accurate as possible as Ambulance Service performance metrics are primarily based on arrival times. In previous research, two different routing engines were used by Poulton *et. al.* [14] in ambulance simulations and found that variance in predicted Trust performance varied depending on the routing engine in use. This is significant in that reveals that the quality of results obtained from ambulance simulation research is directly affected by the underlying routing methodology employed.

4

Data Sets

In this chapter we describe the data that was kindly supplied by the LAS. The data was used throughout the research, required for reassembling journeys, speed model generation, and in general characterisation of emergency dispatch processes.

The task of handling source data is divided into two sections, a) loading and processing of the Blue Light Road Network, and b) loading, processing and analysis of LAS operational data. Section 4.2 describes the loading and remodelling the road network so that it can be used for routing emergency vehicles operating under BLS conditions. Section 4.3 through to Section 4.6 describe how the large volume of historic LAS oper-

ational data is loaded, including emergency vehicle telemetry, dispatch and emergency event data. Specific attention is paid to the GPS location dataset cleansing and a standard baseline road speed model is formulated by snapping historic GPS data to nearest road links. Characterisation of emergency event data is presented in Section 4.7, providing useful insights into the timing of events that lead up to the dispatch of a vehicle and its arrival on scene.

The outcomes of the work described in this chapter are as follows a) creation of the Blue Light Road Network and b) preparation of cleansed emergency tracks. This data provides the basis of route reconstruction and road speed modelling described in later chapters.

4.1 Ethics Approval

We applied for London Ambulance Trust Research and Development Approval for permission to access the LAS data for this research topic. Approval was granted by the Head of Forecasting and Planning at LAS. In addition, we were required to complete a Confidentiality Agreement. Additional restrictions were put in place by LAS such that patient-identifiable information was not included in the supplied data. Specifically, emergency incident data was redacted to remove patient-specific information such as name, age. Coordinates of incidents were also rounded to the nearest 100m.

In addition to LAS approval, the project was also reviewed and accepted under the Birkbeck College Ethics Review procedure, conducted by the Ethics Review Committee.

4.2 The Blue Light Road Network

An essential ingredient for our research is comprehensive knowledge of the physical road network. There are several potential sources of this information such as Open Street Map [139], Here Road Network data [140] (formerly NavTeq), Tom Tom MultiNet [141]

and Ordnance Survey's [142] ITN. We chose to use the ITN layer not just because of its availability and authoritative source but also this is the dataset that the LAS CAD system uses for routing calculations.

For our research we modify the standard ITN road network dataset so that it properly reflects the road network as it appears to the emergency vehicle driver. To this end, the network is reduced in complexity and minor adjustments made to accommodate specific rules of the road that apply only to them. This process ultimately optimises the network for consumption by the emergency vehicle routing prediction models.

The ITN is a dataset available from the Ordnance Survey that provides comprehensive information on the UK road network. The dataset contains the road centreline geometry of $550,000$ km of motorways, roads and urban paths, as well as providing additional information such as road name, type and nature. The dataset is supplied as a set of directories containing 515Mb of zipped Geography Mark-up Language (GML) files. The network information is coded as a vector model containing a set of nodes and edges. The edges represent sections of roads whereas the nodes serve to link the edges together. The edges are augmented with meta-data describing aspects such as road name, road type and any driving restrictions. Road routing information is also attached to network edges and define additional routing attributes such as: Access restrictions; Bridge Over Road; Firing Range; Ford; Gate; Level Crossing; Mandatory and Banned Turns; One Way; Physical Weight and Width Restrictions; Rising Bollards; Severe Turn; Steep and Very Steep Gradients; Through Route; Toll Indicator and Traffic Calming. Every road link also includes a geometry object, declared in Well-Known Text (WKT) format [143]. The geometry is typically a multi-line string defining the centreline of that section of road. All coordinates used within the dataset are specified as Eastings and Northing within the British National Grid map projection [144], which has a standard Spatial Reference Id (SRID) of EPSG:27700. Bi-directional roads, where the two lanes are not physically separated, are represented using a single poly-line that tracks the centreline of that road. The path traced by the poly-line is not exact and may lie some distance

from the actual centreline. This is also the case for multi-lane carriageways, only the centreline of the carriageway may be defined instead of each lane. This feature of the geometry has implications for the route reconstruction process and is fully discussed later on in Chapter 5.

4.2.1 Dealing with Level Grades

The road network is complex in that, in the real world, the road network is three dimensional. That is, there are sections of the road network such as bridges and flyovers, that lie above the ground level network. The ITN road network implements some nodes that are shared between different vertical levels of road links, even though it is not possible to route between all links attached to a node. To overcome this, each road link also defines a source and destination grade, or vertical level at which the link starts and finishes. For example, a ramp to a flyover may start at level 0 and finish at level 1, whereas, the underlying road would start at level 0 and finish at level 0. However, the ramp and the underlying road may, for the convenience of the ITN network authors, both terminate at the same node.

Figure 4.1: Sample road section showing shared nodes at different grades

The light green lines show road links at ground level. The lighter blue shows ascending road links from grade 0 to grade 1. The darker blue shows descending road links from grade 1 to grade 0. The dark red circles are shared nodes, this does not mean that it is possible to travel between all adjacent links.

The example in Figure 4.1 shows an area near London Bridge with an underpass. The light green lines show roads at ground level. The lighter blue shows ascending road links from grade 0 to 1. The darker blue shows descending road links from grade 1 to 0. The dark red circles are shared nodes, this does not mean that it is possible to travel between all adjacent links. By way of example, it is not possible to navigate from Montague Court to the A3 as the start and end grades are different.

4.2.2 Importing the Integrated Transport Network

The process of loading the network into our database is a fairly complex one. Initially data files are loaded, without translation, into the database and then post-processed to produce two database tables. These tables contain the essential information necessary for road routing.

Loading the Integrated Transport Network Data

As previously mentioned, the road network is supplied as several directories of compressed GML files, where each directory represents a specific area of the UK. The GML files for London and surrounding areas were loaded. The GML data defines several object types necessary for the construction of a road network. The objects have a relationship to each other via the use of unique ids and are defined by the schema in Figure 4.2. The GML files are parsed and database records created. Our database model implements five tables that mirror the key object types relevant for building the network. The tables, their content and record counts are as follows:

1. **Road** - 235,458 records each containing a unique id and road name.

2. **RoadLink** - 1,320,376 records containing road link information. Each link has a unique id and references a start and end node. The shape, and therefore the length of the road link is also encoded as a polyline.

3. **RoadNode** - 1,110,591 records containing nodes on the network. Each node has a unique id and can be referenced as a start or end node by a road link.

4. **RoadNetworkMember** - 1,243,596 records. These records connect road links to road names. Many roads in the England have a name and a code, so each link can potential have a reference to more than one road record.

5. **RoadRouteInformation (RRI)** - 280,683 records containing specific restrictions that may apply to a road link or pair of road links. These restrictions are necessary for accurate routing. The restrictions are prioritised in the following order:

   (a) **One Way** - the road link is one way and the direction is indicated within the RRI data. This restriction applies only to a single road link. This information is used when loading the road network into memory whilst initialising the routing engine. If this item is specified on a road link then the internal directed graph is built with only a single directed link. If this item is absent then two internal links are generated, one for each direction of travel.

   (b) **Access Limited To / Access Prohibited To** - these items restrict access in some form. These are ignored by the routing engine.

   (c) **No Entry** - this attribute marks the road link as no entry between the times given in the RRI. This is ignored by the routing engine.

   (d) **Mandatory Turn** - this is ignored by the routing engine.

   (e) **No Turn** - this is ignored by the routing engine as emergency vehicles do not obey this road restriction.

The ITN road network for September 2014 was loaded into our research database as five different tables, one for each entity listed above. Only GML files that covered the LAS operational area including a 5km buffer zone were loaded as the entire UK

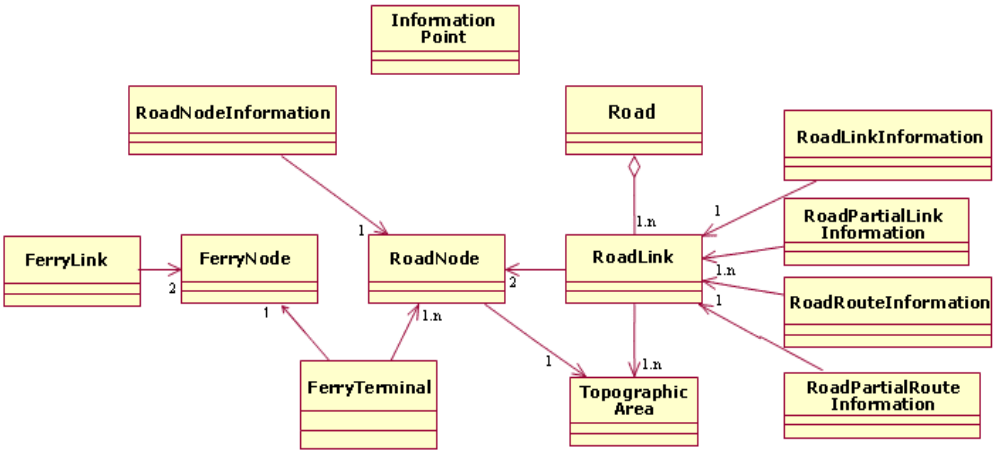


Figure 4.2: ITN road network schema

dataset was not required. The five ITN network tables are then post-processed in order to simplify the dataset and prepare it in a way that is suitable for fast loading.

Creating the Blue Light Road Network

The ITN road network is designed for civilian road users, capturing the rules that relate to them, such as permitted turns and restrictions that are displayed on signposts. We produce a BLRN, a modified version of the ITN road network that reflects the rules of the road as seen through the eyes of emergency service drivers travelling under BLS conditions. Consider a network $G = (L, R)$ where L is a set of road links and R is a set of routing connections. Each road link, $l \in L$, contains attributes such as a unique edge id, the physical geometry of the road, start and ending grade and node reference. Additionally, R is a set of routing connections consisting of ordered pairs of road link references. Each routing connection, $r \in R$, joins to two road links, such that $r = (l, l^{'})$. The presence of a connection between road links indicates that it is possible to navigate from l to $l^{'}$. Refer to an example fragment of a road network illustrated in Figure 4.3 when reviewing the following examples.

Example 1 If drivers are permitted travel along both directions of the main road

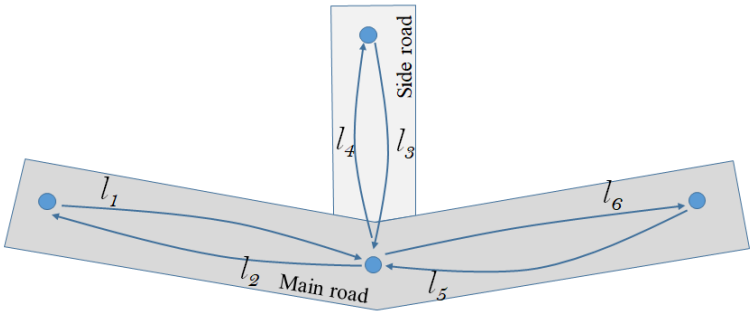


Figure 4.3: Example of a directed road network
Refer to the text for examples of how *One Way* and *No Turn* restrictions modify this network.

and also turn into the side road from the main road from either direction then the following routing information exists: $R = \{(l_1, l_4), (l_5, l_4), (l_1, l_6), (l_5, l_2)\}$

Example 2 If drivers are permitted to turn from the side road to either direction on the main road then the following routing information exist: $R = \{(l_3, l_2), (l_3, l_6), \}$

Example 3 If drivers are only permitted to turn left from the side road onto the main road then the following routing information exist: $\{(l_3, l_6)\}$

The BLRN is built from the ITN road network tables. The first stage reads the five ITN tables and flattens them into two intermediate tables: 'StaticRoadLinks' and 'StaticRoadNodes' table. The StaticRoadLinks table is similar in structure and content to the RoadLink data and also includes a two boolean fields to indicate whether a *One Way* routing restriction exists and in which direction it should be applied (i.e. To or From the direction that the link was digitised). In the second stage the BLRN is built from the two intermediate tables, expanding each ITN road link into one or two directional BLRN road links, depending on whether a *One Way* restriction exists and also taking into consideration the direction of digitisation. In addition, road link *routing* records are also produced for each BLRN road link by finding other road links with a start node equal to the ending node of the road link. The restrictions a) Access Limited To b) Access Prohibited To c) No Entry d) Mandatory Turn and e) No Turn are ignored as these do not apply to emergency vehicles.

This data import process resulted in the population of tables RoadLinkEdge and

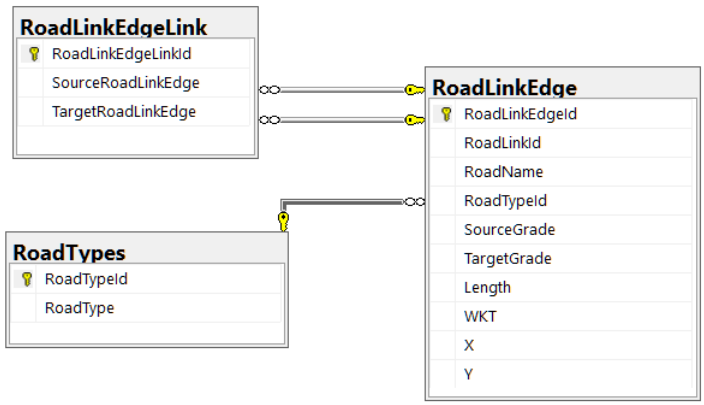


Figure 4.4: BLRN network schema

RoadlinkEdgeLink, with 336, 272 and 249, 959 record respectively. We characterise the road types by average and total length in Table 4.1. The key features of these aggregates are that a) the greatest total road length is consumed by local streets b) motorway links are long compared to other road types c) other road types have similar average lengths.

Table 4.1: Average and total length of road links by road type within London

RoadType	Average Length (m)	Total Length (km)
A Road	67.7	3,103
Alley	55.9	1,300
B Road	64.0	849
Local Street	76.3	13,165
Minor Road	81.4	2,961
Motorway	436.5	504
Pedestrianised Street	75.1	15
Private - Accessible	60.0	238
Private - Restricted	82.9	3,290

4.3 LAS Data

The data from the LAS is supplied as a set of Comma-Separated Values (CSV) files containing several datasets, namely a) emergency event data b) activation data (vehicles dispatched to each incident) and c) AVLS data - GPS and status information

transmitted from emergency vehicles. A summary of the quantities and data loaded is shown in Table 4.2.

Table 4.2: Number of data range for the source data kindly supplied by LAS

Dataset	No. Records	From Date	To Date
Incidents	7,078,201	2011-01-01	2016-11-01
Activations	6,199,278	2011-01-01	2016-11-01
AVLS Data	392,579,544	2014-03-01	2016-12-31

4.4 Emergency Event Data

The LAS receives thousands of calls a day, most of which become an 'emergency event'. Each of these events are prioritised, geo-located and recorded. The *Incident* CSV files contain a single row for each emergency event. The file contains the following columns:

- The incident identifier. This is an incrementing number encoding the date and a 5-digit sequential number for that day.
- A UTC time stamp that the call was received by the control room.
- A UTC time stamp that the first resource was dispatched.
- A UTC time stamp that the address of the patient was identified (if any).
- A UTC time stamp that the severity of the patients' condition was identified.
- A UTC time stamp that the first resource arrived on scene (if any).
- The time delay in attending to the patient
- The category of the incident defined by the NHS (see Table 3.1).
- The severity of the patients condition (see Section 3.5.1).
- The geographical coordinates of the incident.

It can be seen in Figure 4.5 that the number of R1 (imminent danger of death) incidents is relatively small, and that the most common category is R2 (life-threatening). We will be dealing predominant with these two categories of incident.

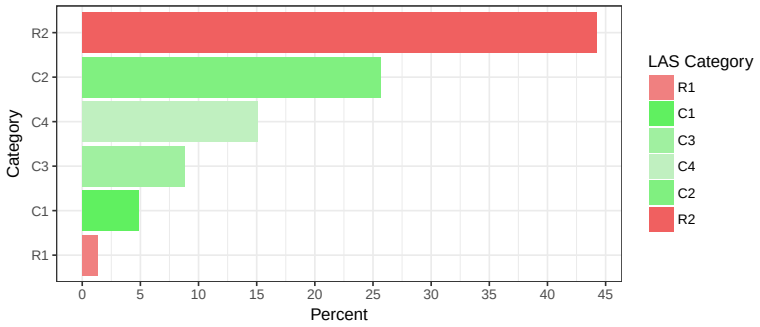


Figure 4.5: Incidents by LAS Category for the years 2015/6

Category A, which consists of Red 1 and 2, make up nearly half of all incidents.

For completeness we also include a bar chart showing the top 15 chief complaints by number of incidents, in Figure 4.6. Falls from Height top the list, with chest pain and breathing problems in the top seven.

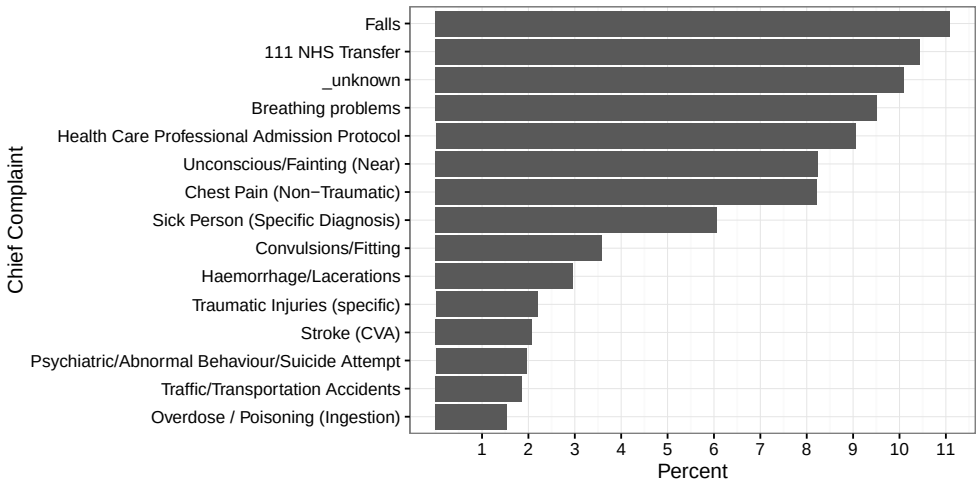


Figure 4.6: Incidents by Chief Complaint for the years 2015/6

Falls account for 11% of all incidents attended to by the LAS. Falls also account for the majority of trauma cases and death in the workplace.

Moreover, AEU and FRU travel for a specific purpose that is, in response to emergency medical incidents and the temporal and spatial characteristics of such events also follow particular patterns. These patterns follow the daily, weekly and so on routines of urban life for example with commuters flowing from the suburbs into the City of London in the morning and returning to their homes in the evening. In general, medical

emergencies are more likely to occur at specific times and places in London as depicted in aggregate in Figure 4.7.

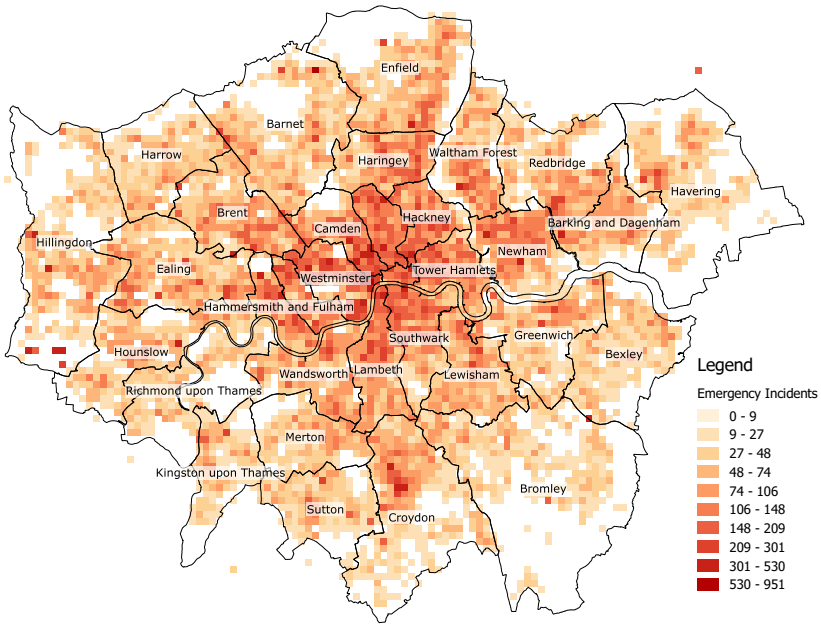


Figure 4.7: Spatial distribution of emergency incident occurrence

Category A incident occurrence density in London from September to November 2016

4.5 Activation Data

The *Activations* CSV files contain information relating to dispatch instructions for each incident. There will be one record for each emergency vehicle dispatched to an incident, and in some cases dependent on the patients' needs more that one emergency vehicle is dispatched to a single incident. The table contains rows, one for each dispatch command sent to a vehicle. Specifically, the file contains the following columns:

- The associated incident identifier, if any, that the vehicle is attending.
- A UTC time stamp that the resource was dispatched.
- A UTC time stamp that the resource arrived on scene.

- The callsign of the vehicle. This is a code that the dispatch staff and crew use on the radio for the purposes of identification.
- The responder type, such as AEU or FRU.
- An approximate position of the vehicle when it was dispatched.

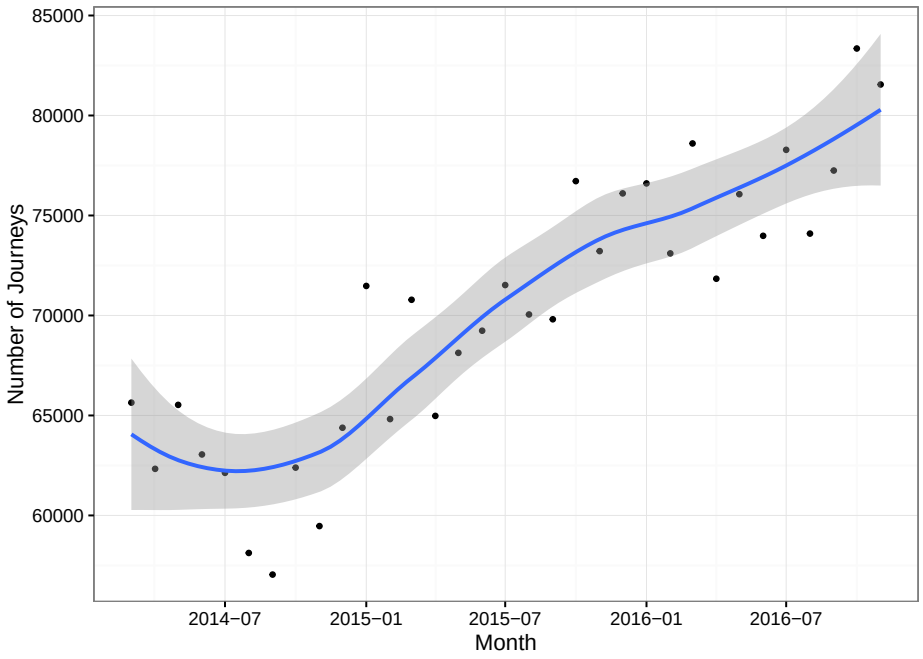


Figure 4.8: Number of journeys to Category A incidents

Black dots are the number of journeys by ambulances for each month from March 2014 to November 2016. The Blue line is a fitted line using the LOESS algorithm which is a non-parametric technique that uses local weighted regression to fit a smooth curve through points in a plot. The grey shaded area represents the 95% confidence interval.

A basic analysis in Figure 4.8 shows that dispatches to category A incidents for AEU and FRU has been steadily rising since 2014. A log-normal distribution of trip times has been observed in several studies and used as a model for predicting travel times [124, 145]. To explore how closely this holds true for emergency vehicles in London we use data from the Activations table to plot histograms of log travel times for both AEU and FRU vehicles, see Figure 4.9. It is apparent that FRUs performing shorter duration trips than AEU. There are several possible causes for the shorter trip times for FRU in addition to faster drive times, including better positioning of vehicles or

selective dispatch. We apply diagnostics in the form of a QQ-plot in Figure 4.10 which reveals that the log of both AEU and FRU trip times diverges strongly from normality for shorter journeys, and are approximately normally distributed in the central region. We also note a slight divergence from the normal for longer trips.

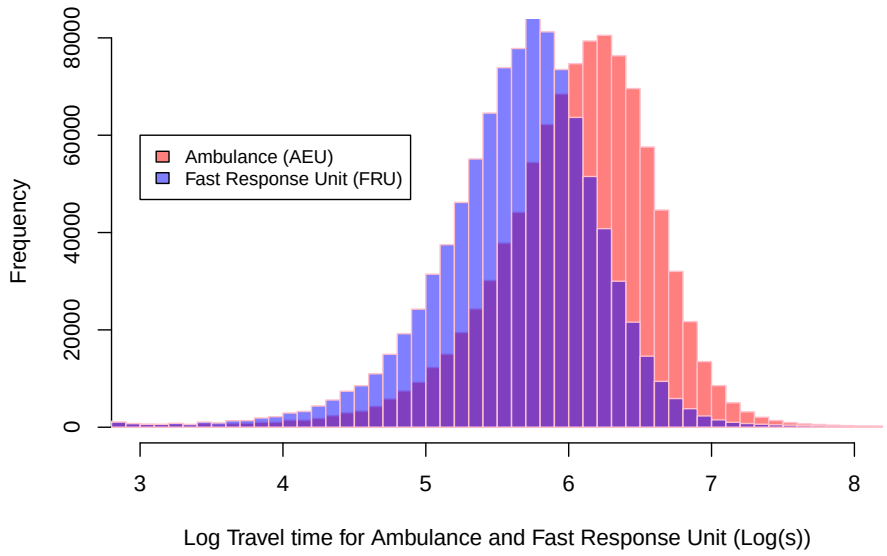


Figure 4.9: Histogram of trip times for both AEU and FRU
FRUs are shown in blue and AEUs in red. The x-axis is the log-normal of the travel time in seconds. The FRUs are clearly quicker to arrive on scene than the AEUs.

Not all calls to the LAS emergency service result in the dispatch of a vehicle. Category A Red 2 calls were by far the most common type of incident that required an emergency vehicle. Figure 4.11 shows the split between the different categories of incident. Category A Red 1 calls made up 3% with Red 2 calls making up 84% of all dispatch requests.

4.6 AVLS Processing

The *AVLS* CSV files contain data transmitted from the Automatic Vehicle Location System on board the AEU and FRU vehicles and contains the GPS position and status

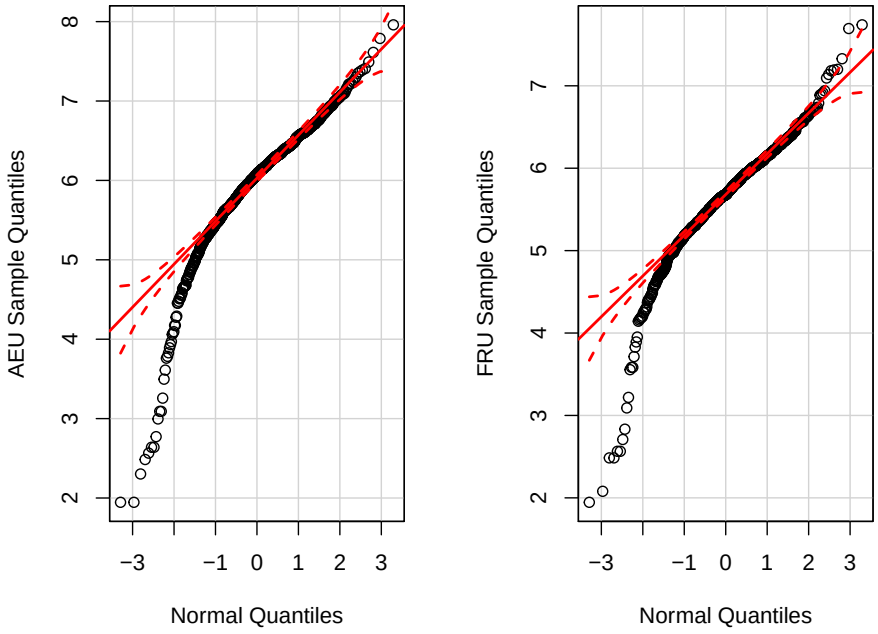


Figure 4.10: QQ Plots of log travel times for both AEU and FRU
The QQ plots of log travel time for both AEU and FRU vehicles indicates that travel times diverge strongly from normality for shorter journeys, and other journey times approximately normally distributed. There is also a slight divergence from the normal for long trips.

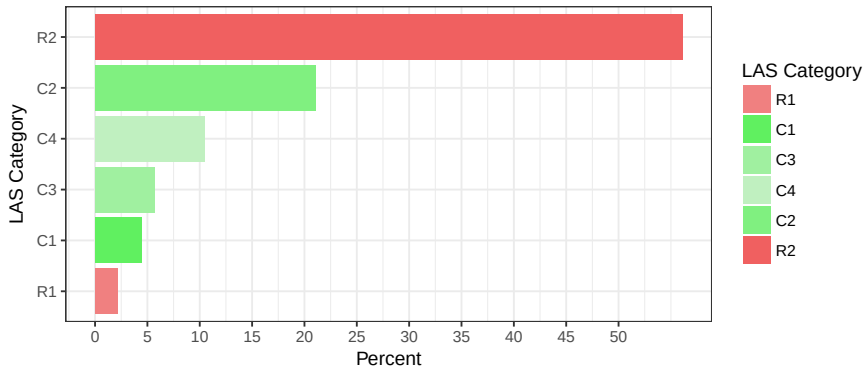


Figure 4.11: Activations by LAS Category for the years 2015/6

information. The data consists of timestamped rows, one for each position report sent from a the vehicle. Specifically, the table contains the following columns:

1. The associated incident identifier, if any, that the vehicle is attending.
2. A time stamp containing the UTC time that the report was received.

3. The callsign of the vehicle. This is a code that the dispatch staff and crew use on the radio for the purposes of identification.
4. The fleet number of the vehicle. This identifies a specific vehicle.
5. The vehicle type, such as AEU or FRU.
6. The latitude and longitude of the vehicle as reported by the GPS unit.
7. A status code such as On Scene, En route, Available etc.
8. Vehicle speed reported by the GPS unit to the nearest 5mph.
9. The heading, in degrees from true north in 15 degrees increments.

In total $392,579,544$ AVLS status records were collected by the LAS during the period March 2014 to December 2016. This data contains information relating to the position, type and status of the emergency vehicle and originates from several different vehicle types. These vehicles transit from various operational states, such as Off Duty, At Station, At Hospital etc. Our study is restricted to AEU and FRU vehicle types that are en route to a life-threatening emergency incident and so the AVLS data must be marked to identify those journeys. This AVLS dataset is augmented with a '*process*' flag such that the flag is set to true when:

1. The vehicle type is an AEU or FRU, other resource types were ignored.
2. Vehicle status is en route to an incident, i.e. in *En-route* status.
3. Vehicle was attending a Category A incident.

Table 4.3: Usable AVLS data

To Process	**No. Records**	**% of total**
AEU/FRU en route to Cat A	71,542,467	18.22%
Others	321,037,077	81.78%

Table 4.3 shows that a significant number of records were removed, with only 18.22% GPS observations remaining from the original dataset. The processing pipeline then constructs a single record for each journey. This is achieved by populating the dataset

IncidentRoutes from the AVLS data by grouping the AVLS data by callsign, incident Id, and vehicle type. Table 4.4 shows that we have over 2.3 million journeys attending 1.3 millions Category A emergency events.

Table 4.4: Number of usable emergency journeys and events extracted from LAS supplied data

Description	No. Records
AEU/FRU journeys	2,311,661
Cat A Incidents	1,367,649
Resources per Incident	1.69

4.6.1 AVLS Data Quality

As depicted in Figure 4.12 a relatively steady flow of GPS data was received throughout the sample period. The month of May 2014 saw more GPS traffic than usual. The LAS archives show that weekend of 17-18th May was one of the busiest for the Service since records began, with Sunday 18th May being the sixth busiest day ever.

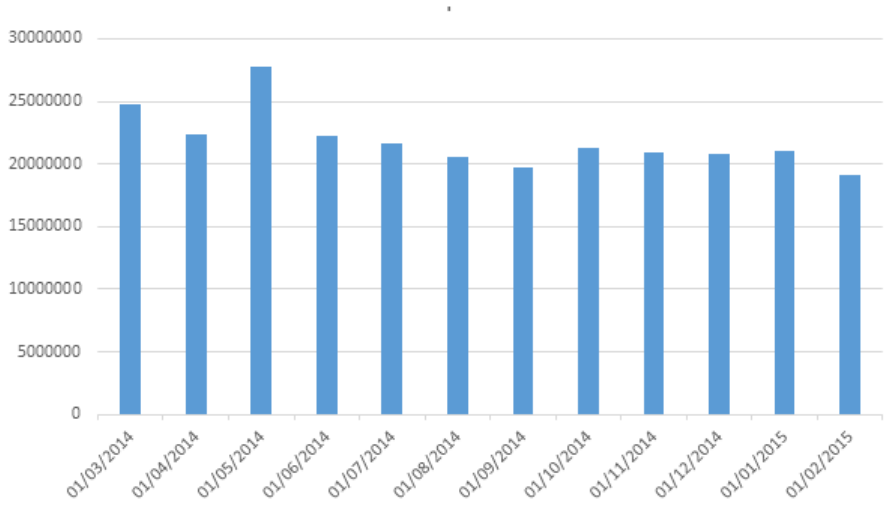


Figure 4.12: GPS fixes per month
Histogram depicting the number of GPS messages from all LAS vehicles between 01 March 2014 and 01 March 2015. The number of messages remains relatively steady at around 20 M messages per month.

We found several issues with the raw AVLS data that needed to be addressed before it could be used for route reconstruction. Some of these artefacts are generated by human error, such as delayed button presses by crew, and some generated by the system mainly originating from poor mobile data communications. We summarise the errors and our corrective action as follows.

Incorrect States

From interviews, we assessed that changing to and from *En-route* status is not completely reliable. On occasion, it appears that the crew forget to press the On-scene status for some time or that the signal was not received by HQ correctly. This impacts the calculated journey times and is responsible for many outliers in our data.

GPS Sample Rates

It is important to determine from the raw GPS data how often position reports were being sent. The GPS data was first grouped into individual tracks, by grouping records by incident and callsign. The tracks are collected as an ordered set of GPS position reports for each journey that a vehicle undertook. This data was selected from filtered GPS data so only included AEU and FRU en route to category A incidents.

This reporting frequency analysis was generated by the MapMatcherManager module (*AnalyseTrackSpeeds* method) within the software developed for this research. As can be seen from Figure 4.13, most reports come at 15 second intervals. Other notable frequencies are at 0, 10 and 20 second intervals. Further investigation into the cause of these smaller peaks revealed that this is due to the transmission of status information from the MDT to HQ. Internally, the MDT only polls the attached GPS unit once every 15 seconds. If the crew press a status change button then the MDT sends the status request with stale position information it cached during the last poll.

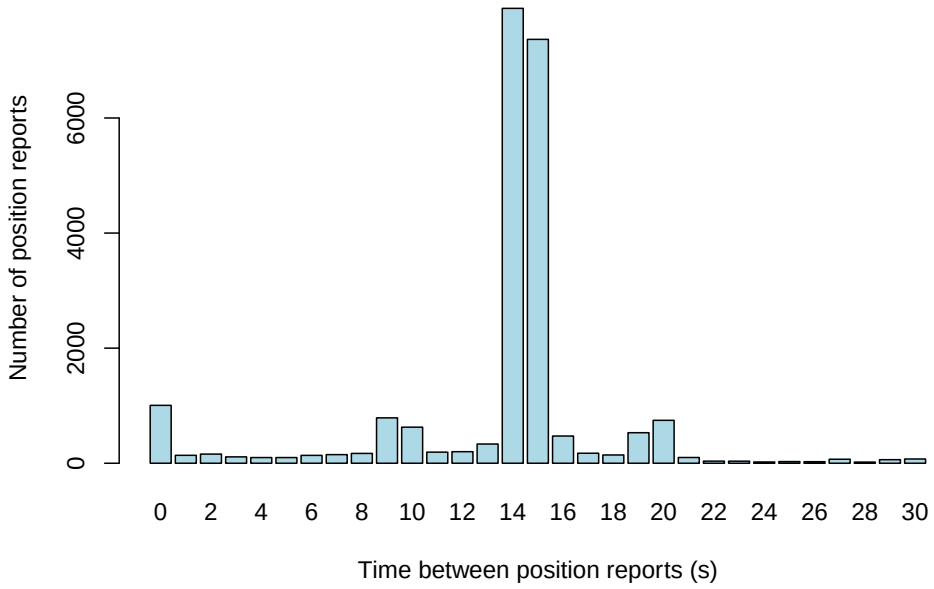


Figure 4.13: Time duration between subsequent GPS fixes

The histogram shows that most GPS fixes are spaced at 15 second intervals with other observed peaks at 0, 10 and 20 second intervals. The cause of these minor peaks at 10 and 20 seconds is due to additional status reports other than those generated by the GPS.

Duplicate Records

In addition, there appear to be several cases where multiple reports arrive together at the same time, either with different position information or duplicate position information. One source of the duplicates is thought to originate from the on-board computer which detects a network failure and transmits the same packet on the backup 2G network. Duplicate packets are discarded such that only unique records exist for each track.

Track Cleansing

We collate GPS positions into tracks representing each journey, in preparation for conversion into routes. On average each journey consisted of 101 GPS position fixes. Due to the quality of the original data, several filter conditions are applied to remove suspect fixes from a track. The filtering process moves sequentially through the fixes in the track looking for the following possible error conditions.

1. **Corrupt Coordinates** - from the cohort of AVLS messages 2,439 were corrupt and could not be converted to British National Grid coordinates (EPSG:27700).
2. **Too Close in Time** - if the time interval between two sequential fixes is less than t^f_{min} seconds then the later fix is removed from the track.
3. **Too Close in Distance** - if the distance between two sequential fixes is less than d^f_{min} metres then the later fix is removed from the track.
4. **Duplicate** - if two sequential fixes are identical in location, heading and speed then the later position fix is removed from the track.
5. **Initial Movement** - Often a vehicle will be stationary at the time the status changes to *Enroute*. The heading information is unreliable at zero and low speeds so by removing the first n^f_{skip} fixes we allow the heading information to settle. We find that 3 is a suitable number for the n^f_{skip} parameter.
6. **Excessive Estimated Speed** - Incorrect/corrupt timestamps significantly affect estimated speeds. An estimated speed is calculated between sequential fixes and if this is found to be greater than v^f_{max} m/s then the track is discarded.

The pseudo-code in Algorithm 1 describes how the track fixes are processed such that all remaining fixes conform to the above filtering conditions. In total, 365,068 (15.8%) 'bad' routes were identified out of a total of 2,311,661 emergency routes between 2014-03-01 and 2016-11-29. The bad routes were defined as any route with fewer than 4 fixes or where the estimate speed between any two fixes exceeded 80 mph.

4.6.2 AVLS Analysis

Temporal Distribution of Speeds

The routing engine in use at LAS does not have the capability to alter its speed profile based on the time of day. To test whether road speed did in fact change temporally we average the en route road speeds for both AEU and FRU for each hour of the day. The results are shown in Figure 4.14.

Algorithm 1 Procedure to clean a track ready for map matching

```
1: procedure CleanTrack(track, n^f_skip, v^f_max, t^f_min, d^f_min)
2:     last = NULL;
3:     filteredTrack = EMPTY;
4:     for fix ∈ track do
5:         if last ≠ NULL then
6:             if IsDuplicate(last, current) then
7:                 continue;
8:             end if
9:             if DistanceBetween(last, current) < d^f_min then
10:                continue;
11:            end if
12:            if TimeBetween(last, current) < t^f_min then
13:                continue;
14:            end if
15:            if SpeedBetween(last, current) > v^f_max then
16:                returnfailed;
17:            end if
18:        end if
19:        filteredTrack.Add(current)
20:        last ← current
21:    end for
       // remove 'skip' records from the front of the list
22:    filteredTrack = filteredTrack.Skip(n^f_skip);
23: end procedure
```

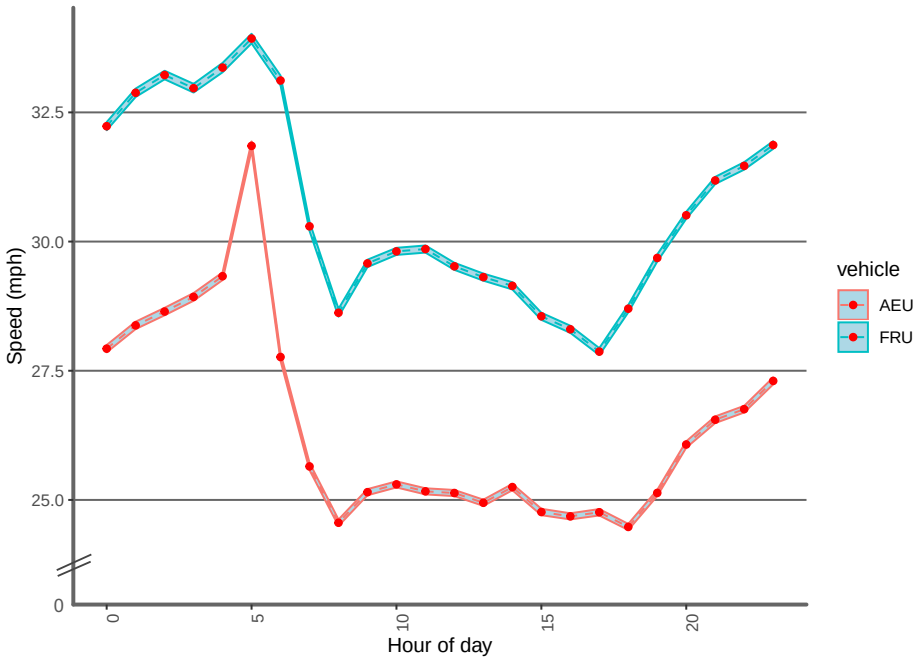


Figure 4.14: Average road speeds throughout the day for AEU and FRU resource types

It can be seen from Figure 4.14 that the road speeds fluctuate throughout the day for AEUs between approximately 24-32 mph, and FRUs between approximately 28-34 mph. In general, the pattern of fluctuation is the same for each vehicle type, with the difference in average speed of 2 and 5 mph between AEU and FRU at any given hour. AEU are, on average, slower than FRU for any given hour, probably due to the size and weight of the vehicle. Two troughs in speed can clearly be seen at the London morning (06:00-09:00) and evening (16:00-19:00) rush-hour times. A slight improvement in road speeds through the day time period can be seen, and a significant increase throughout the night.

We conclude from this that the speed profile of vehicles is significantly dependent on time of day and that routing calculations must take this into account.

Spatial Distribution of Speeds

The map showing the speed distribution in London is presented in Figure 4.15. The map was produced by plotting the average speed of en route vehicles from GPS observation grouped within 500x500m tiles. Speeds are clearly depressed in the centre of London, gradually rising towards the suburbs.

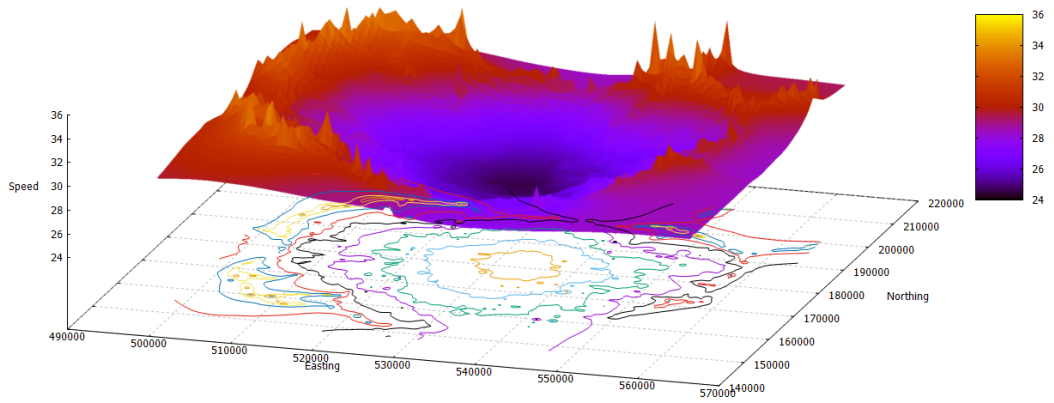


Figure 4.15: Average speed of en route vehicles across London
This is a 3D plot of GPS speeds reported by vehicles en route to an incident. Speeds are aggregated by 500x500m tiles across London. A marked dip in the 3D plot of GPS speed observations demonstrates how the centre of London is affected by congestion.

The map shown is produced in Figure 4.16 by plotting the number of GPS points within 500x500m tiles, demonstrating that the area around centre of London is the busiest for emergency medical vehicles. There are a number of minor peaks that correspond to large towns within London. The contrast in shape compared to speed map highlights the difficulties the LAS face, with the lowest speeds in the busiest areas.

4.6.3 Assessing Road Coverage

Following on from the data cleansing we investigate how historic GPS and telemetry data can be used to build a basic road speed model for emergency vehicles in the London area. The first question to answer is whether we have sufficient data covering the geographic boundaries of London as this will inform as to whether certain areas or

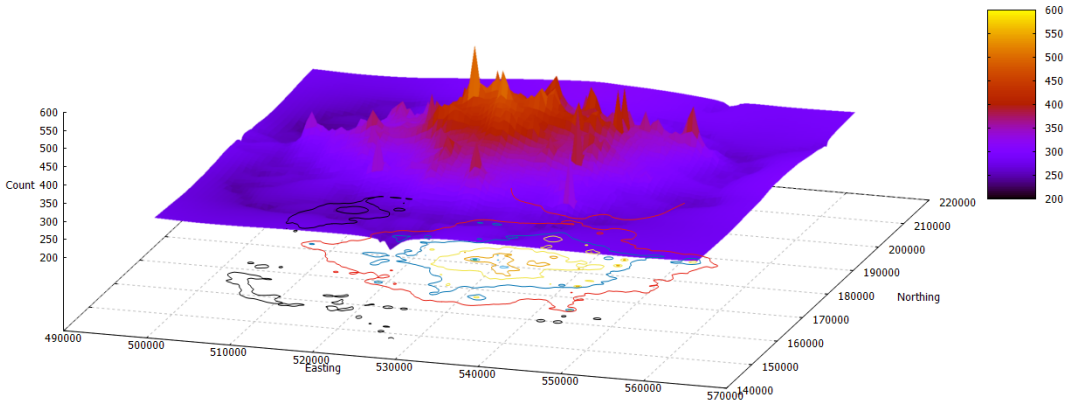


Figure 4.16: Number of GPS observations of en route vehicles across London
This is a count of GPS observations aggregated by 500x500m tiles across London. The peaks represent busy areas of mobility, occurring mainly in the centre of London.

different road types will have lower or higher quality data associated with them.

GPS Snapping

In order to assess the BLRN coverage, i.e. the percentage of roads within London that can be associated with a GPS position fix, we take every AVLS record from the 2.3 million remaining cleansed records and identify its nearest road. To achieve this we use a two-step process. First we use a primary filter to determine nearby road links. Second, we calculate which of those road links is the nearest to the GPS fix. Note that each GPS fix is taken in isolation and has no knowledge of previous fixes from the same vehicle.

The detail of the process is as follows. The road link geometries for London are first indexed using a Quad tree [146]. Each rectangular quad tree cell within the index contains references to road links that are contained *within* or *cross* its bounds. We use the quad tree index as a *primary filter* to identify candidate roads within 100 metres of a given GPS position fix. Note that we are locating these roads using the road shape rather than their start or end nodes as this allows us to find long roads that may be several hundred metres and long whose end-points are not necessarily within 100 metres

of the GPS position fix.

The nearest candidate road to the GPS position fix is determined by calculating shortest distance to each road poly-line using a distance operator. The distance operator is capable of determining the minimum distance between our input GPS position fix and a road shape. We used the NetTopologySuite library Distance operator in our implementation, which uses straightforward $O(n^2)$ comparisons for distance calculations. Figure 4.17 shows how the accumulation of unique road links over time increases coverage of the BLRN with gradually decreasing increments. Using this method we calculate that approximately 63% of the road network in London has one or more GPS fixes associated with it.

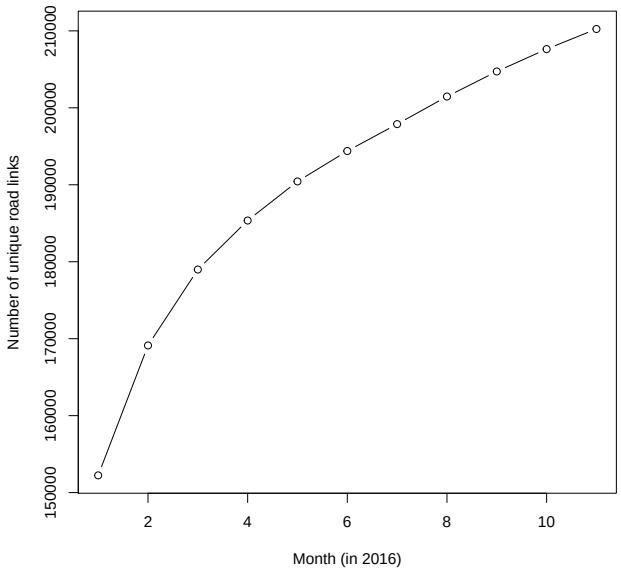


Figure 4.17: Accumulation of BLRN coverage using GPS snapping
Approximate accumulation of roads traversed by AEU and FRU over time.

To gain insight into the accumulation of unique road links over time we use the R stats::lm method [147] to fit a linear model of unique road links to the log transformed month number. This model can be used to predict the number of unique road links that could be accumulated over time. In figure 4.18 we again plot the number of unique road links and additionally overlay the predictive model for a period of up to 60

months. There are approximately 336,000 road links in London, so the model offers a coarse prediction that it could take approximately 3 years data to achieve 72% coverage (242,000 road links) and 4 years for 75% coverage (251,000 road links) of the London road network.

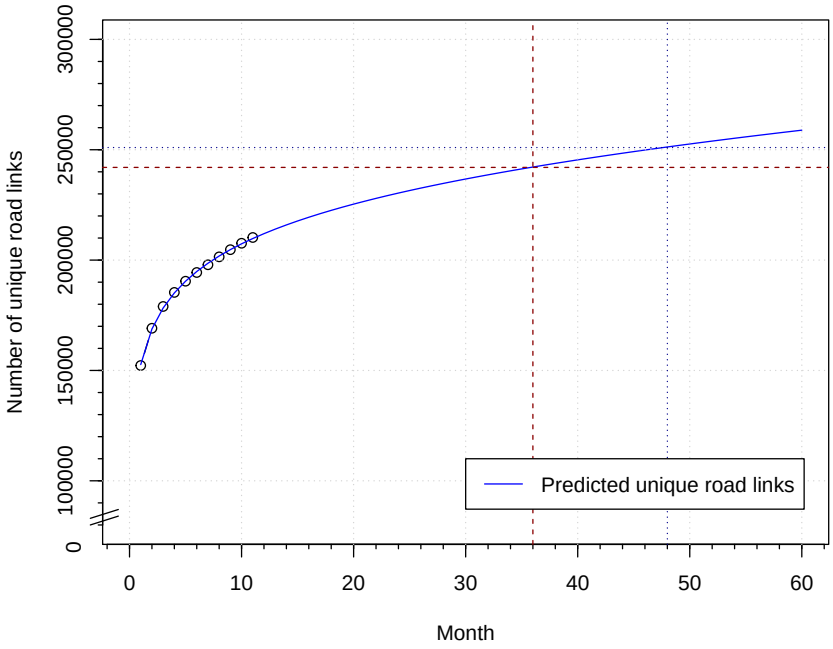


Figure 4.18: Accumulation of BLRN coverage using GPS snapping
Extrapolation of accumulated unique roads traversed by AEU and FRU over time. The extrapolation was calculated by fitting linear model of unique road links to the log of the month number. The model indicates that at least 3 years worth of data is necessary to cover approximately 72% the BLRN, increasing to 75% after 4 years.

Road Selection Errors

The causes of GPS positional errors are well known and include items such as atmospheric effects due to the ionosphere and atmospheric pressure, terrain, and inaccurate satellite clocks [111]. London's tall buildings and relatively narrow streets (known as urban canyons) play a large role in reducing GPS accuracy by obscuring satellite signals, causing a lack of visibility to an adequate number of satellites. Tall buildings with

reflective surfaces, such as those constructed of steel and glass, also create multipath effects which disrupt GPS accuracy.

Figure 4.19 illustrates how road snapping can sometimes select an incorrect road, especially where a GPS position fix happens to occur near a road junction. Any journey undertaken by an emergency vehicle will only have GPS position reports every 15 seconds. Travelling at a speed of 30mph will result in position reports approximately every 200m. From Table 4.1 we can see that the average length of each road link, except for motorways, is considerably less than 200m. We infer then that several road links will be missed and this is also clear from Figure 4.19. London has a dense road network which inevitably means that simple road snapping cannot be used reliably for accurate selection of the correct road link.

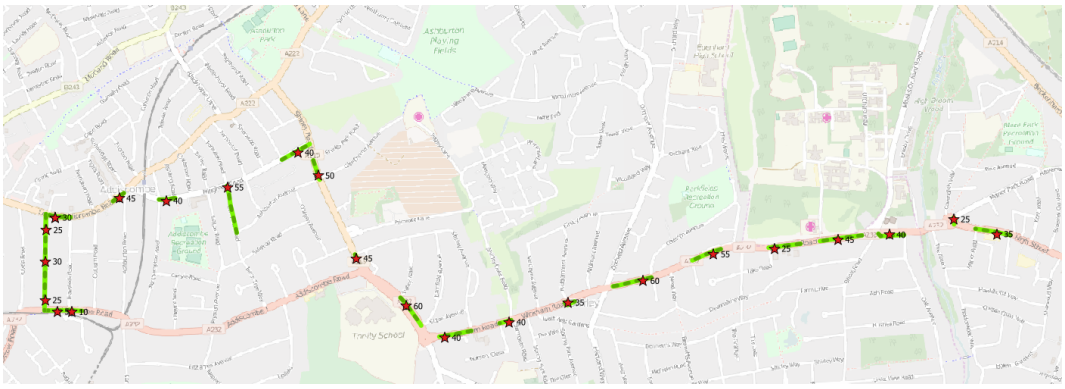


Figure 4.19: Sample route by road link snapping

Sparsity

We assess the sparsity of the AVLS data using simple GPS snapping to determine the proportion of London's road network that was covered during the sample period. Table 4.5 lists for each road type the number of road links and the percentage of that road type in the network. The last two columns show the number of GPS position fixes that were snapped, and what percentage of coverage this gives us for that road type. It can be seen in Table 4.5 that 86.49% of A roads have a GPS hit, however this only

corresponds to 13.62% of the total number of road links on the network. Similarly, local streets account for 51.33% of the overall network with only 67.79% having an associated GPS position fix. In total, only 65% of the BLRN is covered when using the GPS snapping method.

Table 4.5: Road type coverage by snapping GPS fixes to nearest road links

Road type	**Total Links**	**% roads**	**Fixes**	**GPS %**
A Road	45,788	13.62	39,601	86.49
Alley	23,249	6.91	6,410	27.57
B Road	13,259	3.94	9,753	73.56
Local Street	172,615	51.33	117,021	67.79
Minor Road	36,352	10.81	25,990	71.5
Motorway	1,154	0.34	897	77.73
Pedestrianised Street	204	0.06	161	78.92
Private - Accessible	3,968	1.18	2,120	53.43
Private - Restricted	39,683	11.8	16,669	42.01
Total	**336272**		**218622**	**65.01**

By plotting the GPS fixes, shown in Figure 4.20, we can also visually inspect the density pattern across London. The plot shows how frequently roads in London are travelled, indicated by the darkness of the roads. Arterial routes are clearly marked, and are used over and above other types of road as a key means of transport.

4.7 Call Flow

This section analyses some of the timing characteristics of the various stages in handling an emergency event. As previously described, the general process in handling emergency calls during the period 2015-2016 is to first determine the location of the caller, then determine the patients' condition prior to sending a vehicle if required. In practice, the dispatch of a vehicle can actually occur early on during the process of determining the patients condition. These key timing marks are present for each emergency event in the supplied incident data. This research is focused on category A emergency vehicle

Figure 4.20: Road usage density map obtained from GPS snapping

journey time so we analyse only those incidents in this section. Remember that category A performance is measured from call start to arriving on scene so it is important to understand the timing of the steps that lead up to dispatch of a vehicle. In reality the amount of time a vehicle has to reach the patient is less than 8 minutes as several minutes may have already been elapsed in assessing the patients location and condition.

Timing information was present for each incident record in the form of a date and time stamp that the call arrived, and several integer columns within the original dataset that are offsets in seconds that a particular event occurs. These timing marks are as follows:

1. **T0** - Call Connect - Date and time that the call was received at the call centre.

2. **T1** - Call Start - Number of seconds to answer the call.

3. **T2** - Time the patients' location was confirmed. For the period covered by our dataset this is the first step in the process and is completed by the call-taker.

4. **T3** - Time the patients' Chief Complaint was determined. This again is completed by the call-taker.

5. **T4** - Time the patients' Determinant (Patients condition or AMPDS code) was determined.

6. **First Dispatch** - The time that the first vehicle was dispatched. This normally occurs after both the location and the Determinant have been obtained.

7. **First Arrival** - The time that the first vehicle arrived on scene (which may be different from the first vehicle dispatched).

The first step in the call flow is to obtain the patients' location, the time taken to achieve this captured as $T2 - T1$. The histogram shown in Figure 4.21 shows how long it takes to do this. We estimate the parameters for a Log-normal curve using Maximum Likelihood Estimation (MLE) via the R package *fitdistrplus*. This package estimates that determining the patients' location has an approximately Log-normal distribution with the mean log of 3.81 and a log standard deviation of 0.83. The quantiles reveal that it takes over a minute (72s) to obtain the location for 75% of the calls and over 3 minutes (204s) to obtain the location for 95% of calls.

Once the patients' location has been identified, the call-taker's next step is to determine the patients condition so that appropriate treatment can be deployed. As previously described in Section 3.6 this stage consists of two steps. The first step is to identify the general condition of the patient (Chief Complaint). In the second step questioning continues until the call-taker establishes the dispatch determinant. The time taken to obtain the Chief Complaint and Determinant (AMPDS code) are plotted

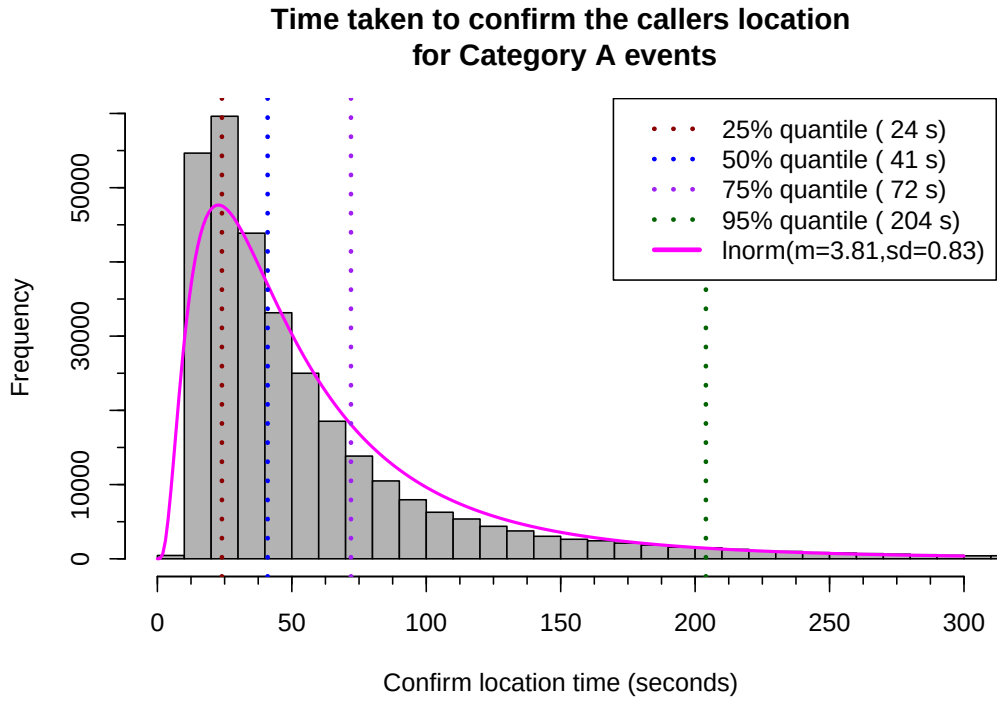


Figure 4.21: Histogram of time taken to confirm the patients' location

The duration is measured as the timespan between call start (T1) and location confirmation (T2). The eight minute target is also drawn (vertical red bar) as a frame of reference.

in Figure 4.22. The times are measured from Call Start and so also includes time taken to determine the patients' location, illustrating that obtaining the patients' condition and location can take several minutes to complete.

The action of dispatching one or more vehicles to a patient can be triggered at any time during the process of obtaining the patients' condition. The AMPDS protocol is designed to quickly assess the severity of the patients' condition such that dispatch can occur, if required, before a specific AMPDS code is determined.

Figure 4.23 shows the density plot of time taken to dispatch a vehicle relative to completing the AMPDS assessment. The data reveals that this is possibly bimodal in nature. Negative values on the left of the curve indicate that first dispatch time is often before the determinant is obtained. This because the Service can dispatch a vehicle as

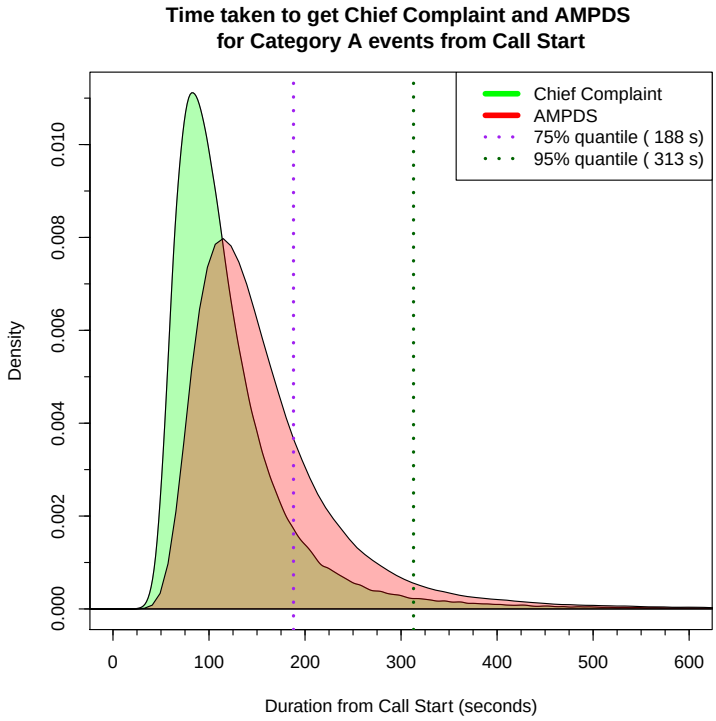


Figure 4.22: Smoothed histogram of time taken to get the patients' condition

Two plots show the time taken to obtain the Chief Complaint and Determinant. Times are measured from Call Start. The initial delay is taken up by first identifying the patients' location.

soon as the patient is known to be critically ill, which is often known well in advance of obtaining a full AMPDS code. The right hand side of the curve represent delays in dispatch after the full AMPDS code has been determined. Delays are most likely due to unavailability of vehicles. The density is approximately symmetric from which we can draw the conclusion that the benefits of early identification of a serious condition are balanced by unavailability of a resource to attend scene.

The pre-dispatch delays are summarised in Figure 4.24 and show that, when you take into consideration the delays in identifying the patients' location, condition and other operational situations, that the overall delay in dispatching a vehicle to the most critically ill patients can be up to 175 seconds for 75% of cases. This significantly impacts on the total time available for a vehicle to arrive on scene if attempting to hit

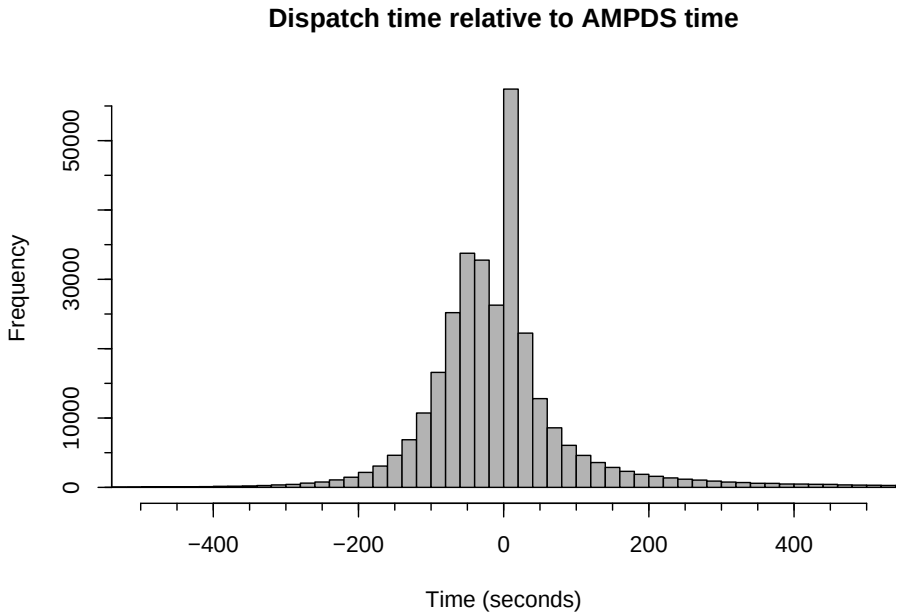


Figure 4.23: Histogram of time taken to dispatch a vehicle relative to obtaining the patients' condition

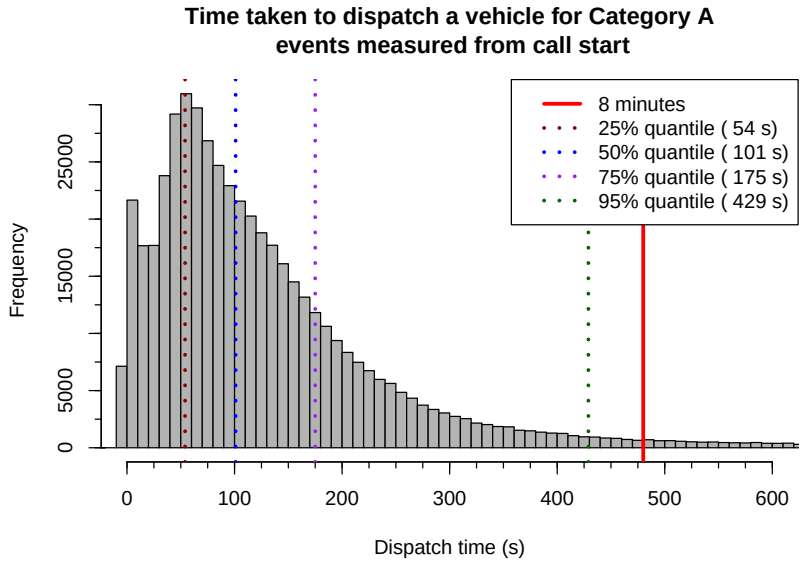


Figure 4.24: Histogram of overall time taken to dispatch a vehicle

a target of 75% vehicle arrival within 8 minutes from call connect.

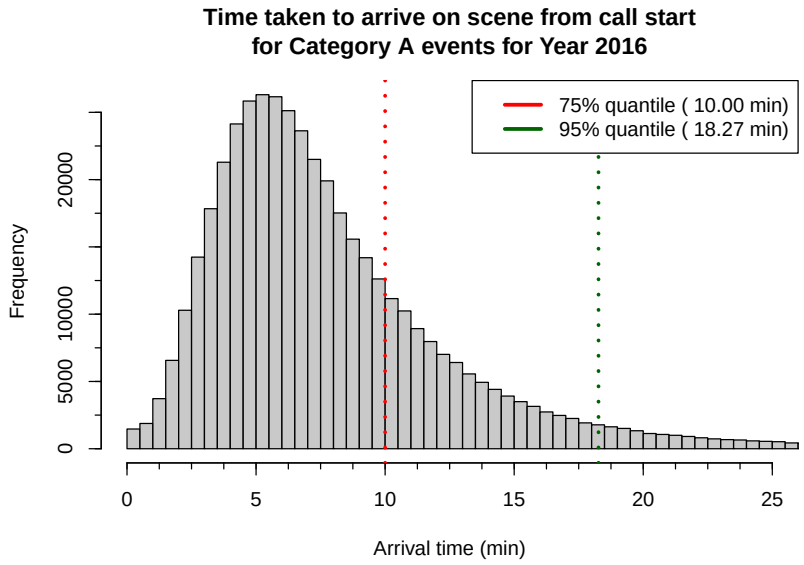


Figure 4.25: Histogram of time taken to get on scene from call connect

Figure 4.25 shows the distribution of arrival times for category A calls. Hospital transfers which were also included in the data are excluded from this analysis as these events occur outside of the standard 999 call workflow. Recall that performance targets for Category A incidents for the year 2016 were set to 8 minutes for 75% of incidents and 19 minutes for 95% of incidents. Only 61.43% of incidents were attended to within 8 minutes. For a different perspective, the 75% of arrivals took up to 10 minutes. The 19 minute target was met as 95% of incidents were attended to within 18.27 minutes.

Figure 4.26 shows a breakdown of journey time, i.e. from dispatch to arrival, for both AEU and FRU vehicles. FRUs are more that three minutes faster when measured against the 75% quantile. In many cases multiple vehicles are sent to a single incident for both for clinical and performance reasons. Multiple crew are often dispatched where more that two technicians are required to treat the patient. Additionally a vehicle can be diverted from lower priority work, at the discretion of the dispatcher if they feel that this vehicle will arrive sooner than any other vehicles dispatched. These performance metrics

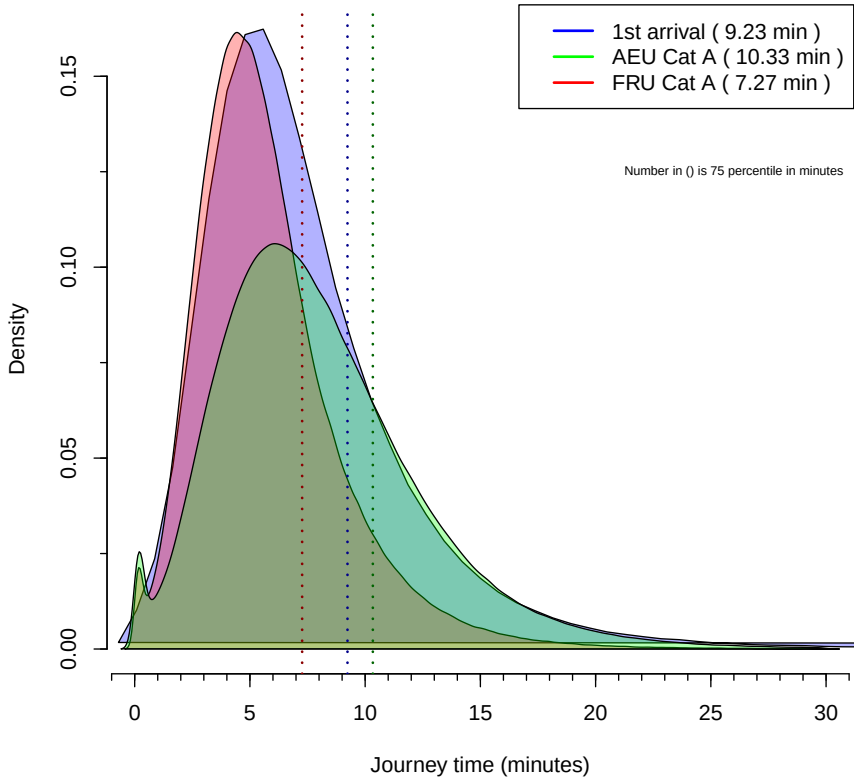


Figure 4.26: Histogram of actual AEU, FRU and First Arrival Cat A journey time

Histogram of journey time for individual journeys in AEU and FRU vehicles. Additionally, First Arrival is also plotted which is from the incident perspective, i.e. how long from the first dispatched vehicle to the first vehicle to arrive. It must be noted that the first vehicle to arrive might differ from the first vehicle dispatched.

show that a large proportion, roughly a third, of the allocated eight minute target is consumed by verifying the patients' location and condition, with the remaining time consumed with travelling to the patient. Significant reduction in travel time is achieved by allocating multiple vehicles to a single incident.

5

Route Reconstruction

We use the technique of map-matching to re-assemble the entire route taken by an emergency vehicle from GPS observations, from which we estimate vehicle speeds along road links not captured from GPS road link snapping.

From cleansed GPS and BLRN information we estimate the initial parameters necessary to run our two candidate map-matching techniques. Evaluation of reconstructed routes is performed on synthetic routes, and parameters subsequently refined using those routes. Finally, we map-match our entire historic GPS data to produce a database of reconstructed journeys carried out by emergency vehicles on BLS. Our results show that

the HMM/V algorithm was most suited to the task of map matching with urban low frequency GPS position fixes.

5.1 Map Matching Parameter Estimation

Two candidate map-matching algorithms are considered for route reconstruction: a) Particle Filter and, b) Hidden Markov Model with Viterbi. Refer to Section 2.5.3 for a description of these algorithms. Both require several tuning parameters and selection of appropriate Probability Density Functions (PDF) that govern their behaviour and accuracy of matching. We describe briefly the parameters used by the two algorithms:

1. **Emission PDF**, $p(y|x)$ - Estimates the probability density of a GPS position fix, y, occurring a given distance from the actual location of the GPS receiver, x.

2. **Transition Probability**, $a(x_{i-1}, x_i)$ - Estimates the probability of transitioning from one state, x_{i-1}, to the next, x_i. In map-matching a 'state' refers to the location and discrete motion vector of a vehicle.

3. **Road Geometry Range**, d_{geom} - Used when locating roads nearby to a GPS position fix. This parameter determines the maximum distance, in metres, that a candidate road can be from a GPS fix.

4. **Maximum Routes**, n_{rmax} - Determines the maximum number of candidate routes to be considered between sequential GPS fixes. Only used in HMM/V.

5. **Number of Particles**, n_p - Determines the number of particles to generate. Only used in the PF algorithm.

6. **Re-sample Cutoff**, n_{cutoff} - This is a threshold parameter that triggers regeneration of particles. Only used in the PF algorithm.

7. **Max Candidates**, n_{cmax} - The number of nearby road positions to consider for each observation.

5.1.1 Emission PDF

The Emission PDF estimates the probability density of a GPS position fix, y, occurring a given distance from the actual location of the GPS receiver, x. From the candidate emission functions described previously in Section 2.5.4 none take into consideration that we are attempting find an emission probability for a vehicle constrained to a digital representation of the road network. The GPS receiver antenna on most emergency vehicles is located centrally along the longitudinal axis of the vehicle. The polylines defined in the BLRN describe only the approximate layout of the road, and not the physical geometry. For example, a single bi-directional carriageway, known as an S2 road, is represented by a single polyline that tracks the centre-line of that carriageway. Central reservations, median lines, bollards and other obstacles are dealt with by splitting the polyline into two directions of travel as shown in Figure 5.1, item A.

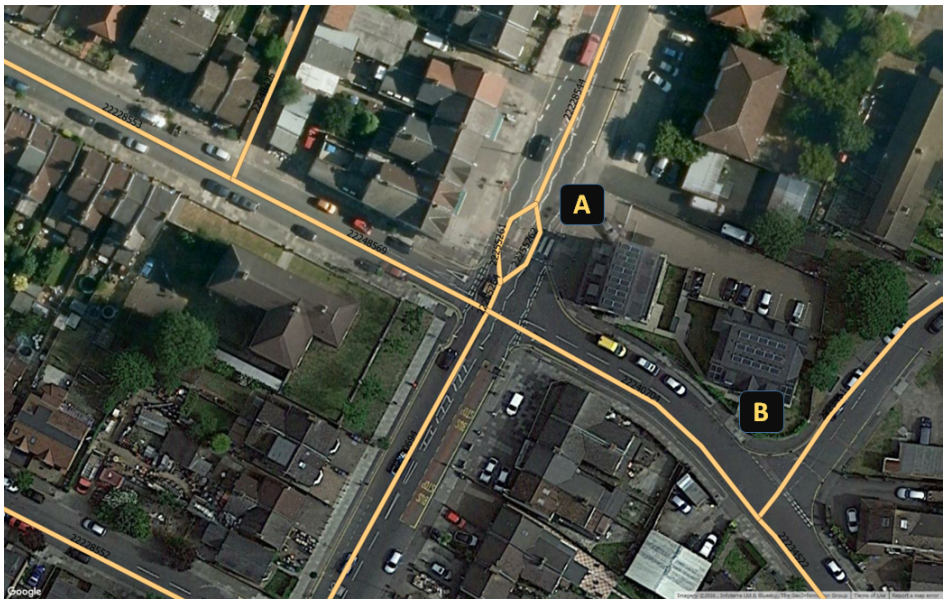


Figure 5.1: Satellite image of a section of London overlaid with BLRN road polylines

Label A) shows how islands, central reservations (or median strips) are represented with a polyline dividing into two sections around the island and then merging back into a single polyline. B) It can be seen in this section that the polyline is not always accurately positioned over the road centreline.

The lines are annotated with metadata indicating whether it allows one-way or bidirectional travel. Although there are no standard road widths in the UK, a typical S2 road can vary between 4.5 and 7.5 metres in width, with traffic flowing on two opposing lanes. As the vehicle will be generally on the left or right lane depending on the direction of travel, we can expect the physical location of the GPS receiver antenna to be offset from the centreline of the physical road by several metres. Additionally, the accuracy and quantity of coordinates captured by the digitisation process will add further differences between the physical and digital representations of the network, as shown in Figure 5.1 item B where the digital centreline is some distance from the physical centreline. Because of the differences between the physical road network and its digital representation in the BLRN we expect to find a non-Gaussian emission distribution of observations.

We can easily compare our existing large dataset of GPS fixes to the geometries contained within the BLRN to get a general insight into how close GPS fixes are to the digitised road centrelines. The distance from a fix to the nearest road segment in the BLRN can be calculated by extracting all lines for each road segment in the BLRN and then finding the line that has the minimum distance to the fix. The distance from a point to a line can be calculated by rearranging the formula for the area of a triangle to yield the height, h, as follows:

$$h = \frac{2A}{b}, \tag{5.1}$$

where A is the area of triangle calculated using the coordinates of the point and line endpoints, and b is the length of the line. The height, h, is the shortest distance from the point to the line.

One million GPS fixes were extracted from the AVLS dataset and associated distances to the nearest road calculated. A Cullen and Frey graph [148] can be used to visually compare the skewness and kurtosis of several univariate parametric distributions against our data. Figure 5.2 is a bootstrapped Cullen and Frey graph using the

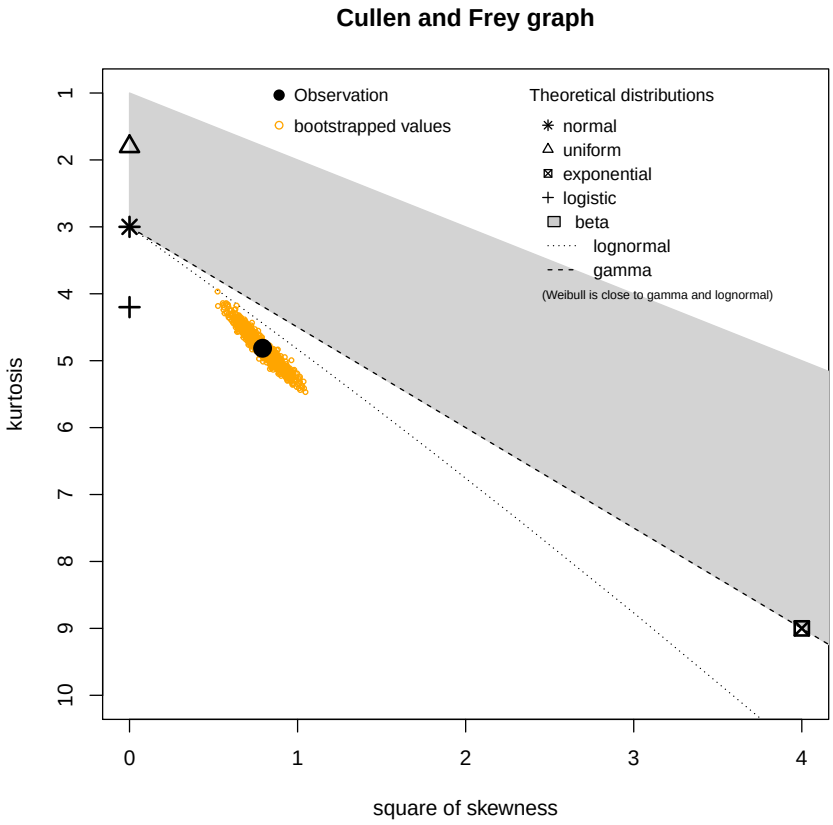


Figure 5.2: Cullen and Frey graph of GPS distance to the nearest road
The position of the large black dot (Observation) relative to the distribution lines visually represents how well the *DistanceToRoad* data fits these theoretical distributions.

GPS Distance-to-Road data. It gives insight into whether our approximation of the distribution $P(Y|X)$ is similar to one or more standard unimodal distributions. The black dot (observation) lies outside the range of possible distributions which indicates that none perfectly fit the data we obtained.

Further exploration can be achieved by using Q-Q plots for common probability distribution comparisons. Figure 5.3 compares the Gamma, Gaussian, Log-normal and Exponential distributions with our *distance-to-road* data. The Q-Q plot is constructed by plotting empirical quantiles (black dots) on the Y-axis against their theoretical counterparts on the X-axis. The red line on each graph is fitted through the 1^{st} and 3^{rd}

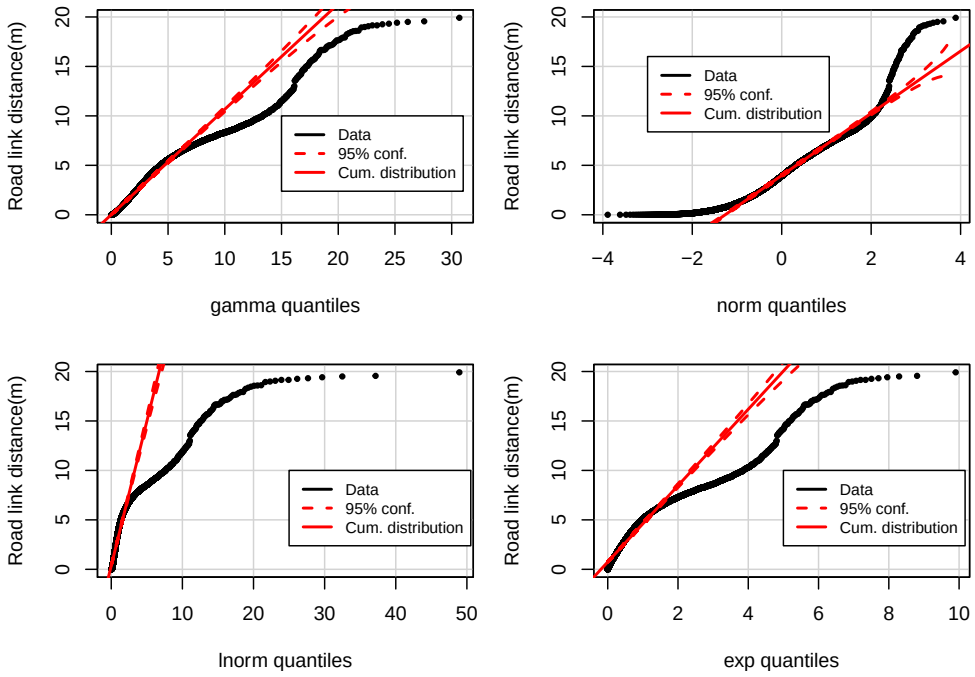


Figure 5.3: Four Q-Q plots comparing *distance-to-road* with standard distributions: Gamma, Gaussian, Log-normal and Exponential

All except the exponential distribution fit reasonably well in some sections of the quantile. Gamma and Log-normal appear to fit well at close range with the Gaussian distribution fitting well from distances further than 2 metres.

quartiles. The 95% confidence envelope is based on the standard error of the order statistics of an independent random sample from the comparison distribution.

The Gaussian distribution seems to fit well between 2 and 7 metres. The Log-normal and Gamma distributions fit reasonably well up to approximately 5 metres and then diverge rapidly. This series of plots suggests that none of these standard distributions fits our data.

Figure 5.4 shows a histogram of GPS distances to the nearest position fix, in 0.5 metre increments. For illustration we fit a Gaussian curve (dash magenta line, $\sigma = 0.22$) over this histogram. Rather than the expected Gaussian distribution cited in previous works the histogram appears to be possibly bimodal with two peaks at ranges 0.0-0.5 and 5.5-6.0 metres. This could be caused by an artefact of the digital representation of

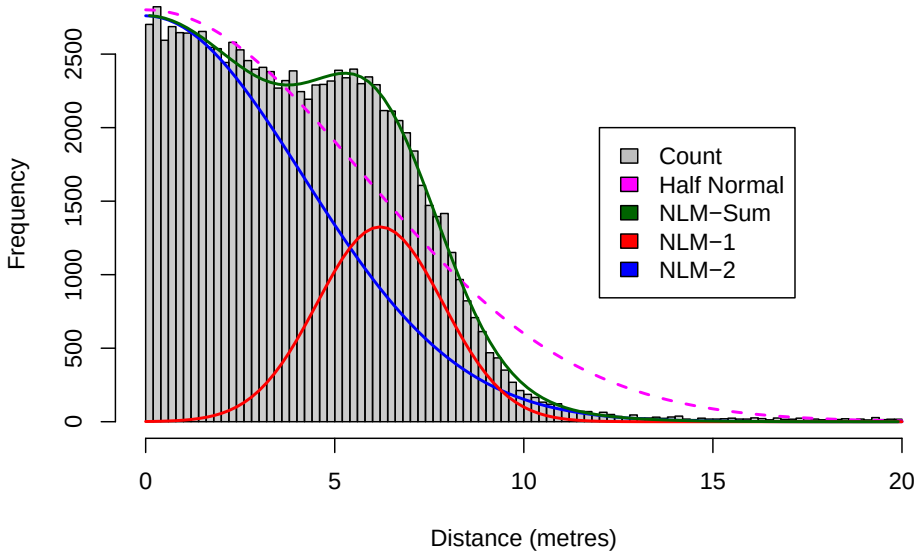


Figure 5.4: Histogram showing the density of distances, in metres, of GPS fixes to the nearest road geometry
The histogram appears bimodal with two possible peaks at ranges 0.0-0.5 and 5.5-6.0 metres. The dashed-magenta line is a half-normal distribution with $\sigma = 0.22$. A non-linear model (NLM-Sum, in green), is the sum of a normal (NLM-1 in red) and a half-normal distribution (NLM-2 in blue), fitted using the R *nls* method.

the BLRN due to the fact that a *single* polyline is used to represent the two lanes of a single carriageway S2 road. To investigate the bimodal nature of the data we also fit a non-linear model composed of a normal and half-normal distribution using the following equation:

$$P(x) = s_1.N(\mu, \sigma) + s_2.H(\theta), \tag{5.2}$$

where N is a normal PDF and H is a half-normal PDF, each scaled by S_1 and S_2 respectively. Figure 5.4 also shows the resultant non-linear model (NLM-Sum in green) together with the normal (NLM-1 in red) and half-normal (NLM-2 in blue) distributions. This appears to provide a closer fit to the observed empirical data. The R function *nls* was used to estimate the parameters which provided the values $s_1 = 0.718$, $\mu = 6.198$, $\sigma = 1.671$, $s_2 = 0.277$, and $\theta = 0.302$.

We conclude that no single standard distribution fit well enough for our purposes and reduce complexity in our algorithm we resort to a custom distribution using the data at hand. We use the density of GPS distances to the nearest road positions, shown in Figure 5.4, to implement the empirical emission function within the map matching software.

5.1.2 Transition Probability

Transition Probabilities in Hidden Markov Models are used to determine the probability of transitioning from one state to another. In our use case, this relates to the probability of a vehicle moving from a candidate position on the BLRN to the next position along the road network whilst abiding by the road regulations for emergency vehicles.

There are several models for determining the transition probability in our context of a map matching problem. These models rely on different data such as sensor data or aspects of the road network. Sensor data can include, but not be limited to, data received from the GPS unit such as speed, heading, linear and angular acceleration, and signal quality data in the form of Horizontal Dilution of Precision (HDoP) and Vertical Dilution of Precision (VDoP). Data supplied by the LAS contains location, speed and heading information which we make use of within our HMM/V model. Quality indicators such as HDoP and VDoP were not available in the source data and so could not be used for the transition function.

Transition Function Evaluation

Through the data available to us and iterative testing we explore the suitability of several transition probability formulae. Recall from Table 2.1 that there have been many attempts at deriving suitable transition functions, each with varying degrees of success. From our experience in PFs we exclude the use of δt_i as the duration between fixes is relatively long at 15 seconds, resulting in large errors. Of the remaining metrics there are four main transition function candidates summarised in Table 5.1.

Table 5.1: Different transition functions used in map-matching algorithms

Reference	Transition function, $a(x_{i-1}, x_i)$
Lou *et al.*[66], Jagadeesh and Srikanthan [70]	$\frac{d_y}{d_r}$
Newson *et al.*[95], Koller *et al.*[98]	$e^{-\frac{1}{\beta}\lvert d_r - d_y \rvert}$
Raymond *et al.*[96]	$e^{-\beta \lvert d_x \rvert}$
Mohamed *et al.*[72]	$e^{-\beta \lvert d_y - d_x \rvert}$

1. $x_i \in X_i$ is a set of candidate road positions at time i,
2. d_x is the Euclidean distance, in metres, between candidates, $||x_{i-1} - x_i||$,
3. d_r is shortest path distance, in metres, between candidates $||x_{i-1}, x_i||_{route}$,
4. d_y is the distance, in metres, between subsequent GPS fixes, $||y_{i-1} - y_i||$,

To evaluate the effects of different transition functions and their ability to effectively map match a route, a test set of tracks were selected from the database by loading all cleansed tracks for January 2016 and extracting journeys comprised of at least 100 fixes. This provided a test set of 459 journeys, each approximately 25 minutes in duration and representative of the most complex journeys that the HMM/V algorithm would be tasked to map match. The test tracks were processed by the HMM/V algorithm to generate estimated routes from the fixes for each transition function listed in Table 5.1. By plotting each route, together with fixes, we found that visual inspection could be used effectively to determine successful map-matching as failed routes were easily identifiable. Map-matching with the inverse exponential of $||d_r - d_y||$ [95, 98] transition function was the most robust, successfully producing routes from the fixes in approximately 80% of cases from a visually inspected sampled set of routes. Other transition functions listed in Table 5.1 had a high failure rate, either producing an erroneous route or failing to produce a route altogether. Successful outcomes were only produced in approximately 25%-50% of the routes we sampled. The reasons behind these failures were generally either a) inappropriate road selection or, b) failure of the HMM/V algorithm to generate a route due to very low transition probability values.

To determine the β parameter of the Newson transition function we use the HMM/V

implementation to quantify the difference between GPS position fix distance and road route distances using our test set of tracks. This was achieved by selecting several β values until a range of working values was established through visual inspection. The test set of 459 tracks were then processed through HMM/V. For each track the HMM/V algorithm generates a sequential set of successful candidate states for each GPS position fix, i.e. candidate road positions that correctly represent the route that the vehicle took. The HMM/V also determines the most likely route taken between those states. For each GPS position fix, y_i, the software calculates the candidate road position, x_i, and the road travel distance in metres, d_r, from the previous candidate, x_{i-1}. Additionally, the Euclidean distance, d_x, in metres between x_{i-1} and x_i is calculated by first converting the latitude and longitude to British National Grid Easting and Northing coordinates (European Petroleum Survey Group (EPSG):27700) and then using Pythagoras theorem to determine the approximate distance in metres. Similarly, the distance between GPS fixes is calculated, d_y. The process generates $19,659$ records, one row for each transition.

The graph in Figure 5.5 is a plot of two histograms converted to density curves, one of Euclidean distances between two sequential fixes and the other of road route distances. The first thing to note is that Euclidean and road route profiles are very similar with road route and Euclidean distances peaking between 150-200 metres, which, with a 15 second interval between reports gives a speed in the of 29.8mph. Most transitions are under 500 metres, with longer transitions occurring where the time interval between GPS fixes is longer. These longer gaps are due to missed or delayed AVLS data. We expect the road route distances to be slightly longer than their equivalent Euclidean distances, and this difference is visible on the curves. For subsequent analysis, outliers were removed by deleting records where the position fix distance, d_y, or road distance, d_x were zero or greater than 600 metres.

Use of Exponential PDF

To explore previous findings [95, 98, 72] that the difference between the GPS position fix and road route distances is exponential we plot a histogram using the data generated from the test set, Figure 5.6, of the difference in metres between the fix distance and road-route distances such that $d = |d_r - d_y|$. For illustration, the histogram is also overlaid with an exponential probability density function with the rate parameter, λ, estimated using the R package $fitdistr$ (the ***fitdistr*** method performs Maximum-likelihood Fitting of Univariate Distributions [149]). The figure shows that the empirical distribution is close to exponential with $\lambda = 0.2085$ metres.

The HMM/V was then trialled by taking a further set of 10 unseen routes, using the transition function

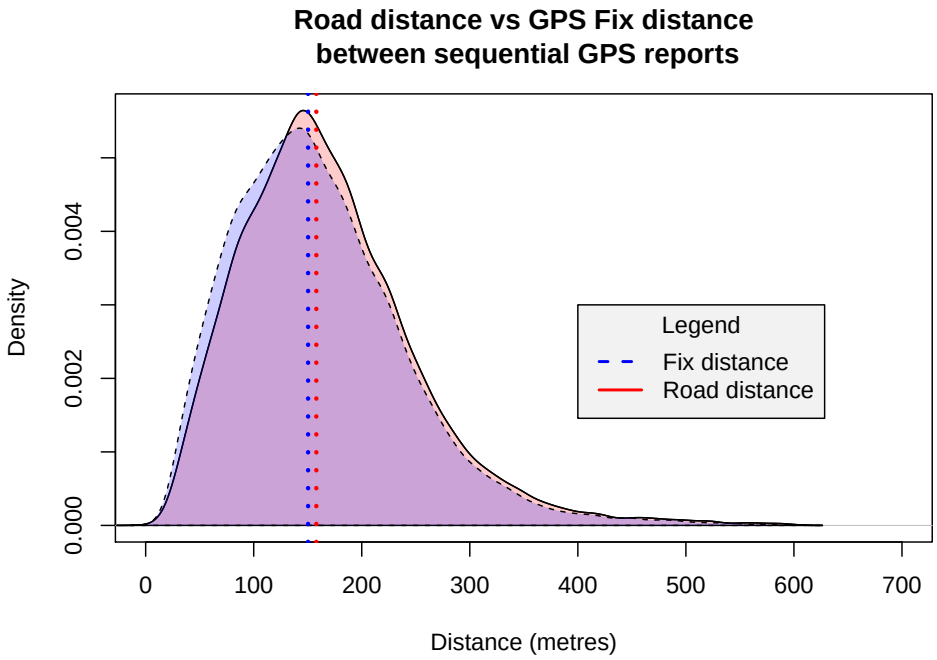


Figure 5.5: Density curves of road and position fix distances between GPS fixes
The peaks correspond to a speed of ≈ 30mph when the time between fixes is 15 seconds.

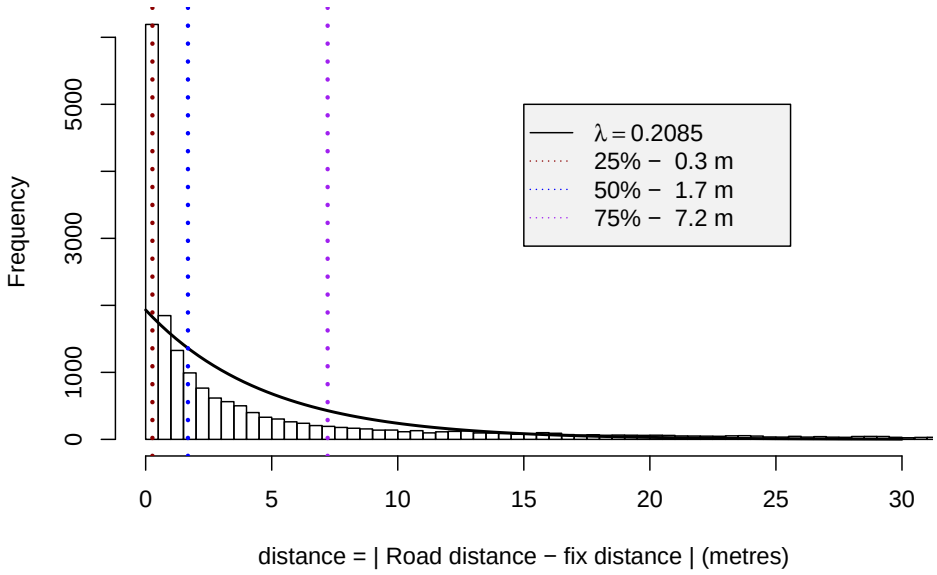


Figure 5.6: Histogram of transition distances between GPS position fix and road-route

Histogram of 19500 samples where each sample is the difference between GPS position fix and road-route distances between two sequential observations, x_t and x_{t+1}. The figure is overlaid with an inverse exponential PDF, where $\lambda = 0.2085$ metres. Also shown are several quantiles. The 99% percentile (not shown) is less than or equal to 221 meters.

$$P(x) = \lambda e^{-\lambda |d_r - d_y|}, \tag{5.3}$$

where d_r is the road route distance, and d_y is the GPS fix distance. Starting with $\lambda = 0.2085$ metres we alter this rate parameter to gauge its effect on route matching accuracy. A large range of values for the parameter λ were used with the HMM/V algorithm (between 0.01 and 10) and the results visually inspected. The HMM/V algorithm appeared to be mainly insensitive to changes in λ, insofar as the route selected for each track did not alter significantly between trials. The main artefacts were that incorrect direction of road were selected. This occasionally occurred on tracks that have a) dual carriageways and b) a direction parameter that is set to 0 (i.e. ignore the GPS

direction value) or c) the GPS position fix is on the wrong carriage way. This causes long loops where the routing engine attempts to route from the incorrect carriageway back to the next GPS position.

Although the transition function does not appear to be a perfect fit for the historic data we trialled, it is sufficiently reliable and better reflects the requirements for our research. This is mainly because this type of transition function is fast to process, has been previously used successfully, gives good results, and is easy to implement. We conclude that the conditional transition probability can be represented using an exponential distribution in Equation 5.3 such that $\lambda = 0.0168$ metres.

5.1.3 Road Geometry Range

The road geometry range parameter, d_{geom}, defines the maximum distance between a GPS position fix and a candidate particle. It is used in the PF when generating candidate particles/states that are constrained to the BLRN and within the HMM/V to select candidate roads. This parameter ensures that candidate particles or road links are within a reasonable distance from the position fix. As shown in Figure 5.4, most fixes lie up to 15m of the nearest road link. In order to generate sufficient candidates, we increase the distance limit such that we obtain a suitable number of candidates such that the algorithms can perform adequately. We set d_{geom} to 50 metres as there is a very small probability (0.0002) that a GPS observation would be more then 50 metres from a road link.

5.2 Road-Constrained Particle Filter

The role of the Road-constrained PF is to estimate the vehicle state using a large set of particles which are updated for each GPS observation. A complete description of the algorithm details can be found in previous research [150, 107, 84, 108] and in our literature review in Section 2.5.3. Our implementation of the PF constrains particles

(vehicle vector estimates) to remain on the known road network. Although this is not true all of the time, as emergency vehicles sometimes attend incidents in opens spaces or other locations that are not recorded in the BLRN. We will ignore this in our study as we are generally only interested in finding road speeds for the journey to scene.

Our implementation of the PF takes a track of GPS fixes and generates an estimate route with associated motion vectors. The filter works sequentially through the list of GPS fixes, processing each position fix in time order. For each GPS position fix at time t the filter maintains i particles, p_t^i, that represent candidate vehicle positions and vector information. Each particle holds a importance weight, w_t^i, and a motion vector, v_t^i, such that each particle state is represented as $p_t^i \doteq (v_t^i, w_t^i)$. The motion vector comprises of the following elements:

1. A coordinate containing an integer Easting and Northing in EPSG:27700 (British National Grid).

2. A velocity vector containing a speed in miles per hour, and a heading in degrees relative to true north.

3. A reference to the current road link and an offset along it.

A parameter n_p determines the number of particles that the algorithm should use at each step. The algorithm requires that a large number of particles are generated in order for it to run successfully. We set this parameter to $n_p = 2000$ as a compromise between performance and precision.

5.2.1 Initialisation

Particles collectively form the posterior distribution of a hidden state, which is the position and motion vector of a vehicle. During initialisation the posterior distribution is unknown, so a random particle cloud constrained to the road network is generated that surrounds the first GPS position fix. At the start of the process each particle is randomly

positioned on a road near the initial observation, y_0. With no prior information this is reasonable approximation of a density field around the hidden estimated position, x_0.

To calculate the initial position of each particle, a set of candidate road links, C_0, are drawn from the BLRN that lie in close proximity to the observation y_0. The candidate road links are obtained in a standard two-step method by first using a Quad tree index [146] to extract all road link geometries within d_{geom} metres of y_0, and then drawing up to n_{cmax} nearest road links to the position fix by calculating the distance from y_0 to the nearest point on the geometry of the road link. This two-step approach is used as an optimisation method such that the number of computationally expensive distance calculations is significantly reduced. The final number of suitable candidate road links for observation y_0 is n_c which may be less than n_{cmax}. For clarification, y_0 is the initial observation, n_{cmax} is the number of nearby candidate road links to find, and n_c is the number of road links that were found in the BLRN.

The process of positioning each particle, p, on the BLRN around the observation, y_0, is as follows. A random number is generated between $[0, n_c]$ and the road link is drawn from C_0. The nearest position on the selected road link geometry from the observation is determined. The nearest position along the geometry is then perturbed by a random amount, along the geometry of the road link, set by a simulation parameter. The speed and heading of the particle are likewise perturbed by an amount set by simulation parameters.

For each position fix, the initial position and next position fix, x_1, are used to produce each particles' heading and speed motion vector. Finally, the weight of all particles are normalised by assigning a probability of $1/n_p$ to each particle.

5.2.2 Move and Perturb

The Move and Perturb step are part of the prediction phase outlined in Section 2.5.3. These steps apply a state transition to the each particle using the particles' motion model. The step is carried out whenever a new position fix is received. We implement

and evaluate two different motion models.

1. **Motion Model 1** - A standard PF moves particles according to their motion vector and motion model rules. A subsequent Prediction stage then calculates the emission probability of each particle based on its distance from the new observation. Our first Move algorithm embodies this principle.

   Particles move along the road network according to their motion vector, taking into consideration road traffic rules that apply to emergency vehicles encoded in the BLRN. Using the time difference from the previous GPS fix and the particle speed we can calculate the distance travelled, d_t^p, which is then perturbed using a uniform random distribution up to d_m metres to produce a final distance to move, d_f. This perturbation, or roughening, is necessary to reduce sample impoverishment [151, 152]. The particle is then moved along the road network from its starting position by d_f metres. The algorithm chooses a random road link using a uniform random distribution if a junction is encountered.

2. **Motion Model 2** - An alternative method was developed that, instead of using the motion vector speed to determine the distance travelled, we instead estimate the speed from the distance between the two fixes. We calculate the Euclidean distance and time difference between the two fixes y_{t-1} and y_t the time period between y_{t-1} and y_t to estimate the distance, d, that the particle would travel. Although this method does not take into account road curvature it produced fewer failures that Motion Model 1 algorithm. The particle then travels along the road network from its starting position by d metres, choosing a random road link using a uniform random distribution if a junction is encountered.

5.2.3 Update

When a new observation is processed, the PF updates the importance weight of each particle based on new sensor information. The importance weight is calculated using

the empirical emission function (Section 5.1.1) using the distance between the observed GPS position fix and the particle.

5.2.4 Resampling

In practical terms the update step can still lead to sample degeneracy where only a few particles have a significant weight and others a low weight. The effect is that the number of *effective* particles is reduced [151]. This condition can be detected when the estimate of effective samples drops below a given threshold, n_{cutoff}. The effective sample size can be calculated using the equation [153, 84]

$$\hat{N_{eff}} = \frac{1}{\sum_{i=1}^{n_p}(w_t^i)^2} \tag{5.4}$$

When the effective number of samples drops below a given threshold the algorithm draws the n_{sampl} percent highest probability samples to create a new set of particles. Particles weights are then normalised by assigning each new particle a weight of $1/n_p$. Unfortunately this procedure can lead to another problem called sample impoverishment. During re-sampling it is likely that particles with a high importance weight will be re-sampled many times. In severe cases this could lead to all particles being identical having the same position and weight [109]. This can be avoided in a number of ways including the use of roughening and kernel density functions to estimate the probability distribution [151].

Our implementation of re-sampling, see Listing 1 in the Appendix A, first creates a numeric array that represents a Cumulative Distribution Function (CDF) using the top n_{sampl} % of particles by ordered by weight. A set of n_p particles are then generated as follows. For each new particle, a weight drawn randomly from the CDF. If a particle with the same weight already exists then the new particles' vector is perturbed slightly. This perturbation prevents the algorithm from adding duplicate particles and maintains population diversity.

5.2.5 Results

The particle filter was trialled extensively with many variations in the parameters. Table 5.2 lists the final parameters chosen to test the filter. For Motion Model 1 we found through iterative adjustment that a value of 20 for the tuning parameter, d_m, worked well in most cases but on occasion found particles that were often dispersed quite a distance away from the new observation. This resulted in a rapid decline in the number of effective particles (as they are eliminated from the pool if their importance value decreases below a given threshold) and eventual failure of the PF.

Table 5.2: Particle Filter Map-Matching Parameters

Parameter	Symbol	Value
Particle perturbation in metres	d_m	20
Number of particles	n_p	2000
Maximum number of candidates	n_{cmax}	100
Re-sample cutoff	n_{cutoff}	0.9
Re-sample percent	n_{sampl}	80%
Road Geometry Range	d_{geom}	50

In practice, the filter using Motion Model 2 appears to work for GPS intervals up to about 15 seconds. However, if time periods are longer or the road network is particularly complex then the particles diverge across the network to the point where there are no particles at or near the later position fix. This is in line with previous research [110, 108] that found that a PF at low sampling rates (about 15 seconds) performed poorly, explained by the fact that low sample rates largely increase the number of possible paths that a vehicle can take. The performance of the PF is also affected by uncertainty of the vehicle position on the road affected by a) GPS position accuracy which can be poor in urban environments and b) the road link is described by the centreline of the road rather than the physical location of the carriageway [108]. We conclude that the PF in its current state is unsuitable for determining routes due to the dense urban environment, long GPS fix intervals greater than 15 seconds and the

fidelity of the centreline-encoded digital road network.

5.3 Map Matching with HMM/V

Recall from the literature review in Section 2.5.3 that the HMM/V algorithm is a statistical approach to both real-time and offline map matching [95]. We use the HMM/V to determine the most likely route on the BLRN, using GPS fixes as our observations. Estimate travel speeds along the route can then be extracted.

The algorithm is split into two stages. In the first stage a trellis HMM is generated, refer to Figure 2.1. For each GPS observation a number of candidate road positions are selected and emission values calculated. The transition probabilities for each combination of candidate road positions are also calculated. The second stage uses the Viterbi algorithm [101] to determine the most likely sequence of road segments from the transition and emission probabilities.

5.3.1 Building the HMM

To build the HMM a number of candidates road positions, $x_{0,t}, \ldots, x_{k,t}$, selected for each GPS position fix, y_t. To find suitable candidates we employ the same technique used in the particle filter initialisation step, i.e. a set of candidate road links are drawn from the BLRN that lie approximately d_{geom} metres of the observation using a Quad tree index. Road links are removed from the set whose relative direction of travel compared to the candidate fix is greater than b^f_{max}. The nearest n_{cmax} road links are then drawn from this set. Finally the closest point on each candidate road link geometry is used as a candidate position. In some circumstances no suitable candidates could be found for a given observation. Whilst our data suggests that this is an infrequent event with an occurrence of approximately 0.1%, the entire route is discarded as a path through the HMM is not possible.

As can be seen in Figure 5.4, the distance between GPS position fix and the nearest

road link is almost always less than 15 metres, so the algorithm should always have at least one candidate. The histogram in Figure 5.7 shows the frequency of road link geometries within 50 metres for a sample set of 1000 GPS fixes.

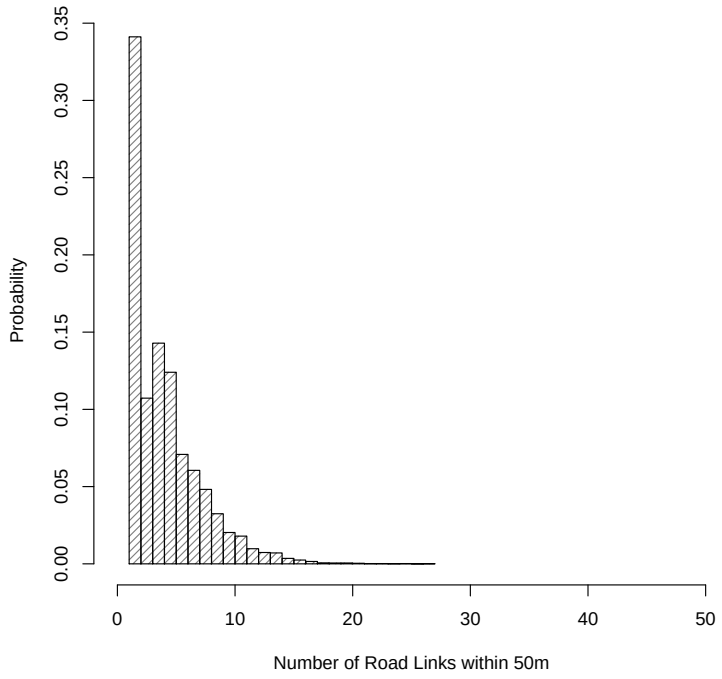


Figure 5.7: Probabilities of obtaining a given number of road links within 50m of a GPS position fix
Generated from a random sample of 1000 GPS fixes and counting the number of road link geometries within a 50 metre radius.

The state information for each candidate road position contains the road link details, the offset along the road link that is nearest to the fix, and an emission probability. We assign an emission probability based on a lookup function, described in Section 5.1.1, using the straight-line distance from the candidate position to the GPS fix. Transition probabilities are also calculated between candidates using the transition function detailed in Equation (5.3).

5.3.2 The Viterbi Algorithm

Once the model has been constructed with candidate road positions with associated transition and emission values the Viterbi algorithm [101] is used to find the most likely sequence of hidden states through the HMM, and in our case, the most likely road route that the vehicle travelled along. With reference to Section 2.5.3, the Viterbi path is the most likely state sequence $x_0, \ldots, x_T$, given the observations, $y_0, \ldots, y_T$, initial probabilities π_i of being in state i and transition probabilities $a_{i,j}$ of transitioning between state i and j.

Viterbi values are calculated for each candidate $x_{k,t}$ as the product of its transition and emission values and the previous candidate,$x_{k,t_{-1}}$, whose Viterbi value was the greatest at that time step. When all fixes have been processed the most likely complete route is extracted by concatenating routes between candidates with the maximum Viterbi value. Our implementation of HMM/V also generates inferred speeds for each road link traversed using the timestamps of the GPS fixes. For each record we store the speed, the road link and time of day in the database for later analysis.

5.4 Map-Matching Validation

Validation of the map-matching process without a known ground truth to compare the results with is a problem often overcome with the use of synthetic routes where the production of these routes can be achieved using a routing engine or other simulation methods. The process is as follows: a synthetic route is produced between two locations, GPS fixes are then sampled along the route and passed as input into the map-matching algorithm. The estimated route generated by the map-matcher is then compared with the original synthetic route, resulting in the production of a similarity metric.

The value of validation through the use of synthetic routes is often contentious as the synthetic routes are generally approximations to their real counterpart. Nonetheless, comparison of the map-match output with synthetic routes can be used as an indicator

of robustness of the implementation. Additionally, the algorithms sensitivity to different GPS sample rates can also be quantified.

We use our routing engine to generate synthetic routes using the start and end locations of 1,000 historic emergency vehicle journeys. The routes were calculated using the road speed model Metric II described later in Section 6.8 (this model represents the current routing engine in use at LAS). To validate the HMM/V map-matching algorithm we generate sets of test GPS fixes of varying intervals and precision along these routes. This is achieved in a two-step process. Sets of *ground truth* fixes were created for each synthetic route by sampling along the route at journey time intervals of 10, 15 and 20 seconds. Then, sets of *test* fixes were obtained from the ground truth fixes by selecting a random location within a set distance of the original. We varied the error between 0 (no error) and 30 metres in increments of 5 metres. The test fixes were then processed through the map-matcher and estimate routes compared with their corresponding synthetic route.

The error between synthetic and estimate routes is often measured using the difference in coincident route length [95, 96, 98]. This is a measure of geometric accuracy of the route known as the Incorrect Road Length Fraction (IRLF) and is defined as:

$$\frac{d_{miss} + d_{err}}{d_{route}} \tag{5.5}$$

where d_{route} is the length of the synthetic route, d_{miss} is the length of all road links which were missed during map matching and d_{err} is the length of all road links which were erroneously found during map matching.

Figure 5.8 shows the IRLF for the 1,000 synthetic routes showing 95% confident intervals (bars) and 10/90th percentile (whiskers). The HMM/V algorithm performs well, demonstrating approximately 7% mismatches for 15s intervals and up to 10m GPS inaccuracy. As previously observed [95], where GPS accuracy is poor MM accuracy appears to improve when GPS observation frequency is decreased. We assume this is

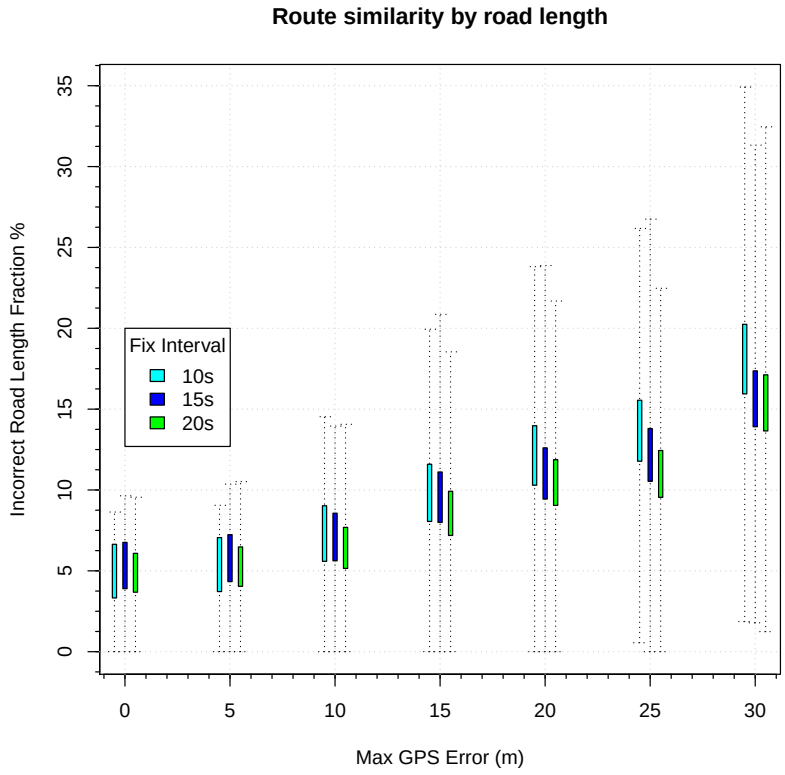


Figure 5.8: IRLF with varying GPS precision and GPS Fix Interval

The fraction is calculated as $\frac{d_{miss}+d_{err}}{d_{route}}$ where d_{route} is the length of the synthetic route, d_{miss} is the length of all road links which were missed during map matching and d_{err} is the length of all road links which were erroneously found during map matching. The HMM/V algorithm performs well, demonstrating approximately 7% mismatched route length for 15s intervals and up to 10m GPS inaccuracy.

due to the fact that a higher number of inaccurate fixes for any given trip (due to the increase GPS reporting frequency) leads to greater instability of route prediction; for example a long route will have half the number of inaccurate 20s-gap fixes than 10s-gap fixes and therefore a greater probability of selecting an matched route.

The expected GPS error in our urban environment is likely to be within 10m (Figure 5.4) and at this level of perturbation there is little difference in the accuracy of map-matching for position fix intervals ranging between 10 and 20 seconds. Finally, for synthetic routes where the GPS interval is 15 seconds and GPS perturbation is 10

metres the median percentage of mismatched road links is 1.93% and the median of the incorrectly identified roads is 4.16%.

5.5 Map-Matching Results

A total of 1,910,941 journeys from a total of 2,311,661 (82.7%) were successfully processed by the HMM-Viterbi algorithm using parameters in Table 5.3. As expected, some routes could not be matched correctly because of corrupt or inaccurate GPS data. Many routes were also unprocessed because of insufficient fixes to form a track. This typically happens when an emergency vehicle is dispatched to an incident and then quickly stood down for operational reasons. The map-matching process produced 177,975,172 road speed records. Each record consists of the road link id, time, date and estimated speed. The process took approximately 2 days to run on a 6-core i7 processor.

Table 5.3: HMM/V Map-Matching Parameters

Final parameter values used for map-matching approximately 2 million ambulance journeys.

Parameter	Symbol	Value
Maximum relative angle	b^f_{max}	120
Maximum number of candidates	n_{cmax}	100
Transition function parameter	λ_a	0.0168
Road Geometry Range	d_{geom}	50
Minimum time between fixes	t^f_{min}	10
Number of fixes to skip	n^f_{skip}	2
Number of fixes to process	n^f_{take}	9999
Minimum distance between fixes	d^f_{min}	25
Maximum speed between fixes	v^f_{max}	80

Analysis

For the purposes of our research we intend to build speed models that can be used to predict journey times across the whole of London. An emergency journey can start and stop in almost any location and any road type within London. It is therefore important that there is sufficient spatial coverage of speed data across on all road types within

the road network such that robust speed models that can be developed that cover the entire London Area.

The initial analysis of road coverage was by visual inspection. The urban area in the south-east of London which includes Croydon and Orpington is on the limits of the LAS operational area. Its road network is shown in Figure 5.9. Green lines are road links, overlaid by purple tracks indicating an emergency vehicle has used this route. There are some minor roads that are not covered by the matching process, however the vast majority of the road network is covered.

Table 5.4: Coverage of road types by HMM/V and GPS snapping

For each road type the table lists a) the number of road links in the road network , b) the number of road links found by map-matched, c) the percentage of the BLRN covered, and d) the percentage covered using GPS snapping. The percentage coverage between GPS and HMM/V are not directly comparable as GPS fixes are snapped to a road without regard to direction of travel, whereas HMM/V road links are split into two for each direction of travel.

Road type	**Links**	**Used**	**HMM/V %**	**GPS %**
A Road	65857	62124	94%	78%
Alley	46473	15579	32%	34%
B Road	21747	18257	84%	64%
Local Street	327827	234624	69%	50%
Minor Road	62052	48853	78%	59%
Motorway	1131	689	61%	90%
Pedestrianised Street	363	317	87%	70%
Private Road- Publicly	6735	3945	56%	55%
Private Road- Restricted	75360	34151	7%	45%
Total	**607545**	**418539**	**69%**	**65%**

Further analysis of coverage by road types is shown in Table 5.4. This shows for each road type the number and percentage of road links covered using HMM/V compared with naïve road snapping using only the uncorrected GPS position fix. The percentage coverages between GPS and HMM/V are not directly comparable as GPS fixes are snapped only to a single road link, whereas HMM/V these road links are split into two, one for each direction of travel. Most of the main road types see a good coverage. The major A roads are very well covered, at 94%, whereas some local roads and especially private roads are not covered particularly well. Three screen-shots are included in

Appendix B that display a map of the map-matching results.

Figure 5.9: Road usage density map obtained from GPS snapping
The urban area of Croydon and Orpington are on the limits of the LAS operational area in the south-east of London. Green lines are roads not traversed by an LAS emergency vehicle for the sample period.

6

Vehicle Speed Models

This chapter begins with a description of our implementation of the BLRN routing engine that can be used as a platform for estimating routes and travel time for AEU and FRU vehicles in London. The engine is dependent on speed models and, as we will see in the next chapter, significantly influences the route and predicted arrival times. For our investigation into routing accuracy and journey time estimation we devise several experimental road speed models of varying complexity.

6.1 Routing Engine Implementation

Our objective is to create a BLRN routing engine that can accurately calculate the quickest route given the start and end location of the journey, the time of day and vehicle type.

Each routing algorithm has its own merits and limitations, in our research we will only use Dijkstra's algorithm as we do not require features such as being able to handle negative weights or heuristic speed optimisations that others provide. Regardless of which algorithm is selected, all require weighted edges that represent the cost of travelling along that egde. The very nature of the physical road network also dictates that we use a directed network.

The purpose of the routing engine is to calculate the set of navigable, i.e. linked, edges that represent the shortest path from one node to another on the network. In Vehicle Routing Problems (VRP) [154] the *shortest path* can be taken to mean the shortest distance or quickest time. For our problem of emergency vehicle routing we are interested only in the quickest time. To this end, the edge weights associated with the BLRN are representative of transit time, rather than distance.

We use Dijkstra's shortest path algorithm to compute journey times using the BLRN. In the standard implementation of this algorithm the estimated travel time for a given journey is calculated as the minimum time taken to travel from one point on the network to another point on the network, using the sum of transit time estimates for each road link edge traversed.

In our implementation of the algorithm we can specify an edge cost function which is responsible for determining the likely duration of a specific road link, given the hour of week, the vehicle type and the road link. An abbreviated pseudo code listing for our implementation of Dijkstra's algorithm is shown in Algorithm 2 in Appendix C.

There are two main complexities in the routing engine that are not detailed in the pseudo code. These are to do with a) changes in elevation such as bridge and flyovers

and, b) dealing with the start and end locations which lie part way down a road link. The routing engine ensures that adjacent outgoing links from a candidate edge must have a starting grade equal to the end grade of the candidate edge.

section on level grades moved to Datasets chapter

6.1.1 Coordinate to Road Link

The first step in calculating a route, given the start and end geographic coordinates, is to determine the road links that these relate to. With reference to the start location in Figure 6.1, indicated by the green star, the nearest road link can be readily identified by the shortest distance between the coordinate and nearby road geometries. In this case a candidate road link $A \rightarrow B$ has been identified. Similarly, for the end coordinate, the road link $C \rightarrow D$ has been identified.

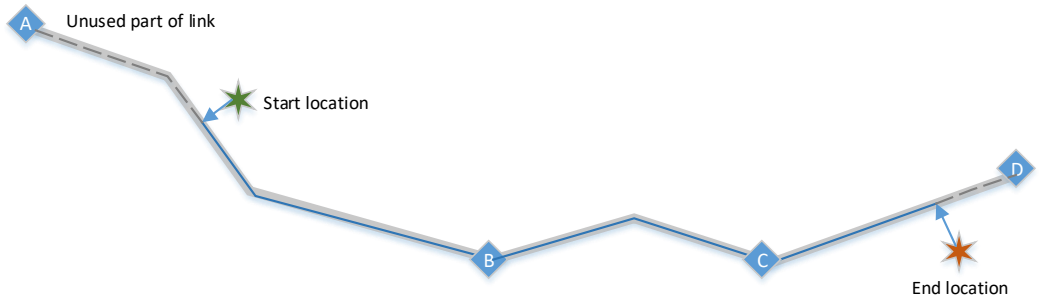


Figure 6.1: Assigning road link + offset from a coordinate

It can be clearly seen that the distance travelled along the road links is shorter than the sum of the road links used. For accurate travel time estimates, the truncated start and end road links must be taken into consideration. We estimate the travel time to be directly proportional to the amount of start and end road link used.

6.2 Calculating Edge Costs

In the context of a routing engine, edge costs relate to vehicle transit times along road segments and the optimal route is the quickest route from A to B within the network. In reality the actual transit time along any particular road network is dependent on prevailing conditions such as road layout, traffic, road and weather conditions, as well as the vehicle in use and driver skill. The total time through the network is also potentially influenced by other factors such as delays in navigating junctions.

Dijkstra's algorithm requires that all edges have a positive edge cost associated with them. As we are attempting to find the quickest time for any given route it follows that edge costs represent the transit time for each edge. We initially develop five edge cost models (hereafter, Metric I..V). We employ a 'plug-in' mechanism such that the routing engine can be configured to use any one of our Metric models when calculating a route. To this end, each Metric must implement an *CalculateEdgeCost* function that, given several parameters regarding the current road being traversed, returns a road vector containing road length, estimated duration and speed. Every implementation of a Metric must take the form,

```
RoadVector CalculateEdgeCost(string vehicletype, int hourOfWeek,
    RoadLinkEdge edge).
```

Specifically, the *CalculateEdgeCost* method is passed a road link, the vehicle type (AEU or FRU), and the hour-of-week in which the route is being estimated.

We devise and compare several edge cost functions of varying complexity for use with Dijkstra's shortest path algorithm to compute journey times. The first, *Metric I*, is based on a simple fixed speed for all road types. The second, *Metric II*, is based on the standard speed profile used by the LAS which uses a different fixed speed for each road type. Metric III & IV also take into consideration spatio-temporal factors at different resolutions, with Metric V using the road link speed information directly. In all cases below, no route speed data produced by the map matching process was used

for the month of November 2016 so that routes in that month could be used for unseen route analysis.

6.3 Metric I - Constant Speed

This naïve edge cost calculator assumes a constant speed regardless of the area, road link, vehicle type or time of day. A constant speed of 22.8mph (13.2m/s) is selected as this is the average speed across all road links. This is obtained as an average of all road speed records mined from HMM/V map matching. The advantage of this edge cost calculator is that it requires very little processing time, resulting in fast route calculations.

6.4 Metric II - Fix Speed by Road Type

The implementation of the LAS routing engine uses a relatively simple speed profile based on a constant road speeds for different road types. No account is taken for the time of day. In addition, the road speed model adds a 'junction' delay of 2.5 seconds each time a BLRN road link is traversed. Unfortunately this is not always indicative of a real junction on the road network as, for example, nodes are inserted into the network to split up long sections of road, and at every intersection for each side road joining the main carriageway. So a vehicle travelling down a single road passing many side roads would accumulate many additional 2.5 second delays.

Table 6.1 contains the road speeds used by LAS for routing purposes. We also compare this with the average speeds obtained from the GPS and from HMM/V. As with the Constant Speed function, this Metric requires little processing time.

Table 6.1: Average road speeds by road type

Road speeds by road type a) average speeds from map-matching b) average speeds from GPS data c) values used by LAS in their routing engine. All speeds are in mph.

Road type	HMM/V Speed	GPS Speed	LAS Speed
A Road	31.21	20.74	29
Alley	22.51	14.61	3
B Road	28.01	18.01	24
Local Street	18.79	10.89	14
Minor Road	26.80	16.69	19
Motorway	42.65	38.79	35
Pedestrianised Street	16.29	8.12	2
Private Road- Publicly	16.58	8.14	5
Private Road- Restricted	18.50	10.55	5

6.5 Metric III - Grid + Hour-of-Day + Vehicle

To calculate edge costs for each road link, a road speed model based on a multidimensional array of real numbers, $\mathbf{M} \in \mathbb{R}^{x,y,v,r,h_d}$, representing average speeds is constructed. The matrix is populated with the harmonic mean road speeds by hour of day, h_d, a grid cell in which the road link lies, x, y, vehicle type, v, and road type, r. Each grid cell is defined as an area $500m \times 500m$, of which there are 6,636 covering the London area.

The matrix is populated by reducing speed data drawn from the road link speed records from Table 5.4. This reduction is obtained by calculating the harmonic mean of all the road speed records created during the map-matching process and then grouping by hour of day, vehicle type, road type and grid cell. This process produces 703,886 items in the 5-dimensional array. A dense matrix would have 2,866,752 records, resulting in a sparse matrix density of 24.55%.

6.5.1 From Sparse to Dense

The matrix must be converted from a sparse matrix to a dense matrix for it to be usable by the routing engine. Missing values must therefore be predicted. We devise a method to populate missing values in the matrix by looping through v, r, h, x, y, and calculating the mean speed by locating nearest non-zero neighbours spatially across x, y.

During routing engine start-up, the matrix is loaded into an in-memory multidimensional array. The array is scanned, looking for cells with no assigned speed value. When a missing speed value is detected, nearby speed values are used to estimate a value for this cell. The estimate is calculated by calculating the harmonic mean from neighbouring cells. The algorithm starts with the nearest neighbouring cells and progressively moves out until a non-zero speed is found.

Estimates and pre-existing values are merged into a new 5-dimensional array. This ensures that estimates are not inadvertently used to calculate other estimates. When complete, the 5-dimensional array is dense and contains a usable speed value for every combination of hour, road type, vehicle type and geographic grid cell.

6.5.2 Calculating an Edge Cost

Edge costs are computed by indexing into the array of average speeds, $\mathbf{M}_{x,y,v,r,h_d}$. This is performed using these steps:

1. Calculate the grid cell, x, y using the supplied coordinate.
2. Calculate the hour-of-day, h_d, by taking the modulus 24 of the $hourOfWeek$ parameter.
3. Look up the speed in the matrix using x, y, h_d, r and v.

A vector is then calculated using the estimated speed and road length to calculate the link duration. The vector is then returned to the routing engine.

6.6 Metric IV - Grid + Hour-of-Week + Vehicle

This is almost identical to Algorithm III except that we use hour-of-week instead of hour-of-day, such that a 5-dimensional matrix, $\mathbf{M}_{x,y,v,r,h_w}$ containing average road speeds is constructed such that h_w is the hour of the week. In this case, because there are 164

hours instead of 24, a dense matrix would require 20,067,264 values. From our data we produce 3,247,386 records reducing the matrix density to 16.18%.

The Metric follows the same process of scanning the matrix for empty cells and estimating new values in order to make the matrix dense.

6.6.1 Calculating an Edge Cost

Edge costs are computed by indexing into the array of average speeds, $\mathbf{M}_{x,y,v,r,h_w}$. This is performed using these steps:

1. Calculate the grid cell, x, y using the supplied coordinate.
2. Look up the speed in the matrix using x, y, h_w, r and v.

A vector is then calculated and returned to the routing engine.

6.7 Metric V - Road Link Speed Data

It is possible to augment the speed information from historic GPS fixes by utilizing the HMM/V map-matching output. This is because the map-matching algorithm infers a complete route with estimated speeds using sparse GPS location fixes and knowledge of the BLRN.

Metric V estimates the road speeds for a given road link by computing the harmonic mean of historic road speeds for that link for any given hour of the week. The historic road speeds are sourced from the HMM/V map-matching algorithm previously described in Section 5.3. To prepare the data, map-matched road speed records are grouped by road link, vehicle type and hour-of-week and the harmonic mean speed calculated for each group. From this, a 3-dimensional matrix of real numbers, $\mathbf{M} \in \mathbb{R}^{l,v,h_w}$, representing harmonic average speeds is constructed where l is the road link identifier, h_w is the hour-of-week, and v is the vehicle type.

6.7.1 Calculating an Edge Cost

This Metric attempts to find a speed for the road link, l, at time h_w and vehicle type, v. If no records can be found then the algorithm reverts to the 5-dimensional matrix look-up defined in 6.6. If records can be found for the same hour and vehicle type then the harmonic mean of the road speed records is returned.

6.8 Evaluating Road Speed Metrics

For each prospective Metric, summarised in Table 6.2, we wish to gain insight into its accuracy and precision to predict emergency vehicle journey times when used in conjunction with the routing engine. We also investigate how routes taken by emergency vehicles differ from routes produced by the routing engine.

Table 6.2: Road Speed Metrics

Metric	Description
Metric I	Constant Speed, regardless of road type, hour of day etc.
Metric II	Fix Speed by Road Type.
Metric III	Spatial matrix + Hour-of-Day + Vehicle type.
Metric IV	Spatial matrix + Hour-of-Week + Vehicle type.
Metric V	Road Link Speed Data mined from map-matching.

To evaluate our five Metrics we devise several experiments that test the temporal and spatial accuracy of the road speed models and routing engine. By utilising historic journeys we can compare actual vs. predicted routes and journey times, quantifying the accuracy and precision of each Metric. For the test data in our experiments we use a trial set of 69,487 routes from the month of November 2016, extracted from the entire incident route set of 2,311,661. Note that these routes were not used to build the edge cost models and were therefore unseen. The experiment is repeated for all the Metrics described in Section 6.2. For each route in our test cohort we perform several experiments as follows:

1. Estimate journey time using an estimate of the actual route that the emergency vehicle took.

2. Use the routing engine to predict the journey time of historic emergency vehicle journeys using only start and end locations.

3. Calculate route similarity for each estimated route and each Metric.

4. Calculate additional route metrics that guide how corrections might be applied to the estimated route.

6.8.1 Calculating Journey Estimate from Actual Route

This experiment produces an estimated journey time for a known (seen) route. The method is independent of the routing engine and tests a Metrics' accuracy of prediction for a given route.

We use output from the map matching process to obtain a list of actual road links traversed for each journey, effectively the map matching process provides us with estimated ground truth for the journey undertaken. This method allows us to examine differences between the estimated time and the actual time, ensuring the routes used are identical. For a given historic route we are able to calculate an estimated duration, t^m_β, using Metric, m, as the sum duration of each road link traversed in the route plus any junction delay. The journey time $t^m_\beta(X)$ is defined by

$$t^m_\beta = \sum_{x \in X} f^m(x, h, v), \tag{6.1}$$

where $x \in X$ are road links in the route X, and $f^m(x, h, v)$ returns the expected duration along road link x at hour h for vehicle type v. The prediction error for a seen route is defined as $\delta t^m_\beta = t^m_\beta - t$, where t is the duration of the historic journey. In this case, positive values represent over-estimation and negative values under-estimation of journey time. The process is repeated for each Metric and historic route in the test

cohort and results captured. An outline of the process is depicted in Figure 6.2.

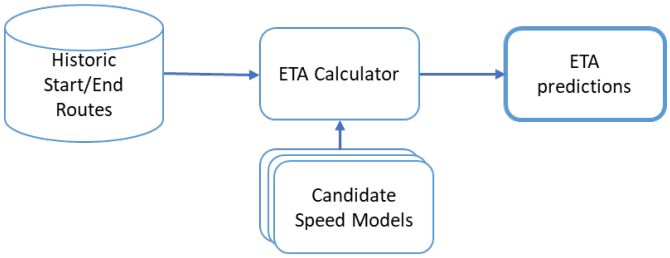


Figure 6.2: Seen route analysis
Historic routes are evaluated by each speed model using estimated route.

6.8.2 Estimating Journey Route and ETA

This experiment uses our routing engine and a given road speed Metric to estimate a quickest route and duration, given only the time of day, start and end locations. This method directly reflects how the routing engine might be used within a simulator or operational environment, where the path that the emergency vehicle will take is unknown. Consequently the route selected from the routing engine may differ from the actual route taken.

In the case of an unseen route we calculate an estimated duration, t_α^m, using the routing engine with each Metric using only source and destination locations of the historic route. The prediction error for the unseen route is defined as $\delta t_\alpha^m = t_\alpha^m - t$. Different aspects of the routing accuracy such as spatial, temporal, route and ETA variance are also examined. An outline of the process is depicted in Figure 6.3. The process is repeated for each Metric and historic route in the test cohort and results captured.

6.8.3 Actual vs Estimate Route Comparison

In addition to the above methods we also compare the spatial similarity of the original trip route to the route generated by the routing engine. We also calculate route similar-

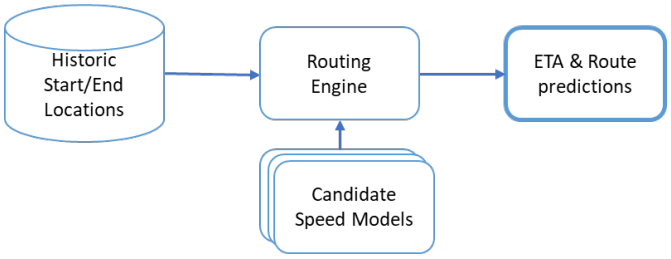


Figure 6.3: Unseen route analysis
Historic routes are evaluated by each speed model using start and end locations.

ity for portions of the journey by dividing the estimated route into four equal portions by length and determining the percentage of coincident distance on the original route determined using map-matching.

6.9 Estimate Route Turn Analysis

Finally, we calculate additional route metrics that guide how corrections might be applied to the estimated route (discussed further in Chapter 7.5). These metrics include the number of roads traversed, the number of large angle changes and the number of road links traversed.

7

Metric Evaluation

The evaluation of Metrics I - V is designed to characterise how these models predict ETA and the journey path taken by AEU and FRU vehicles travelling under BLS conditions. The evaluation is divided into two categories, *seen* and *unseen* routes. Journey time estimation errors are calculated for each category and Metric. We also compare route paths between the two categories and describe route spatio-temporal characteristics. Finally, we offer corrective factors to improve ETA accuracy.

7.1 Seen and Unseen routes

Recall from our previous map-matching experiments that we have generated a large dataset of historic ambulance journeys. We conduct several *seen* route experiments that estimate travel times using the paths from those historic journeys.

For *unseen* route experiments, we estimate both the path and duration, given only the start and end locations. This scenario is the one that faces the control room at LAS where the location of ambulance and patient are known but the route to be taken by the ambulance is unknown. Our experiments with unseen routes are designed to estimate both the duration and route that would be taken using the routing engine and a selected road speed model (see Section 6.8.2).

7.2 Prediction Results

In total 69,487 historic journeys during November 2016 were analysed. With reference to the density plots in Figure 7.1, we calculate the Unseen route prediction error, δt_{α}^{m} for all journeys and each Metric (top figure), and the Actual route prediction error (bottom figure) that assumes the route is known, δt_{β}^{m} .

In both charts, Metric III, IV & V have prediction error modes between 0 and +25 seconds. The accuracy of predictions can be quantified by calculating the bias for each metric. Specifically, we observe good accuracy for unseen routes using Metric IV, Metric V and Metric III with a bias of just 0.4, 8.2 and -14.5 seconds respectively. Metric I has a bias nearly a minute off centre at 58.5 seconds. Metric II has the greatest bias and performs poorly compared with all other Metrics with a value of over 3 minutes (181.1 seconds).

For routing engine prediction error, Metrics III and IV that take into account speed variations due to consideration for hour of day/week and area of London, have similar standard deviations of 118.1 and 119.5 seconds respectively. Metric V provides the best precision with an estimated standard deviation of 97.5 seconds. To test the robustness of

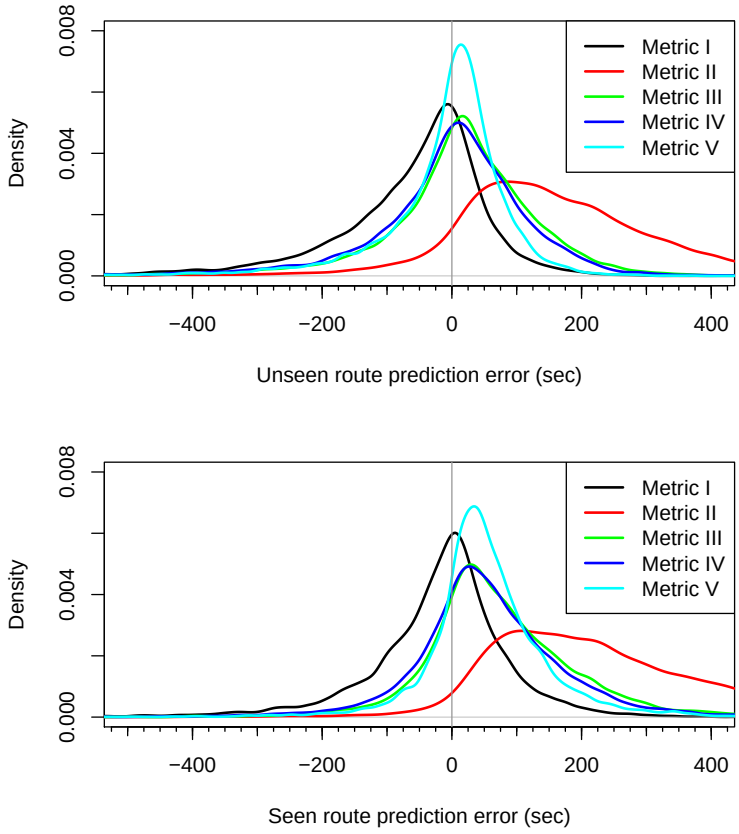


Figure 7.1: Analysis of 69,487 routes showing the density of prediction error

Analysis of routes showing the density of prediction errors for both the (a) Unseen route using a route predicted by the routing engine, t_{α}^{m}, and (b) Seen journey, t_{β}^{m}, using the actual path taken by an ambulance, across all Metrics, m. The curves show that the error obtained when predicting journey times from an unseen route is similar to that obtained when the route is known. Metric II significantly overestimates journey time in most cases.

these calculations we also calculate the 95% confidence intervals of the estimate standard deviations. As the distributions of the predictions errors cannot be guaranteed to be normally distributed we use a non-parametric bootstrap resampling method, using R's boot::boot method with sample size set to 1000, to calculate the confidence intervals. The results of standard deviation confidence intervals are plotted in Figure 7.2. Metric V confidence intervals do not overlap any others and are clearly lower than them. This leads us to conclude that Metric V's precision is demonstrably better than other Metrics presented here.

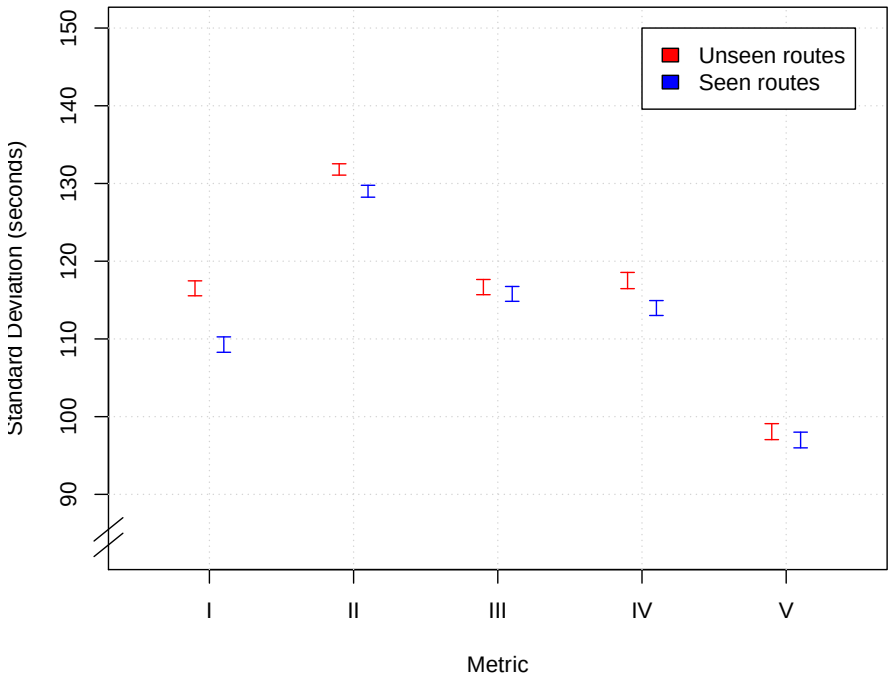


Figure 7.2: Metrics I..V 95% confidence intervals for Standard Deviations of Estimation Errors.

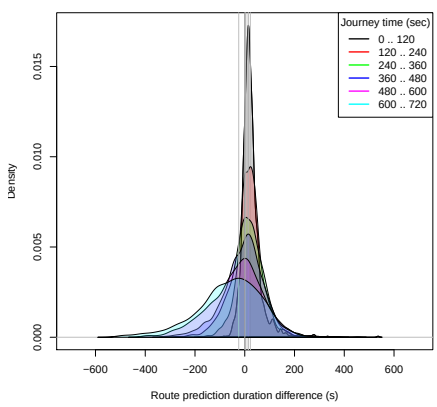
Metric V confidence intervals do not overlap any others and are clearly lower than them.

Our data suggests that using historic speeds captured from reconstructed routes can play a significant role in ETA prediction and can be further exploited to improve navigation planning. We investigate the causes and implications relating to the performance of this approach in the next section. The similarity of actual and estimated density plots suggests that the predictions generated by the routing engine are generally comparable regardless of the actual route taken, and that possibly the predicted route is similar to the actual route.

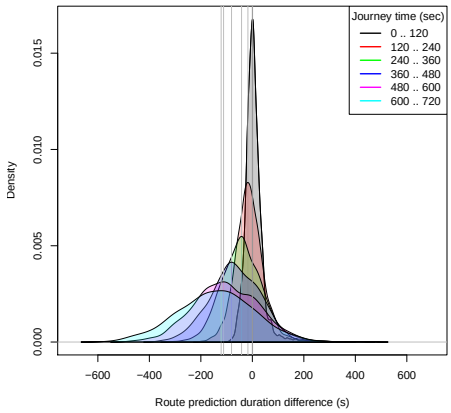
7.2.1 Error by Journey Time

To further characterise the error for unseen routes, i.e. those routes generated by the routing engine, we divide the trips by their journey time. Specifically, we split journeys by unseen route predicted journey time into six groups by 120 second intervals up to

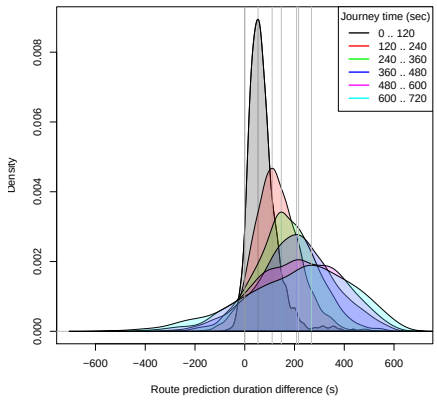
720 seconds and plot the error distribution.



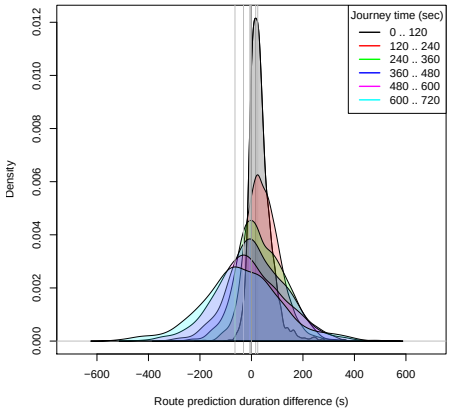
(a) Metric V



(b) Metric I



(c) Metric II



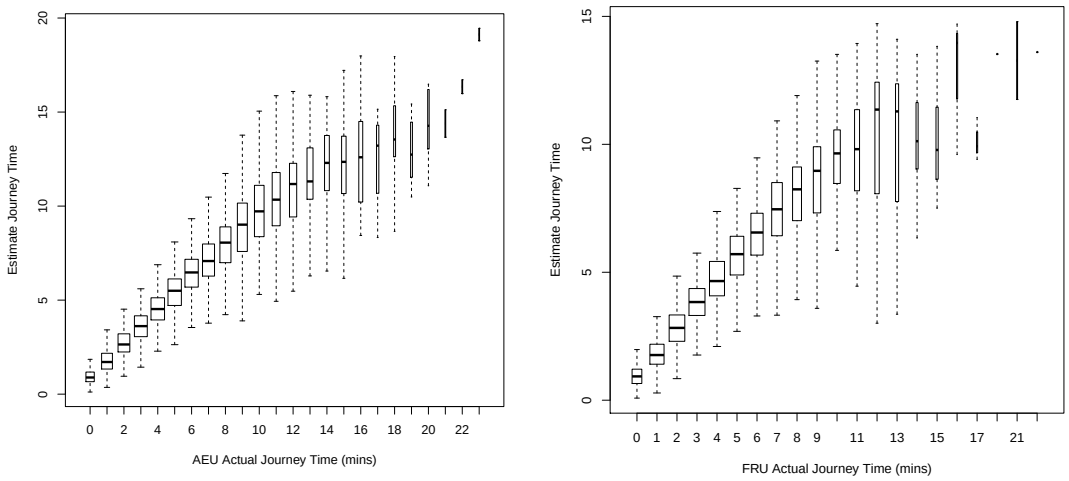
(d) Metric III

Figure 7.3: Prediction error density plots for a) Metric V, b) Metric I, c) Metric II and d) Metric III methods

Figure 4 through Figure 7 in Appendix D are shown together here in thumbnail form in Figure 7.3. These are plots for Metric V (a), Metric I (b), Metric II (c) and (d) Metric III respectively. This series of charts shows clearly that, for all Metrics, the amount of error increases with the trip length. The rate of change of error is more pronounced in the Metric II chart. It is also clear that in all cases, the Metric II significantly overestimates journey time for long trips.

7.2.2 Error by Vehicle Type

Again, using estimates produced by the routing engine for unseen routes, Figure 7.4 plots estimated (a) AEU and (b) FRU journey duration, using Metric V, against the actual duration in minutes. AEU estimated journey time is nearing a linear relationship up to approximately 14 minutes and thereafter the actual journey times appear to be longer that the estimate. The FRU plot (b) shows a similar stability to the AEU estimates, appearing stable up to approximately 8-10 minutes. The bar thickness indicates the number of sample journeys and this highlights that durations longer than 15 minutes are infrequent.



(a) AEU **(b) FRU**

Figure 7.4: AEU and FRU Actual vs. Predicted journey times using Metric V

The estimated journey time is close to a linear relationship with actual journey time up to approximately 14 minutes for AEU and 8-10 minutes for FRU.

7.2.3 Predicted Route Differs from the Actual Route

At the time of determining an estimate journey time the route taken by the crew is unseen. With the benefit of historic route data we can calculate route estimates using the *actual* path taken by the crew as well as route estimates based on the *estimated*

route. A comparison between the two would highlight issues with the route prediction process.

Figure 7.5 is a plot of journey time prediction error using Metric V for two scenarios: a) with knowledge of the actual original taken (in green) and b) without knowledge of the route taken (in blue) i.e. predicting the route using the routing engine. When determining a route the shortest path algorithm will always find the quickest route according to the speed information provided. We can see that the difference between estimated route time and actual route time are similar, at least up to approximately 8 minutes.

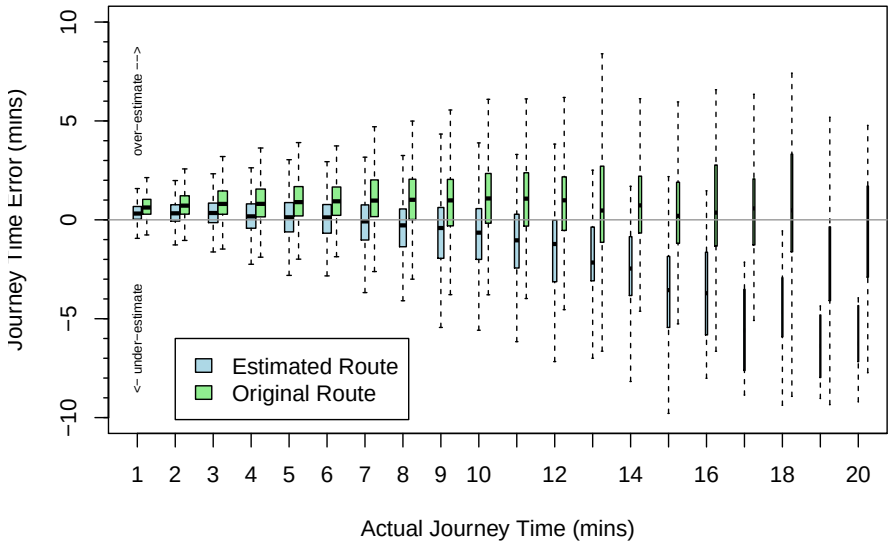


Figure 7.5: Predicted journey time error vs Actual journey times using Metric V for both actual and predicted routes

The mean estimated journey times using the actual route are accurate up to 20 minutes, with lost of precision due to the increasing variance. This leads us to conclude that the mined speeds for the road links are reliable. We conclude that that the sum of the road links used in any particular journey can indeed be used to obtain a reasonable estimate of the journey time.

7.3 Comparison of Route Taken

With reference to Figure 7.5 there are prediction benefits if one could estimate the route that a crew would take. This is especially true beyond an eight minute journey where, for an unseen route, the deviation between actual and predicted journey times becomes significant. To gain insight into how the estimated and actual routes differed we compute the average Incorrect Road Length Fraction (see Equation (5.5) on page 118) of the estimated routes, by Metric, for all journeys in our sample set. The results are shown in Figure 7.6.

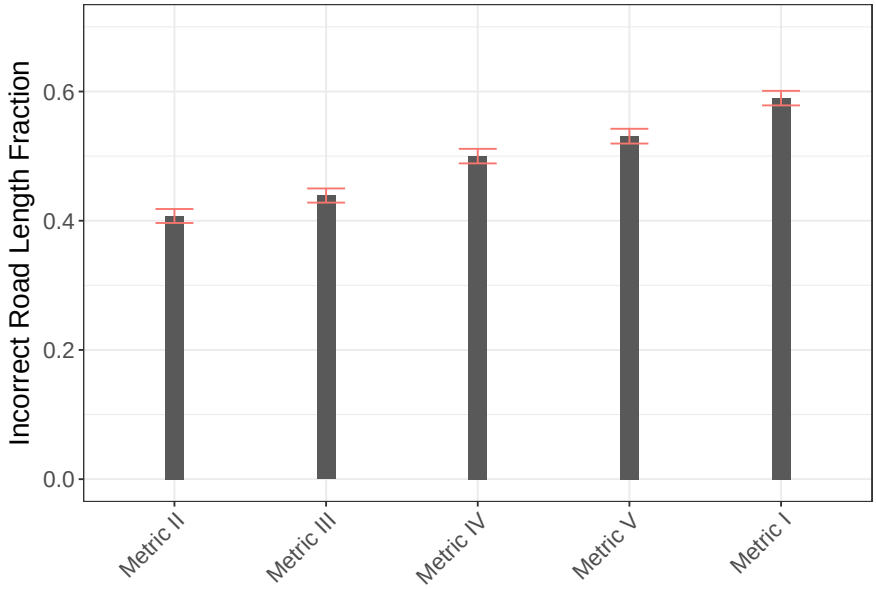


Figure 7.6: Similarity of Estimated vs. Actual route by estimator

Average IRLF for estimated routes by Metric. Confidence intervals are shown as red lines.

The lowest IRLF with a mean of 0.4153 was achieved with Metric II, producing the best overall similarity to the actual routes taken. Metrics V does not score well for route similarity, so we look more closely at the individual portions of the journey. First we divide each route into four equal portions by length. Figure 7.7 reveals that the characteristics of each portion are quite different. The figure contains four plots, once for each portion of the journeys we analysed. The individual plots show a histogram of

the number of journeys by percentage similarity, measured as the number of coincident road segments. With all four portions there are a number of journeys whose similarity ranges from 5% to 95% similarity. These journeys make up a significant portion of the total journeys. Portion Q1 and Q4, the first and last quarters of the journey, show a high peak for 100% similarity, indicating a large percentage of routes have identical estimated and actual routes. Portions Q2 and Q3 show a similar high peak for a complete match in the route and also a peak for no similarity at all. This suggests that most deviation from the route is occurring in the middle part of the journey. This pattern is repeated for all Metrics.

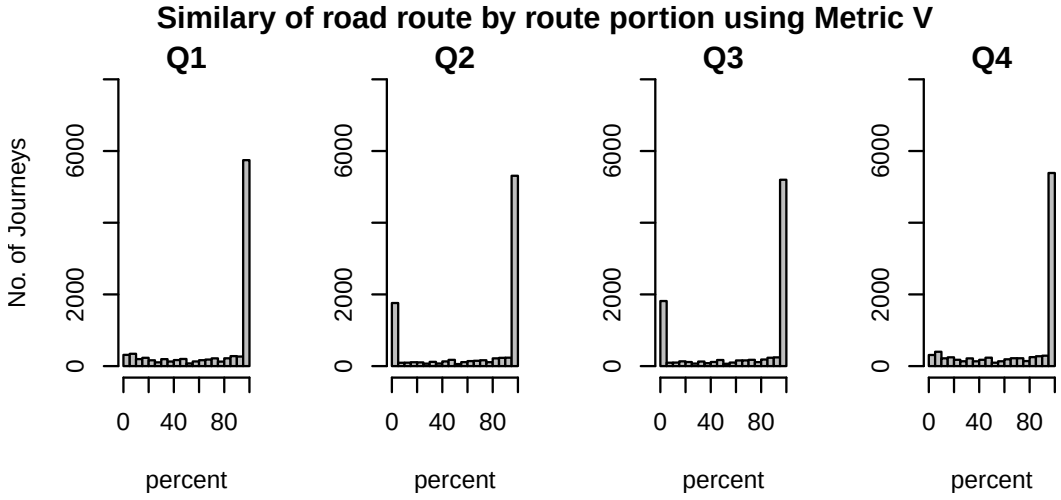


Figure 7.7: Similarity of Estimated vs. Actual route, using the Metric V by each portion of the journey.

7.4 Spatio-Temporal Effects

London is a densely populated area with a complex, roughly circular, road network. What is known from the raw GPS data is that traffic density is higher, and road journeys for comparable distance are slower in the centre of London at certain times of day. Here we explorer Metric V's accuracy in predicting journey times under changing

spatio-temporal dimensions.

7.4.1 Spatial Variation

There are no specific measures for traffic density in London as such, however, a starting point is to measure journey time estimate versus distance in kilometres from the centre of London. In this case we choose the commonly used location of Charing Cross as the centre point of London. For the distance we calculate the straight-line distance between the centre of London and the geographic mid-point between start and end locations of each journey.

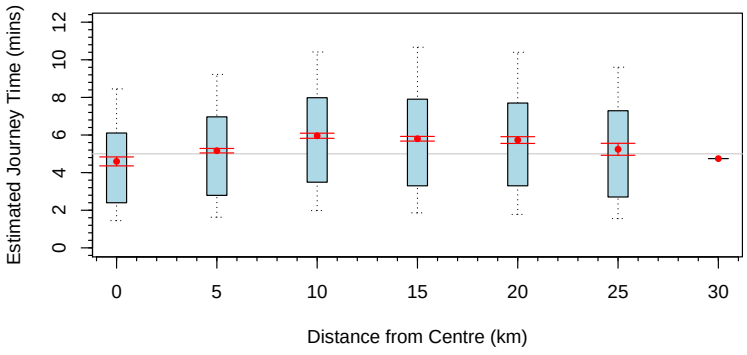


Figure 7.8: Estimated journey time by distance from the centre of London
Boxplot of average journey times, in minutes, predicted by Metric V aggregated by distance from Charing Cross. Whiskers are 10 and 90 percentiles, light blue boxes are 25 and 75 percentiles and red horizontal marks denote the 95% confidence intervals around the mean.

The histogram in Figure 7.8 shows estimated journey time against distance from Charing Cross. The histogram reveals that journey times are roughly consistent across London, at between 4-6 minutes for the average journey, with a peak at approximately 15km from the centre. These figures are, at least in part, indicative of how close a resource might be to an incident, rather than speed of the vehicle, and are likely also to be dependent on the number of resources available at any one time. The figure shows, therefore, that response times are better in the centre of London.

In Figure 7.9 we instead look at a) the journey time prediction error and b) estimated vs actual route similarity as a function of distance from the centre of London, using the

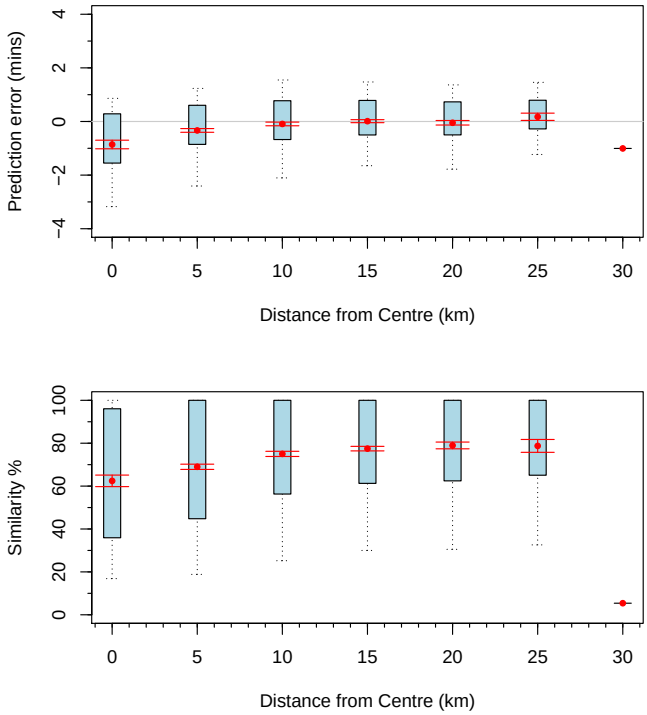


Figure 7.9: Spatial Prediction Error Variation
(a) Prediction error variation by distance from the centre of London and, (b) Route similarity by distance from the centre of London using Metric V. Whiskers are 10 and 90 percentiles, light blue boxes are 25 and 75 percentiles and red horizontal marks denote the 95% confidence intervals around the mean.

Metric V. The mean prediction error and variance is greatest in the centre of London, gradually improving with increased distance from the centre. This pattern is repeated for the route similarity, again, measured as the percentage of coincident route length of estimated and actual routes. We conclude that the greatest variance and instability of route journey time estimates and route prediction is focused in the centre of London.

7.4.2 Temporal Variation

In a similar vein to the spatial analysis, in Figure 7.10 we show prediction error throughout the day for Metric V. It appears to perform well in terms of temporal stability with a mean error (accuracy) close to zero minutes throughout the day. There is a slight

fluctuation at around 07:00 and 18:00. This could be due to rush-hour traffic. Also note that at 07:00 and 19:00 there is a crew shift change which may be a contributory factor. The precision in prediction error is also stable except where there is a slight increase at around 18:00.

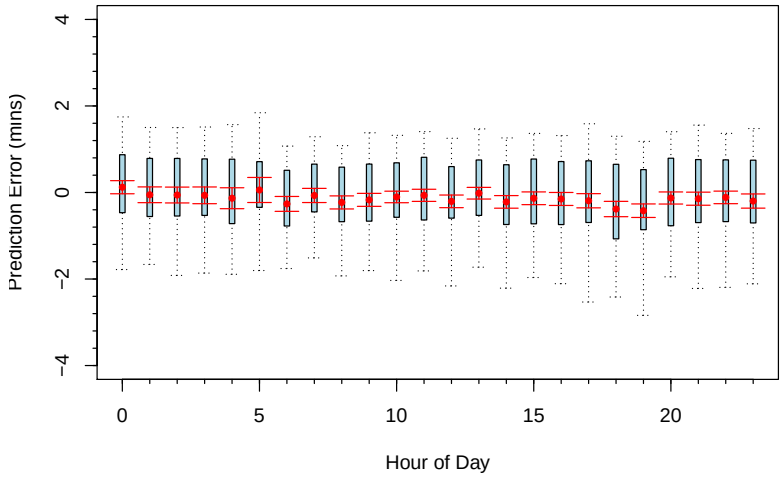


Figure 7.10: Prediction accuracy of Metric V shown by hour of day
Boxplot of prediction error variation throughout the day using Metric V. There is little variation in prediction accuracy except for some slight variation around 07:00 and 19:00. Whiskers are 10 and 90 percentiles, light blue boxes are 25 and 75 percentiles and red horizontal marks denote the 95% confidence intervals around the mean.

7.5 Correcting Estimation Error

We know from our previous analysis in Section 7.2.1 that there is a possible under-estimation in travel time when using Metric V. For a perspective on the uncorrected journey estimation we look at the journey time estimation error against our actual journey time for both vehicle types, see Figure 7.11.

This plot shows more succinctly the under-estimation as journey time increases. The rate of change of variance and bias are similar for both AEU and FRU. The descending (under-estimation) shape of the error in Figure 7.11 is close to linear for journeys up to approximately 10 minutes, thereafter the rate of error increases.

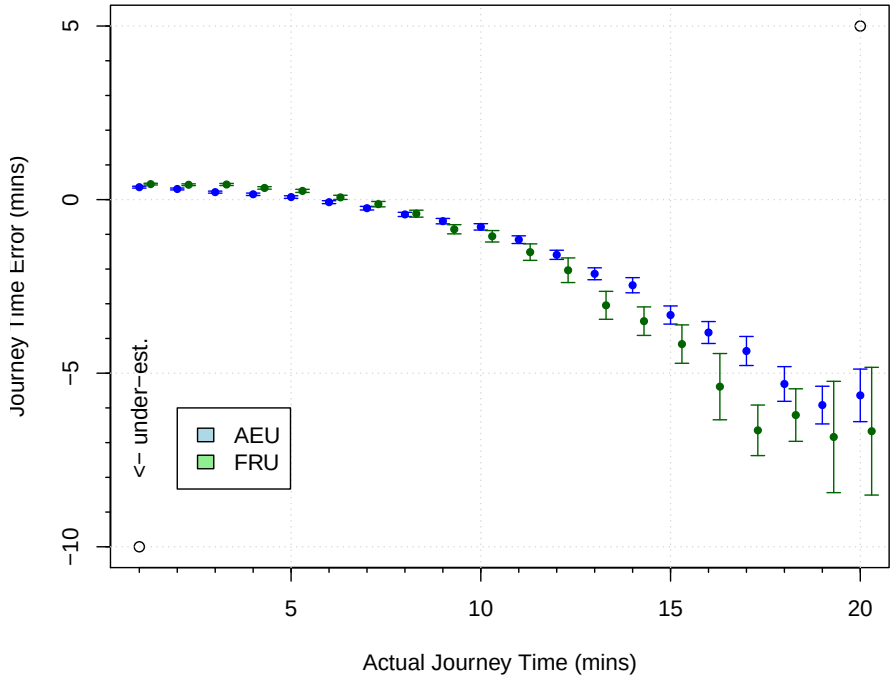


Figure 7.11: Journey time error using metric V
this plot shows the difference between estimated and actual journey times (journey time error) using the routing engine and Metric V to predict route duration for both AEU and FRU, showing 95% mean confidence intervals.

In Figure 7.12 we plot journey time error (*estimated* − *actual*) against several additional explanatory variables (full size plots are in the Appendix D from Figure 8 to 11). These plots are, from top right to bottom left:

1. Delta vs route distance. Journey time error is compared with the estimated route distance.
2. Delta vs the number of roads. The value is calculated as the number of road name changes on the journey.
3. Delta vs the total road angle change. The angle change is measured as the sum of change of direction between each road link.
4. Delta vs the number of turns. This is a count of angle changes between road links greater that 45°.

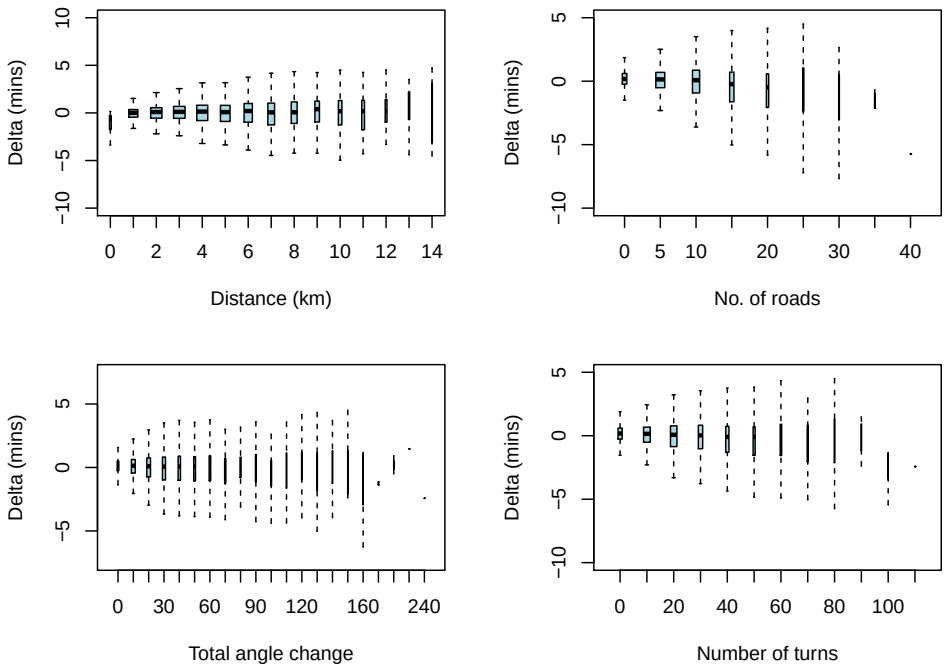


Figure 7.12: Comparison of journey time prediction error using Metric V
These charts compare the journey time prediction error in minutes ($estimate - actual$) with a) route distance, b) the number of road name changes c) total road angle change (i.e the total change in direction in degrees), and d) number of turns.

There is little or no observed correlation with the journey time prediction error to any of the metrics listed above and we conclude that no corrective coefficient can reasonably be derived. This leaves only the estimated journey time as we can see from the summary statistics in Figure 7.5 that actual journey time error is, at least in the first few minutes, proportional to the estimated journey time.

To address the problem of underestimation we adjust travel time predictions through the removal of the expected error by fitting a single order linear bias function to the curve Figure 7.11, namely:

$$t_\chi = t_\beta \times 1.166 - 46.2287, \tag{7.1}$$

where t_β is the original estimated journey time and t_χ is the corrected time in

seconds.

The first order linear model in Equation 7.1 is generated directly using linear regression to estimate actual journey durations. This approach is simplistic as the curve in Figure 7.11 does not follow a linear change (although close to linear for the first ten minutes), suggesting that alternative corrective methods could be applied, such as polynomial regression. Whilst this approach would produce a more accurate estimate of journey time it would not influence the precision of prediction. In any case, we perform this first order regression for illustration of the process in applying a bias function to reduce inaccuracies in prediction.

Figure 7.13 shows the original estimated journey time with the correction bias function applied and reports a significant improvement in accuracy with a mean error of less than one minute for journeys lasting up to 14 minutes. Note that this implies less than a minute mean error for 90% of all journeys completed by LAS in response to a Category A incident.

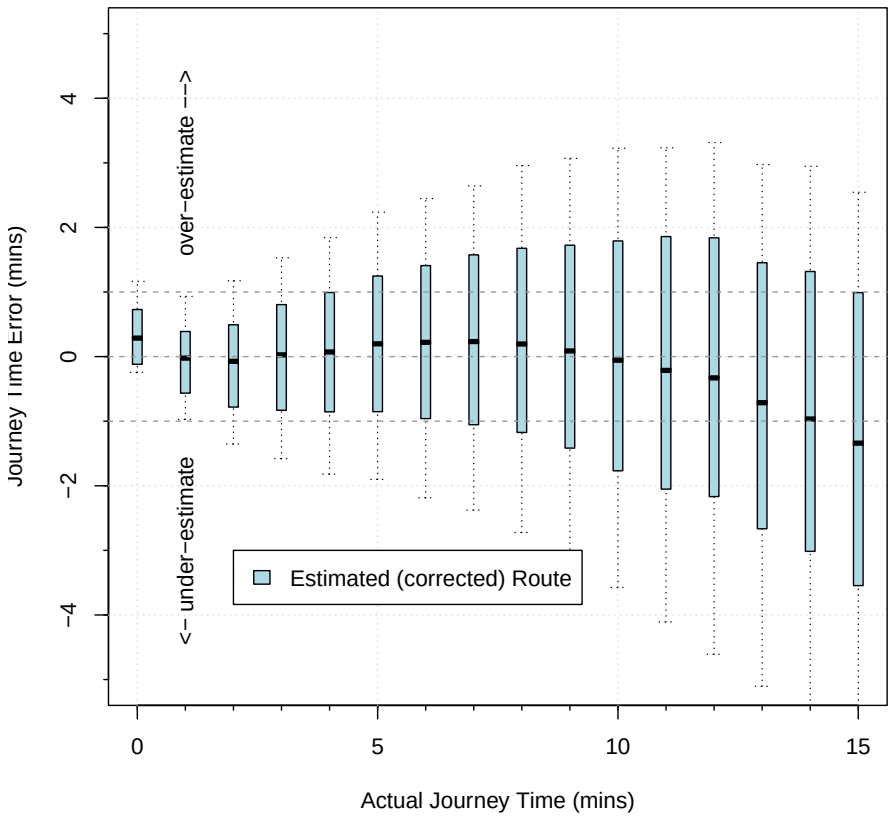


Figure 7.13: Corrected journey time prediction
Corrected Metric V journey time prediction with the application of a bias function: $t_\chi = t_\beta \times 1.166 - 46.2287$.

8

Hybrid Routing Model

Current practice in EMS is to calculate the expected arrival time of vehicles nearby to an emergency event to determine which of those vehicles to dispatch. In Chapter 7, our data suggests that estimation of arrival times at LAS could be overestimated and consequently leading to sub-optimal dispatch decisions. The ability to produce an expected route based on an emergency drivers' route choice model is therefore preferred for both accurate simulation and in live operations to predict realistic arrival times and to aid selection of the most appropriate vehicle.

8.1 Blue Light Route Choice Model

An observation of the Incorrect Route Length Fraction depicted in Figure 7.6 is that Metric II produces the best average route similarity to that taken by crew. We propose a new speed model, the BLRCM, designed to further improve route similarity.

Using Metric II road speeds as a starting point, we use the Nelder-Mead algorithm (NM) [155] to minimise the average IRLF of 1000 historic journeys. We design an objective function for use with NM that accepts a set of road speed parameters and returns the average IRLF between historic and routes generated using those parameters. The NM iteratively perturbs parameters until a convergence threshold is met.

A test set of 1000 historic journeys where chosen at random of a least 20 road links long to allow for sufficient complexity in the routes. We found that initial perturbation of 15 mph for the road speeds, and 5 seconds for the junction delay was sufficient scope for NM to fully explore the variable dimensions. Convergence was set to 0.0001 as accuracy beyond this was not required. Results of the optimisation process are listed in Table 8.1. The optimisation process improved the IRLF of the test cohort from 0.4153 to 0.3544. The junction delay is significantly reduced and all road speeds were increased except for local streets.

Table 8.1: Blue Light Route Choice Speed Model

Optimised speeds improve IRLF from 0.4153 to 0.3544.

Road type	Original Speed	Optimised Speed
Junction delay	2.5	0.45
A Road	29	31.30
Alley	3	5.42
B Road	24	26.62
Local Street	14	12.26
Minor Road	19	20.04
Motorway	35	38.87
Pedestrianised Street	2	5.03
Private Road- Publicly	5	7.84
Private Road- Restricted	5	7.65

8.2 Hybrid Routing Model

The Hybrid Routing Model is an extension of the Dijkstra routing engine used in this research, modified to include an additional post-processing step. Initially the HRM estimates a shortest-path route using the BLRN and BLRCM using speeds listed in Table 8.1. A post-processing step discards the journey time estimate produced by BLRCM for the route and recalculates the total estimated journey time by summing estimated journey time for each road link using Metric V.

In Chapter 7 we put forward a corrected Metric V model described in Section 7.5. To test the performance of HRM we calculate journey time estimates of 10,000 unseen routes using both the HRM and corrected Metric V. For each test route we calculate the delta between the predicted ETA and the actual journey time for both models and plot the results in Figure 8.1.

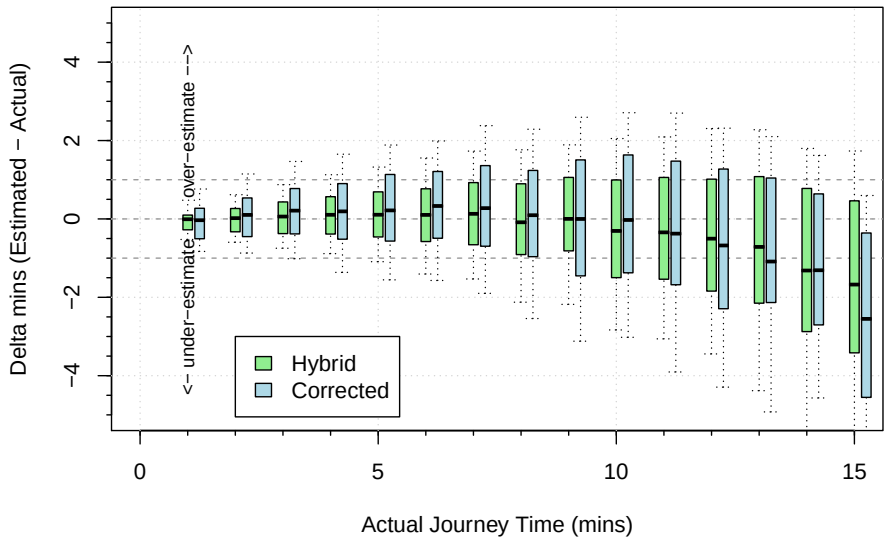


Figure 8.1: Prediction accuracy of the HRM
Box-plot of the HRM accuracy (green) and corrected Metric V (blue) for all journeys up to 15 minutes. Boxes are $25^{th}/75^{th}$ percentiles, whiskers are $10^{th}/90^{th}$ percentiles.

The Figure shows HRM improves on the precision of corrected Metric V journey time

estimation for journeys up to 15 minutes duration. We also know that there is a higher probability that the calculated route will be similar to a route actually undertaken by emergency vehicle crew due to the low IRLF. The estimations appear to be accurate to ±1 minute up to 8 minutes for 50% of the journeys. Table 8.2 lists the prediction error standard deviations as an indicator of precision, for both HRM and Corrected Metric V for different journey durations. In all cases the HRM outperforms the Corrected Metric V in precision.

Table 8.2: HRM and Corrected Metric V Delta Standard Deviations

Standard deviations, in minutes, of the prediction error for both HRM and Corrected Metric V by actual journey time.

Actual Journey Time	Hybrid Routing Model	Corrected Metric V
$\leq$ 4 mins	0.769784	0.920372
$\leq$ 8 mins	1.105841	1.178487
$\leq$ 12 mins	1.329786	1.433538
$\leq$ 16 mins	1.501034	1.584625

Finally, Figure 8.2 shows the spatial variation of the prediction error per London Borough. The error is lower in the outer boroughs, and underestimates journey duration in the centre. A potential reason for this variation is that in central highly populated areas, due to population density and more variable traffic conditions, an emergency vehicle may be more likely to be randomly obstructed resulting in a delay to the planned journey.

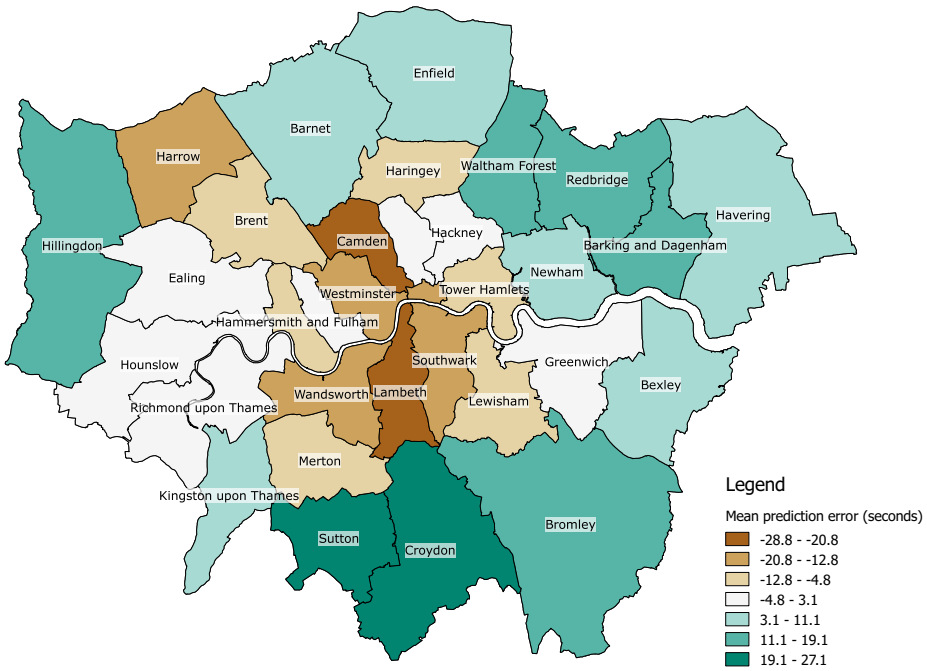


Figure 8.2: Spatial variance in the Hybrid Routing Model

Mean prediction error by London Borough, using the Hybrid Routing Model to predict emergency vehicle journeys.

9

Conclusions and Future Work

In this thesis we show how data-driven methodology can be used for the accurate prediction of journey times and routes for ambulances and fast response vehicles responding to emergency medical incidents whilst travelling on BLS.

The key phases of our approach include the exploration and characterisation of LAS ambulance data and identified key spatio-temporal features; a large-scale reconstruction of emergency vehicle journeys from coarse location tracking data using the HMM/V map-matching algorithm; extrapolation of comprehensive speed information from the estimated ground truth journeys; creation of a navigation road network graph

specifically tailored for emergency vehicles; development of several alternative edge-cost estimator metrics and the assessment of their performance characteristics; and, the use of a combination of metrics for the development of a hybrid model that achieves the highest route similarity whilst minimising the journey duration uncertainty.

Our analysis of the LAS dataset reveals that incident generation is rising steadily and occurs in clusters coincident with London's urban nuclei. Vehicle movement is significantly affected by the ebb and flow of traffic due peak times, with fast response vehicles outperforming ambulance in terms of arrival time. Analysis of routes taken by emergency vehicles shows that there are differences to those generated by standard shortest path algorithms. The hybrid model counters the route differences, offering stable performance across both spatial and temporal dimensions and can be used to accurately determine both journey times and routes for ambulances and fast response vehicles travelling on blue lights and sirens in London.

The hybrid model has considerable implications for ambulance services aiming to improve the accuracy and fidelity of their emergency response as routing technology underpins many of the strategic and tactical processes within an ambulance ecosystem.

9.1 Future Direction

9.1.1 Software Emulation

The use of sophisticated emulations are a key instrument employed for the investigation of more effective and efficient operational policy. Specifically, the ability to trace closely the true movements of emergency vehicles and to estimate their expected arrival time, through increased fidelity, enables detailed exploration of the many facets of ambulance-related operational research through the emulation of realistic scenarios.

Data-driven decision making can offer significant improvements in the planning of key service requirements such as forecasting required ambulance staffing levels and resource placement, enabling services to balance strategy and tactics through the accurate

assessment, for example, of the effects of particular dispatch tactics on meeting their strategic objectives.

9.1.2 Tactical Displays

Moreover, our methodology can be applied in real-time at emergency service operating centres for tactical purposes such as determining the ability of the service to adequately cover its area of responsibility. Additionally, accurate routing engines provide the foundation for such tactical displays and can provide answers to 'what-if' questions such as estimating the effects on system performance and quantifying the effects on its ability to maintain a high level of response resulting from specific vehicle dispatch or relocation decisions or as a result of permitting crews to go on a break. To achieve this aim, systems such as Geotracker at LAS require updating to provide greater accuracy and presentation of routing information.

9.1.3 Real-time Traffic

One feature of this work is that our methodology considers only historical data collected internally by the emergency ambulance service. In future work, we aim to explore potential improvements that can be achieved using real-time data feeds such as traffic, weather and other related context information retrieved from external systems. Although extending the methodology to cater for such data sources appears relatively straightforward, a major challenge relates to assessing the quality of such third-party data sources and validating their accuracy, especially considering the possibility of loss of life that erroneous or intentionally misleading information may have.

9.1.4 Routing and Navigation

Our analysis suggests that a bespoke in-vehicle navigation engine tailored specifically to emergency vehicles travelling on BLS, rather than one developed for general civilian traffic as it is currently the case, could lead to significant reductions in crew arrival times

at the site of an incident. Specifically, the use of Metric V introduced in this thesis has a tendency to select emergency vehicle routes that provide faster arrival times. However, we recognise that extensive work would be required to establish whether this approach would reveal concrete benefits.

9.1.5 Dispatch Algorithms

Deciding which emergency vehicle is dispatched to a critically ill patient relies heavily on routing technology to estimate predicted journey times, and is a core decision-making process for both patient welfare and general service performance. We demonstrate that accurate prediction is not a simple shortest path problem and is better served in a two-step methodology by first estimating the most likely route that crew might take, and then estimating journey time based on that route.

Research shows that the tactic of sending the nearest vehicle is not always in the best interests of the patient. Novel dispatch algorithms using data-driven techniques that fuse high fidelity vehicle coverage, incident generation and vehicle routing information could benefit EMS considerably, however, research in this area is immature.

9.1.6 Use of Data

There are many opportunities to unlock the vast potential of data and technology innovations in EMS. However, policy-makers must make this possible. The NHS, for example, tends to deal with large suppliers using strict procurement routes, and have a reduced exposure and willingness to work with SMEs or academia. That being said, innovation is vital for the long term success of large health organisations. Data techniques can be utilised not only for advanced dispatch, but also delivering tactical information to responders, such as detailed real-time patient records, analysis of hospital availability and queuing times, and projection of video and patient sensor data directly to trauma rooms. Technology in this area is by no means mature and offer many opportunities for innovation.

Appendices

Appendix A

Listing 1: Coding listing for re-sampling particles

```
protected List<MotionParticle> Resample(List<MotionParticle> particles,
    int sampleCount, MapMatcherRequest request)
{
        // take the top (80)n% best particles
        particles = particles.OrderByDescending(x => x.Weight)
        .ToList()
        .Take((int) (particles.Count*request.ResampleCutoff))
        .ToList();

        // create CDF of weights and sample from the CDF so that higher
            weighted
        // particles are selected more often. This results in possible
            duplicates
        // of higher weighted particles and possible removal of lower
            weighted particles.
        var cumulativeWeights = new double[particles.Count];

        var cumSumIdx = 0;
        double cumSum = 0;
        foreach (var p in particles)
        {
                cumSum += p.Weight;
                cumulativeWeights[cumSumIdx++] = cumSum;
        }

        var maxCumWeight = cumulativeWeights[particles.Count - 1];

        var filteredParticles = new List<MotionParticle>();

        // make sampleCount particles
```

```
        for (var i = 0; i < sampleCount; i++)
        {
                var randWeight = RandomProportional.NextDouble(1)*
                    maxCumWeight;

                // find index of particle template
                var particleIdx = 0;
                while (cumulativeWeights[particleIdx] < randWeight)
                particleIdx++;

                // add particle to the new list with a uniform weight
                var p = particles[particleIdx];

                if (filteredParticles.Contains(p))
                {
                        // if the particle list already has this particle
                            then copy it
                        // and make it slightly different
                        var clone = p.Clone();
                        p = (MotionParticle) clone;
                        Perturb(request, p);
                }

                filteredParticles.Add(p);
        }

        return filteredParticles;
}
```

Appendix B

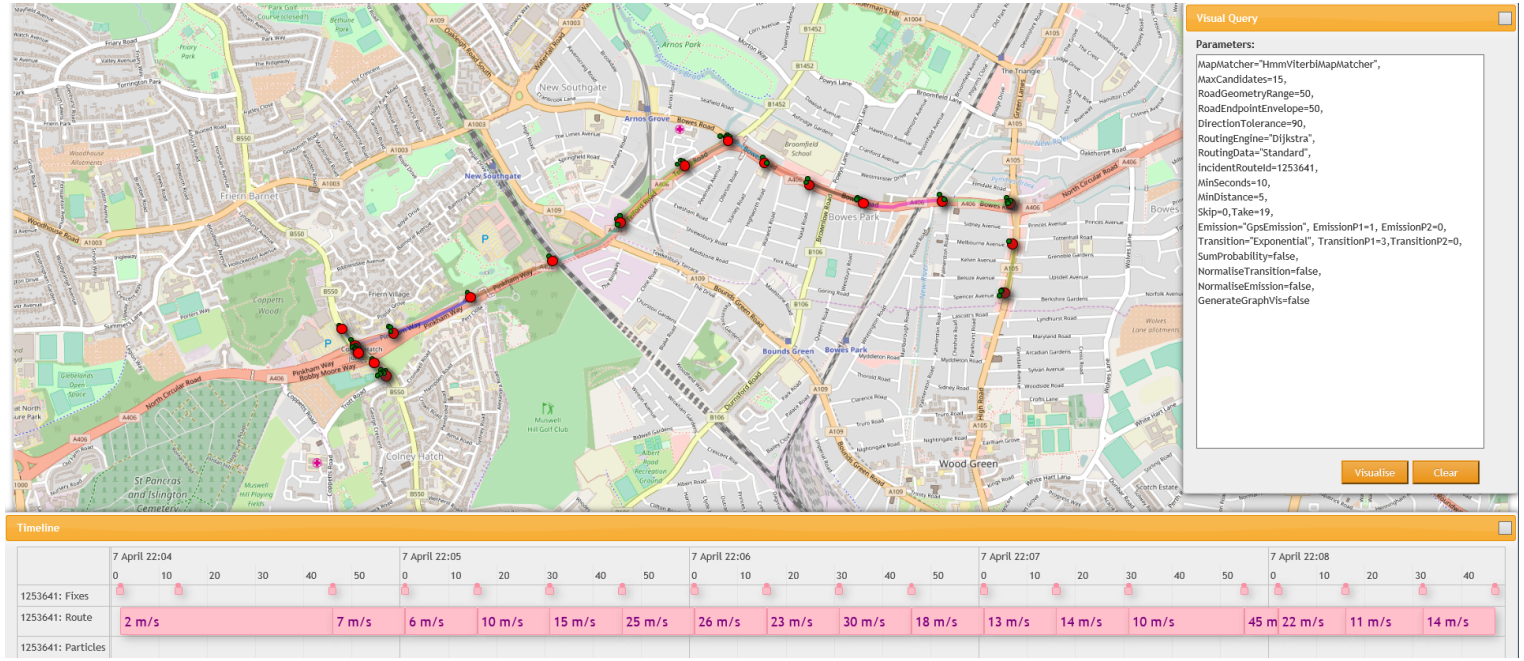


Figure 1: Map-matching example 1

This journey starts on the left hand side and finishes on the right. Large red dots indicate GPS position fixes. Nearby small green dots are candidate road positions that lie directly on the BLRN. The route has matched correctly. The clump of fixes on the left are expanded on the next figure.

Figure 2: Map-matching example 2

The crew start on the top left and need to travel south before performing a U-turn to get onto the slipway for the fly-over. The top-right red dot GPS position lies directly on the dual carriageway, but this road is inaccessible from the direction that the vehicle is coming from. The map-matcher correctly uses the ramp just north of this position fix. Correct matching and adherence to the emergency vehicle road rules is observed throughout.

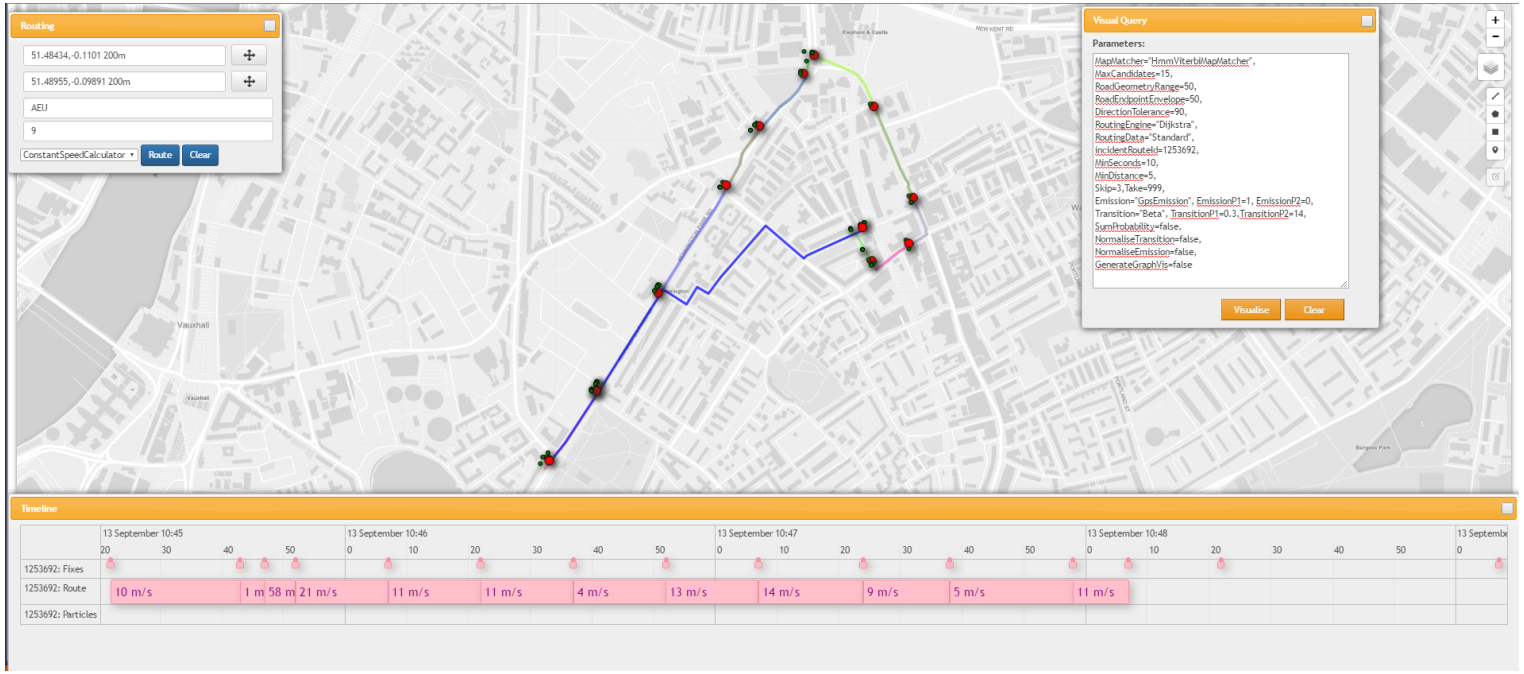


Figure 3: Map-matching example 3

An example of where the route taken by the emergency vehicle is quite different from the shortest path determined by the software. The crew have a rather long way around to get to the scene.

Appendix C

Algorithm 2 Dijkstra's shortest path algorithm

```
 1: procedure DijkstraShortestPath(start, ends, durationMax, edgeCoster)
 2:     startLocation ← GetLinkFromCoordinate(start)
 3:     thisEdgeData.RouteDuration ← 0
 4:     endLocations ← GetLinksFromCoordinates(ends)
 5:     candidates ← FibonacciHeap()              ▷ create a new fibonacci heap [156]
 6:     candidates.Add(startLocation)             ▷ add the start location to fibonacci heap
 7:     visited ← minEdge ← NULL
 8:     while true do                             ▷ keep looping until we find a result
 9:         minEdge ← candidates.GetTop()
10:         if endLocations.Contains(minEdge) then
11:             route ← CreateRoute(candidates)
12:             Return(route)
13:         end if
14:         candidates ← candidates.Remove(minEdge)
15:         minEdgeData.Processed ← true
16:         for edge ∈ minEdge.OuterEdges do
17:             visited.Add(edge)
18:             if edge.Processed ≠ true then     ▷ this is a new edge
19:                 edge.Duration ← edgeCoster(edge)
20:                 thisDuration ← minEdge + edge.Duration
21:                 if thisDuration > thisEdgeData.RouteDuration then
22:                     thisEdge.RouteDuration ← thisDuration
23:                     candidates.Add(edge)
24:                 end if
25:             end if
26:         end for
27:     end while
28: end procedure
```

Appendix D

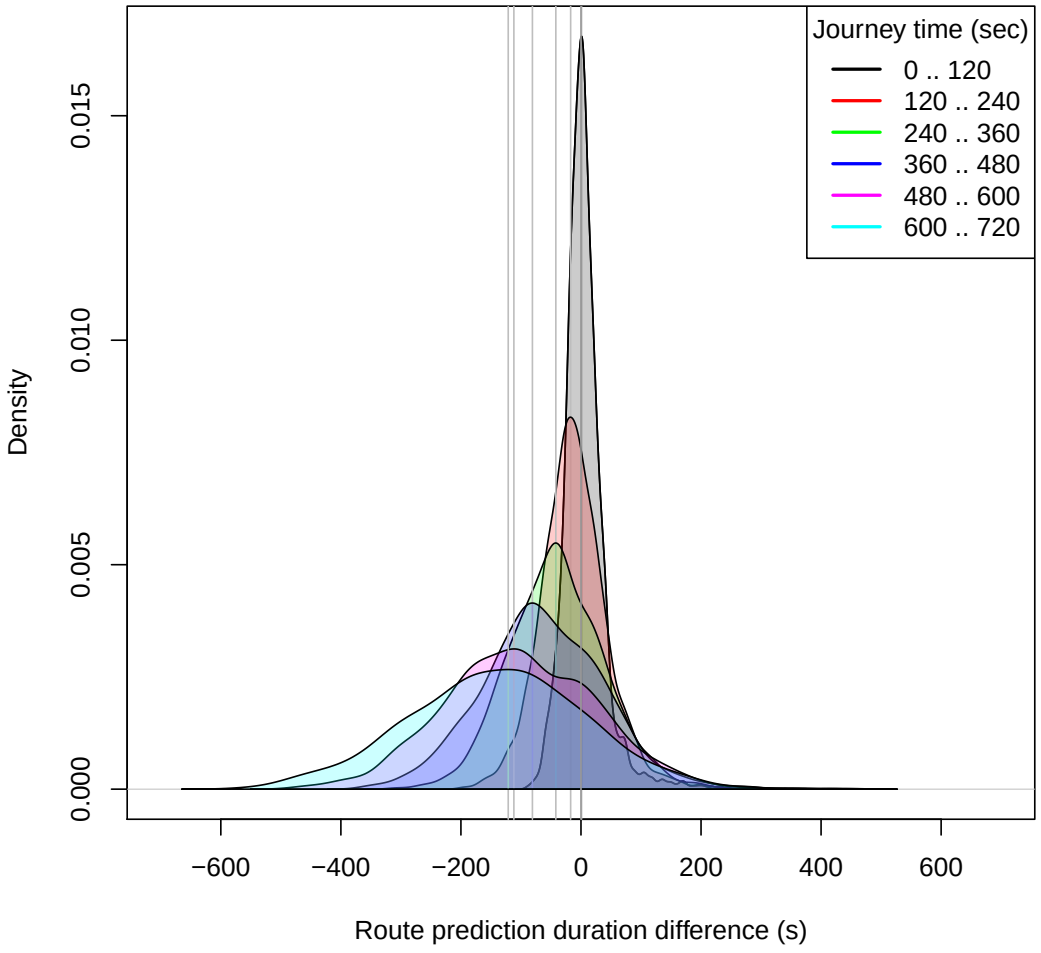


Figure 4: Prediction accuracy by journey time for Metric I

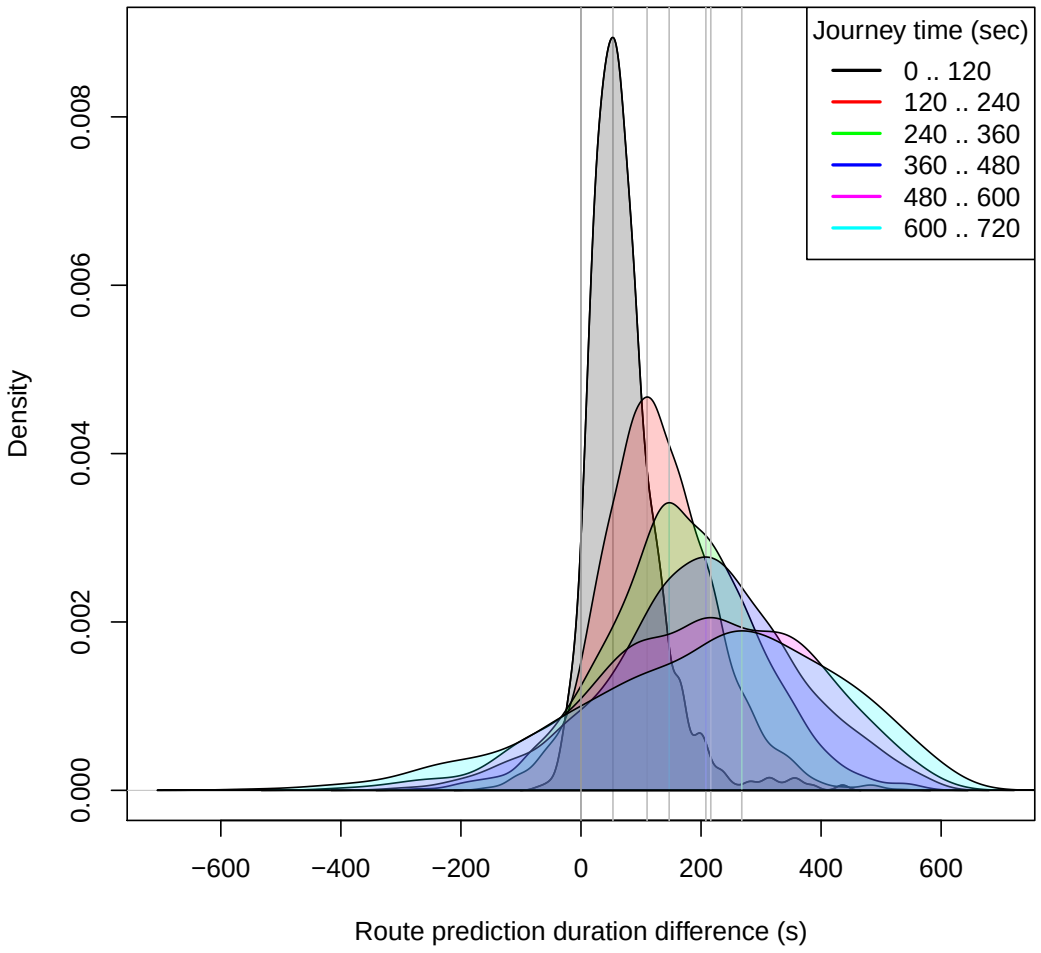


Figure 5: Prediction accuracy by journey time for Metric II

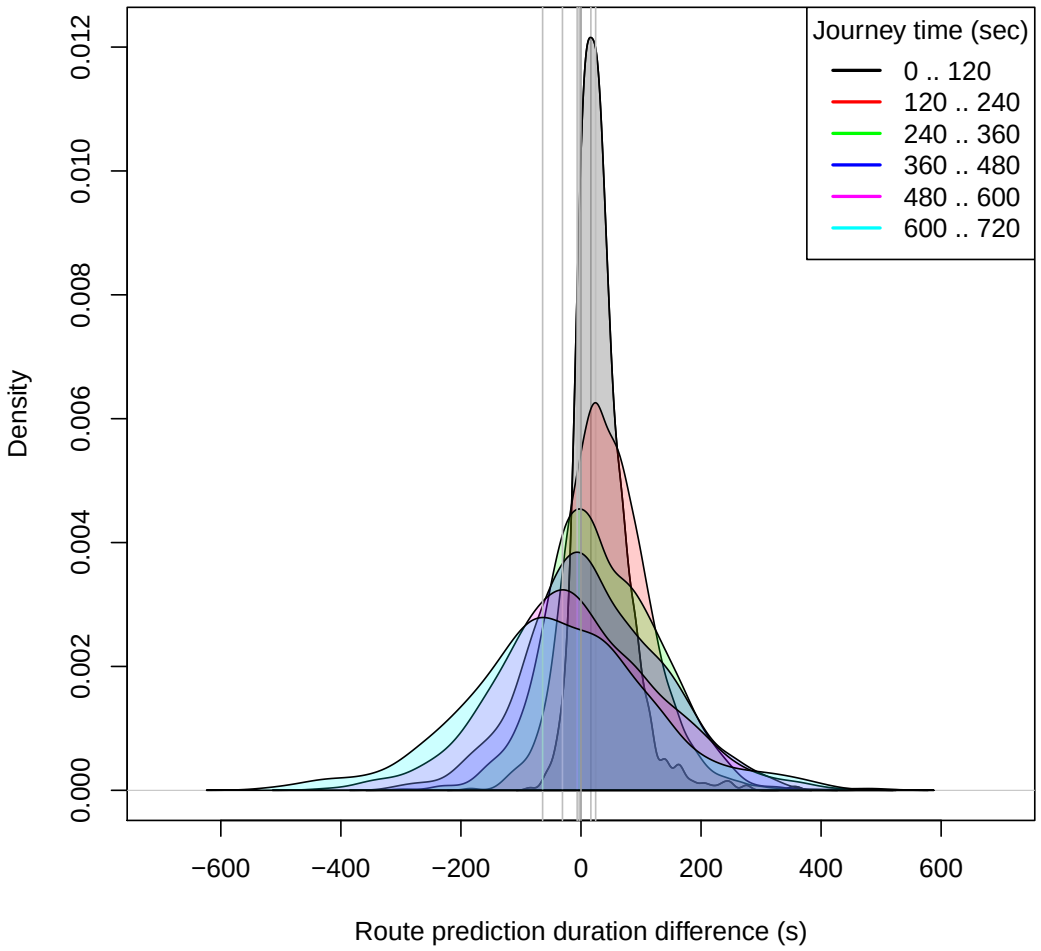


Figure 6: Prediction accuracy by journey time for Metric III

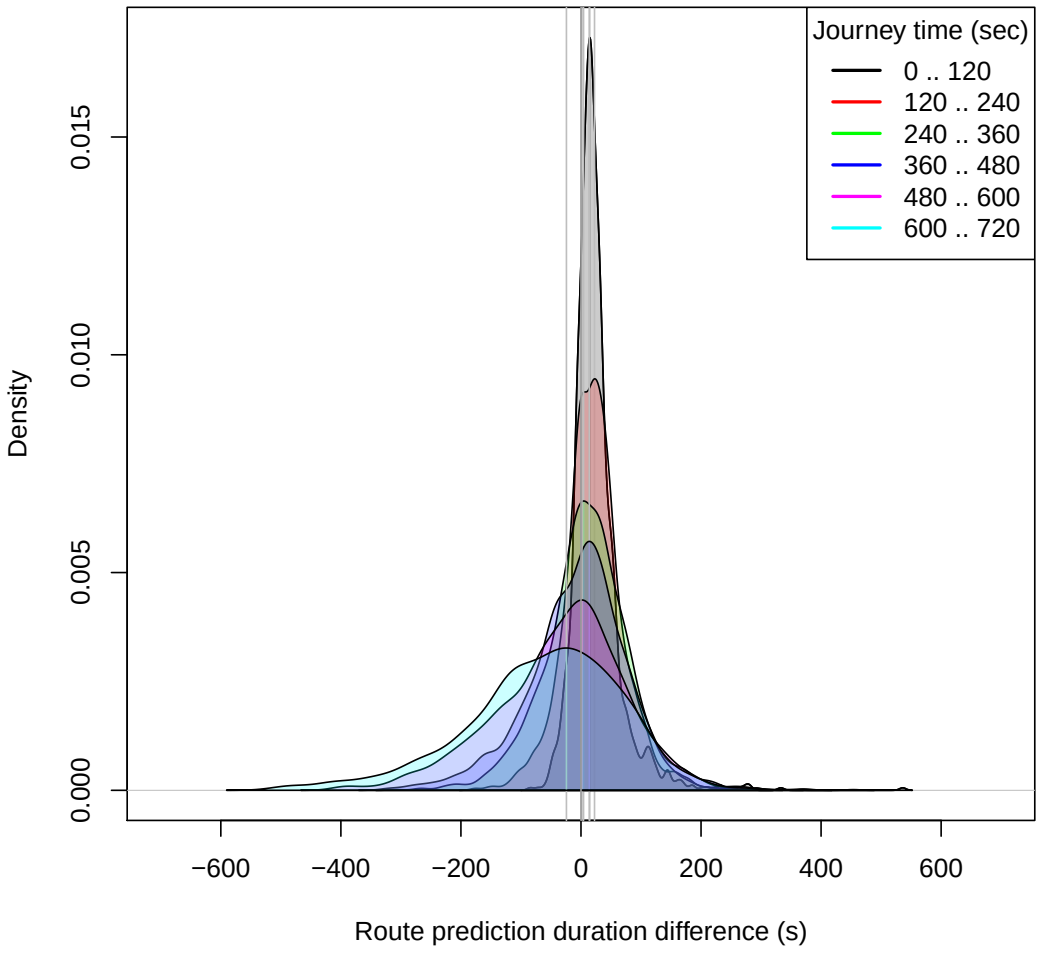


Figure 7: Prediction accuracy by journey time for Metric V

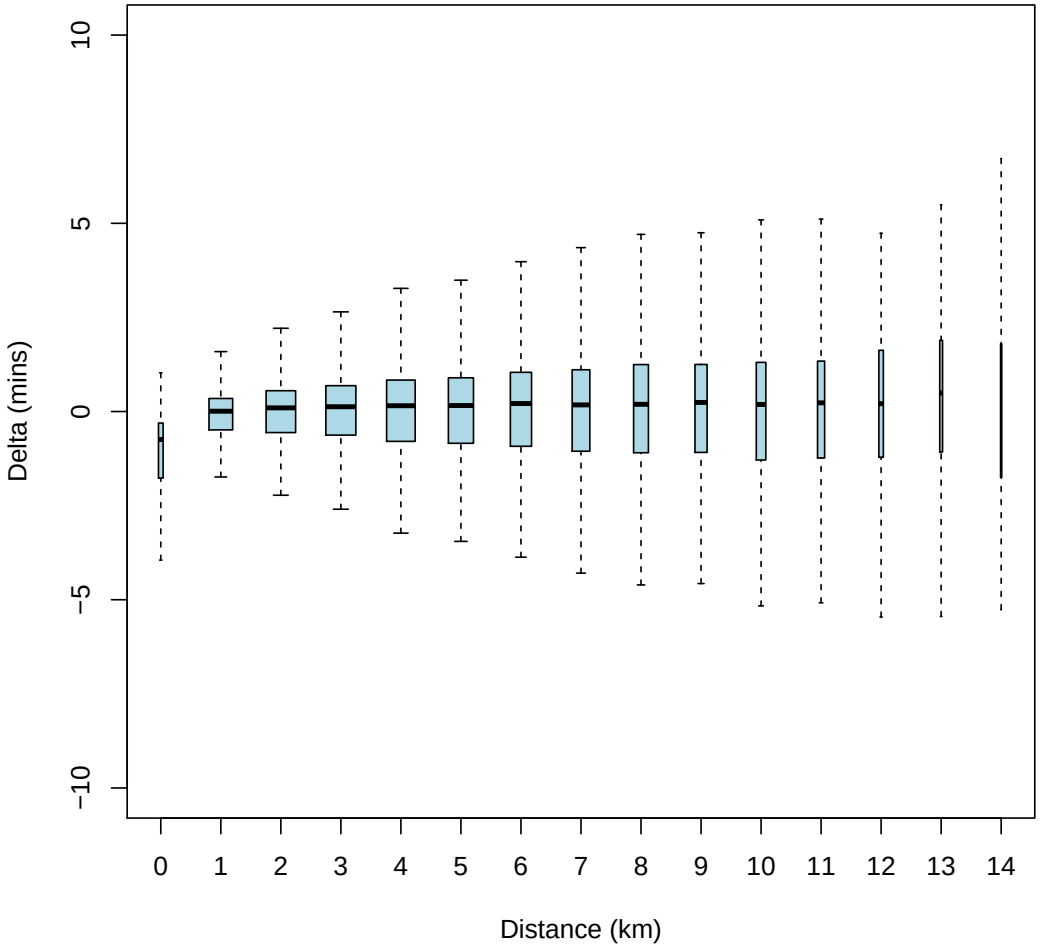


Figure 8: Journey time error by route distance using Metric V
The chart compares estimation error in minutes ($estimate - actual$) against route distance.

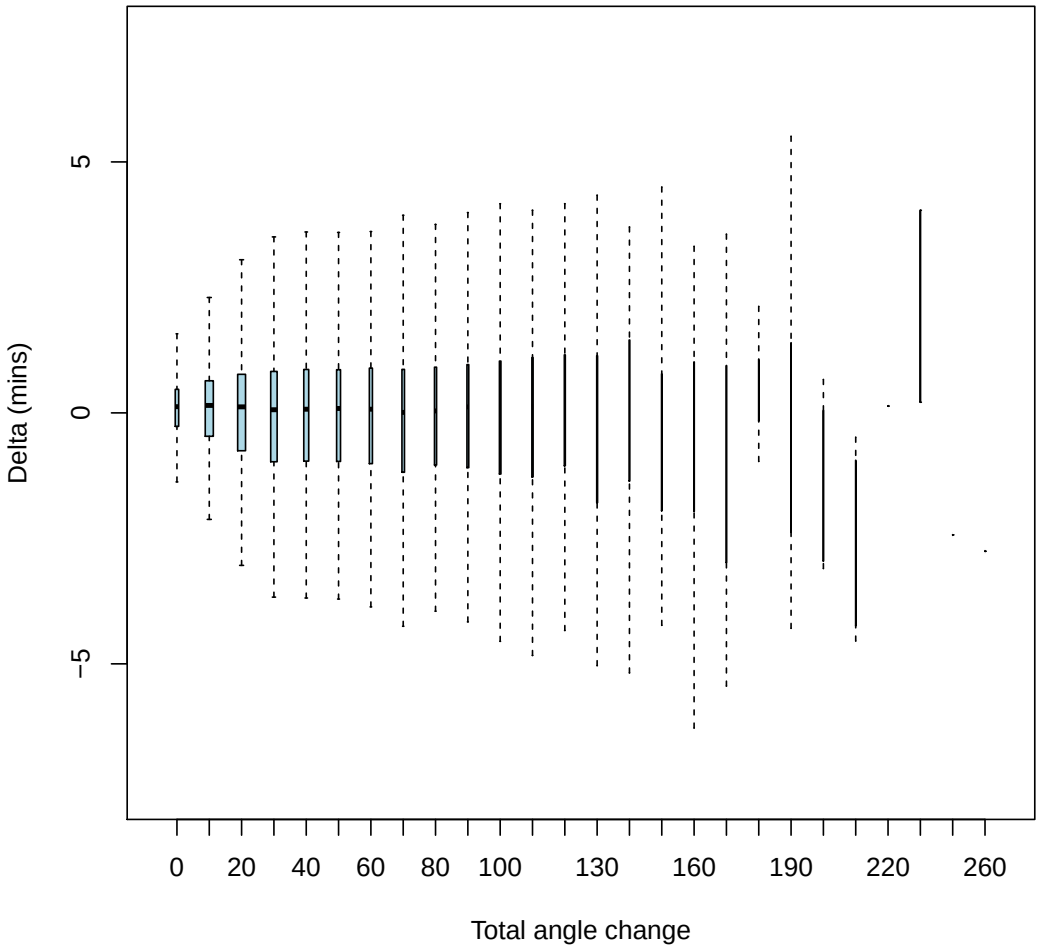


Figure 9: Journey time error by total road angle change using Metric V
The chart compares estimation error in minutes ($estimate - actual$) against total road angle change (i.e the total change in direction in degrees).

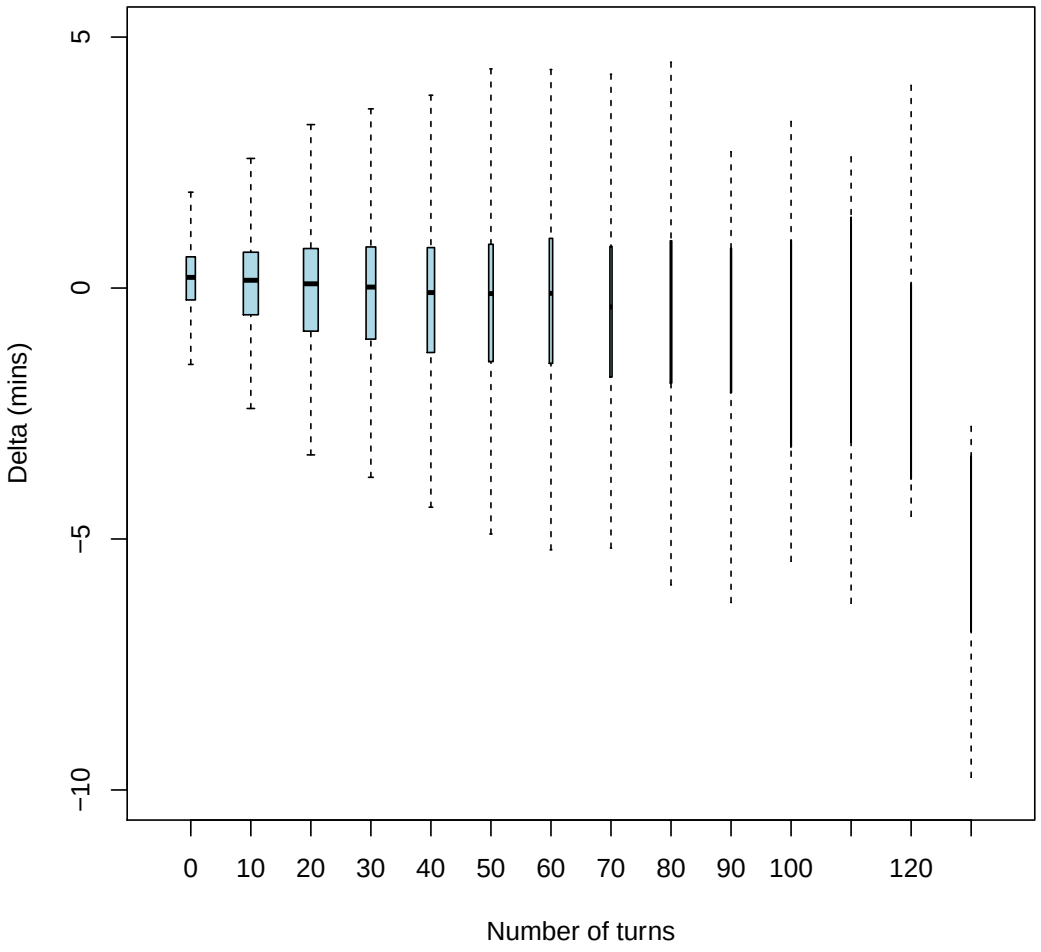


Figure 10: Journey time error by number of turns using Metric V
The chart compares estimation error in minutes ($estimate - actual$) against number of turns.

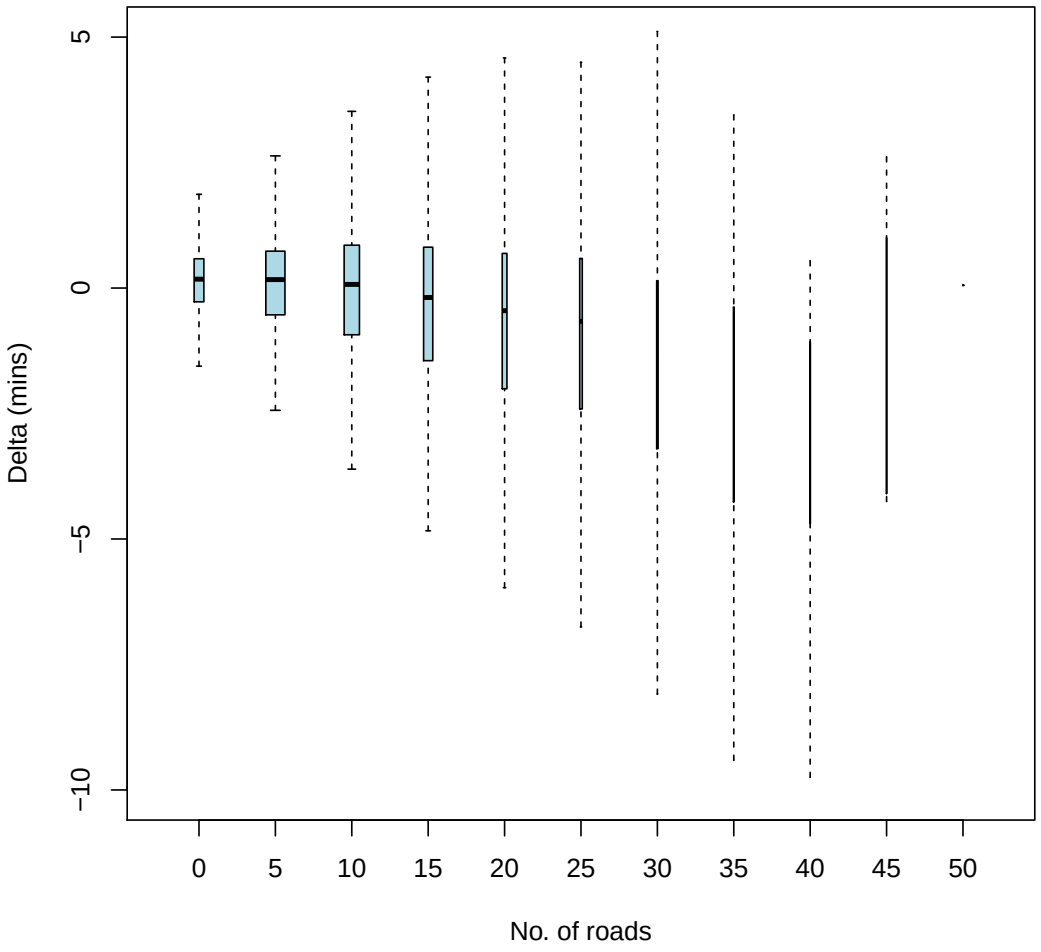


Figure 11: Journey time error by number of roads using Metric V
The chart compares estimation error in minutes ($estimate - actual$) against number of roads.

Symbols

$a(x_{i-1}, x_i)$	Transition function returns the probability of transitioning from hidden state x_{i-1} to x_i
β_a	Transition function inverse exponential β parameter
C_0	Initial set of candidate road links
b^f_{max}	Maximum relative angle between motion vector and road geometry
d_f	Final distance for a particle to move
d_{geom}	Road Geometry Range
d_m	Maximum distance perturbation to apply to a particle in metres
d^f_{min}	Track cleansing; minimum distance between fixes in metres
d_r	Shortest path distance in metres between subsequent candidates $\|\|x_{i-1}, x_i\|\|_{route}$
d_{route}	Length of original route
d_{err}	Length of map-matched route not matched to original route
d_{miss}	Length of a original route not matched to map-matched route
d^p_t	Distance a particle travels between fixes using existing motion vector and time gap
d_x	Distance in metres between candidate road positions $\|\|x_{i-1} - x_i\|\|$
d_y	Distance in metres between subsequent GPS fixes, $\|\|y_{i-1} - y_i\|\|$
δt_i	Temporal error between actual and estimated traversal between GPS fixes

δt_{α}^{m}	Prediction error in seconds between actual and estimated journey time for an unseen route
δt_{β}^{m}	Prediction error in seconds between actual and estimated journey time for a given route
n_{skip}^{f}	Track cleansing; number of GPS fixes to skip before map-matching
n_{take}^{f}	Track cleansing; maximum number of GPS fixes to process during map-matching
n_c	The number of suitable candidates found near an observation
n_{cmax}	Maximum number of candidates to consider for each observation
n_{cutoff}	Re-sample cutoff used to detect low numbers of effective particles
n_{eff}	Number of effective particles
n_p	Number of particles to use in a particle filter
n_{smpl}	Re-sample percentage when discaring low-probability particles
n_{rmax}	Maximum number of transition routes to consider between candidates
$p(y\|x)$	Emission function
p_t^i	Particle i at time t
t_{α}^{m}	Estimated journey time using edge cost estimator m
t_{min}^{f}	Track cleansing; minimum time between fixes
v_{max}^{f}	Track cleansing; maximum speed between fixes
v_t^i	Motion vector of particle i at time t
w_t^i	Weight of particle i at time t
y_i	Observation at step i
x_i	Hidden state at step i

Abbreviations

ABS Automatic Braking System. 37

AEU Ambulance Emergency Unit. 6, 11–13, 43, 44, 52, 54, 58, 74, 76–80, 83, 84, 95, 96, 127, 136, 141, 148

AMPDS Advanced Medical Priority Dispatch System. 52, 53, 91, 93

AVLS Automatic Vehicle Location System. 15, 71, 72, 76, 78, 79, 81, 85, 88

BLRCM Blue Light Route Choice Model. 19, 22, 153, 154

BLRN Blue Light Road Network. 12, 19, 21, 23, 63, 64, 69–71, 85–88, 97, 100, 101, 103, 105, 110–112, 115, 116, 122, 124, 125, 128, 131, 154, 165

BLS Blue Lights and Sirens. 17, 19, 21, 22, 26, 56, 63, 69, 97, 136, 157, 159

CAD Computer-Aided Dispatch. 52, 59–61, 64

CDF Cumulative Distribution Function. 114

CFR Community First Responder. 6, 47

CRU Cycle Response Unit. 6, 11, 46, 54

CSV Comma-Separated Values. 71, 72, 74, 76

DES Discrete Event Simulator. 62

DoH Department of Health. 50, 51

EAC Emergency Ambulance Crew. 48

ECG Electrocardiogram. 44, 46

EMR An Emergency Medical Responder is a person specially trained to provide out-of-hospital care in medical emergencies, such as a paramedics and emergency medical technicians.. 17

EMS Emergency Medical Services. 17, 24, 152

EMT Emergency Medical Technician. 49, 52

EPSG European Petroleum Survey Group. 65, 81, 107, 111

ERU A vehicle that responds to medical emergencies and can include modes such as bicycles, motorbikes, cars, vans, trucks and helicopters.. 182

FRU Fast Response Unit. 6, 11–13, 43–45, 52, 54, 55, 58, 74, 76–80, 83, 84, 95, 96, 127, 136, 141, 148

GML Geography Mark-up Language. 65, 67, 68

GNSS Global Navigation Satellite System. 99

GPS Global Positioning System. 4, 12, 13, 15, 21–23, 26, 28–32, 37–39, 45, 46, 58, 59, 64, 97–110, 113, 115–122, 128, 129, 131, 145, 177, 182

HDoP Horizontal Dilution of Precision. 105

HMM/V Hidden Markov Model with Viterbi. 15, 98, 105–108, 110, 115, 118–122, 128, 129, 131

HRM Hybrid Routing Model. 19, 154

IRLF Incorrect Road Length Fraction. 119, 143, 153

isochrone A line drawn on a map connecting points at which something occurs or arrives at the same time. An example would be a boundary drawn around an ambulance describing how far it could travel in 8 minutes.. 26

ITN Integrated Transport Network. 11, 64–66, 68–70

ITS Intelligent Transportation System. 27

LAA London Air Ambulance. 26, 47

LAS London Ambulance Service. 7, 11, 12, 15, 17, 19, 21–23, 26, 31, 42–50, 53, 55, 57, 59–64, 68, 71–73, 75, 77–79, 83, 85, 105, 118, 127–129, 150, 152

LOESS Loess regression is a nonparametric technique that uses local weighted regression to fit a smooth curve through points in a plot.. 75

MDT Mobile Data Terminal. 57, 59, 80

MEMS Microelectromechanical System. 38

MLE Maximum Likelihood Estimation. 92

MRU Motorcycle Response Unit. 6, 11, 44, 45, 54

NHS National Health Service. 16, 72

OHCA Out-of-Hospital Cardiac Arrest. 16, 17

OPALS Ontario Pre-hospital Advanced Life Support. 17

PF Particle Filter. 34, 37, 38, 98, 105, 110, 112–115

PTU Patient Transfer Unit. 6, 11, 48

SME Small Medium Enterprise. 160

TETRA Terrestrial Trunked Radio. 59

UTC Coordinated Universal Time. 59, 72, 76

VDoP Vertical Dilution of Precision. 105

WKT Well-Known Text. 65

Glossary

Ambulance A type of Emergency Response Unit (ERU), usually a truck, that conveys patients.. 16

Central Limit Theorem When independant random variables are added, their normalise sum tends toward a normal distribution. This hold true even if the random variables are not normally distributed.. 99

Great Circle A great circle is the largest circle that can be drawn on any given sphere. The length of the minor arc of a great circle between two points on a sphere is the shortest path between them.. 99

Heading Most GPS units report the track angle in degrees, which is the direction of travel relative to true north.. 29, 38, 59, 78, 82, 105, 111, 112

NM The Nelder-Mead algorithm is an n-dimensional gradient descent optimiser that attempts to minimise the result of a given objective function by iteratively varying its input parameters. The function typically accepts multiple parameters, one for each dimension.. 153

Position Fix A position estimate. GPS devices report position estimates as latitude and longitude coordinates based on the World Geodetic System (WGS84) reference system. Additional information is often associated with the position fix

including heading, speed, accuracy etc.. 13, 29, 38, 82, 85, 86, 88, 98, 99, 103, 106–112, 115–117, 119, 121, 166, 182

Road Link the Integrated Transport Network (ITN) digital road network provided by Ordnance Survey, used in this thesis, digitises segments of roads as road links. Road links contain attributes that describe notable features, for example whether the section of road is a single carriageway or whether it is bi-directional. The road link also defines the physical geometry of the road. Road links and road segments are conceptually the same.. 21, 23, 66, 97, 110–113, 116, 117, 120, 122, 125, 126, 133

Road Segment it is often useful to break up a road geometry into a sequence of one or more smaller road segments. Road segments typically start and end at road junctions or other notable features of convenience to the road network digitising agency.. 21, 28, 29, 32, 40

Standby Point A location at which an Emergency Response Unit waits before being dispatched to an emergency event.. 48

Track A track is a sequence of position fixes.. 30, 81, 82, 120

Urban Canyons Tall buildings that flank either side of a street, often in dense urban environments, cause a 'canyon effect' that blocks the line-of-sight signals from GPS satellites resulting in poor positioning accuracy.. 86, 99

Bibliography

[1] T. D. Valenzuela, D. J. Roe, S. Cretin, D. W. Spaite, and M. P. Larsen, "Estimating Effectiveness of Cardiac Arrest Interventions: A Logistic Regression Survival Model," *Circulation*, vol. 96, pp. 3308–13, Nov. 1997.

[2] G. D. Perkins, S. J. Brace-McDonnell, O. b. o. t. O. P. Group, A. Whittington, C. Kaye, C. Li, S. wan Hee, C. Hawkes, S. Keohane, J. Gray, N. Siriwardena, R. Andrews, R. Fothergill, K. Han, K. Mackway-Jones, C. Deakin, J. Pateman, A. Smith, R. Whitfield, J. Lumley-Holmes, J. Mark, M. Cooke, G. Perkins, J. Soar, J. O'Sullivan, J. Nolan, F. Moore, D. Whiting, T. Coats, C. Gale, J. Benger, G. Clegg, J. Ward, J. Long, I. Brodie, and D. Chamberlain, "The UK Out of Hospital Cardiac Arrest Outcome (OHCAO) Project," *BMJ Open*, vol. 5, p. e008736, Jan. 2015.

[3] K. Kajino, T. Kitamura, K. Kiyohara, T. Iwami, M. Daya, M. E. H. Ong, T. Shimazu, and D. Sadamitsu, "Temporal Trends in Outcomes after Out-of-Hospital Cardiac Arrests Witnessed by Emergency Medical Services in Japan: A Population-Based Study," *Prehospital Emergency Care*, vol. 0, pp. 1–8, Feb. 2016.

[4] NHS England, "Statistics : Ambulance Quality Indicators." https://www.england.nhs.uk/statistics/statistical-work-areas/ambulance-quality-indicators/, Oct. 2018.

[5] A. P. HSR department at ScHARR University of Sheffield, "A Comparative Review of International Ambulance Service Best Practice," tech. rep., Office of Strategic Health Authorities, 2009.

[6] I. G. Stiell, G. A. Wells, B. Field, D. W. Spaite, L. P. Nesbitt, V. J. De Maio, G. Nichol, D. Cousineau, J. Blackburn, D. Munkley, L. Luinstra-Toohey, T. Campeau, E. Dagnone, and M. Lyver, "Advanced Cardiac Life Support in Out-of-Hospital Cardiac Arrest," *New England Journal of Medicine*, vol. 351, pp. 647–656, 2004.

[7] A. Drake, E. Sklar, L. Smith, S. Parsons, and E. Schneider, "Data for Ambulance Dispatch," tech. rep., Policy Institute at King's College London, May 2018.

[8] U. L. Commission, "Road Traffic Regulation Act 1984," 1984.

[9] London Ambulance Service, "Our five-year strategy." https://www.londonambulance.nhs.uk/about-us/our-plans-for-the-future/, Oct. 2014.

[10] QualityWatch, "Ambulance Response Times - Care Quality Indicators." http://www.qualitywatch.org.uk/indicator/ambulance-response-times, 2016.

[11] L. Aboueljinane, E. Sahin, and Z. Jemai, "A Review on Simulation Models Applied to Emergency Medical Service Operations," *Computers & Industrial Engineering*, vol. 66, pp. 734–750, 2013.

[12] Y. Yue, L. Marla, and R. Krishnan, "An Efficient Simulation-Based Approach to Ambulance Fleet Allocation and Dynamic Redeployment," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, July 2012.

[13] M. Zarkeshzadeh, H. Zare, Z. Heshmati, and M. Teimouri, "A Novel Hybrid Method for Improving Ambulance Dispatching Response Time through a Simu-

lation Study," *Simulation Modelling Practice and Theory*, vol. 60, pp. 170–184, Jan. 2016.

[14] M. Poulton and G. Roussos, "Towards Smarter Metropolitan Emergency Response," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 2576–2580, Sept. 2013.

[15] M. Poulton, A. Noulas, D. Weston, and G. Roussos, "Modeling Metropolitan-Area Ambulance Mobility Under Blue Light Conditions," *IEEE Access*, vol. 7, pp. 1390–1403, 2019.

[16] J. M. Chaiken and R. C. Larson, "Methods for Allocating Urban Emergency Units: A Survey," *Management Science*, vol. 19, no. 4, pp. P110–P130, 1972.

[17] L. Brotcorne, G. Laporte, and F. Semet, "Ambulance Location and Relocation Models," *European Journal of Operational Research*, vol. 147, pp. 451–463, 2003.

[18] M. S. Daskin, "What You Should Know about Location Modeling," *Naval Research Logistics*, vol. 55, pp. 283–294, June 2008.

[19] H. K. Smith, G. Laporte, and P. R. Harper, "Locational Analysis: Highlights of Growth to Maturity," *J Oper Res Soc*, vol. 60, pp. S140–S148, May 2009.

[20] V. Bélanger, A. Ruiz, and P. Soriano, "Recent optimization models and trends in location, relocation, and dispatching of emergency medical vehicles," *European Journal of Operational Research*, Mar. 2018.

[21] R. Aringhieri, M. E. Bruni, S. Khodaparasti, and J. T. van Essen, "Emergency medical services and beyond: Addressing new challenges through a wide literature review," *Computers & Operations Research*, vol. 78, pp. 349–368, Feb. 2017.

[22] Z. Zhou and D. S. Matteson, "Predicting Ambulance Demand: A Spatio-Temporal Kernel Approach," *arXiv:1507.00364 [stat]*, July 2015.

[23] M. E. H. Ong, F. S. P. Ng, J. Overton, S. Yap, D. Andresen, D. K. L. Yong, S. H. Lim, and V. Anantharaman, "Geographic-time distribution of ambulance calls in Singapore: Utility of geographic information system in ambulance deployment (CARE 3)," *Annals of the Academy of Medicine, Singapore*, vol. 38, pp. 184–191, Mar. 2009.

[24] S. S. W. Lam, F. N. H. L. Nguyen, Y. Y. Ng, V. P.-X. Lee, T. H. Wong, S. M. C. Fook-Chong, and M. E. H. Ong, "Factors affecting the ambulance response times of trauma incidents in Singapore," *Accident Analysis & Prevention*, vol. 82, pp. 27–35, Sept. 2015.

[25] D. Degel, L. Wiesche, S. Rachuba, and B. Werners, "Time-dependent ambulance allocation considering data-driven empirically required coverage," *Health Care Management Science*, vol. 18, pp. 444–458, Dec. 2015.

[26] F. Regis-Hernández, E. Lanzarone, V. Bélanger, and A. Ruiz, "A Data-Driven Districting to Improve Emergency Medical Service Systems," *IFAC-PapersOnLine*, vol. 51, pp. 998–1003, Jan. 2018.

[27] R. McCormack and G. Coates, "A simulation model to enable the optimization of ambulance fleet allocation and base station location for increased patient survival," *European Journal of Operational Research*, vol. 247, pp. 294–309, Nov. 2015.

[28] M. Amorim, S. Ferreira, and A. Couto, "Emergency Medical Service Response: Analyzing Vehicle Dispatching Rules," *Transportation Research Record*, p. 0361198118781645, June 2018.

[29] D. Bandara, M. E. Mayorga, and L. A. McLay, "Priority dispatching strategies for EMS systems," *Journal of the Operational Research Society*, vol. 65, pp. 572–587, Apr. 2014.

[30] A. Anagnostou and S. J. E. Taylor, "A Distributed Simulation Methodological Framework for OR/MS Applications," *Simulation Modelling Practice and Theory*, vol. 70, pp. 101–119, Jan. 2017.

[31] H. K. Wu C, "Using a Discrete-Event Simulation to Balance Ambulance Availability and Demand in Static Deployment Systems," *ACADEMIC EMERGENCY MEDICINE*, vol. 16, 2009.

[32] M. van Buuren, G. J. Kommer, R. van der Mei, and S. Bhulai, "A Simulation Model for Emergency Medical Services Call Centers," in *2015 Winter Simulation Conference (WSC)*, pp. 844–855, Dec. 2015.

[33] M. Fakhimi and J. Probert, "Operations Research within UK Healthcare: A Review," *Journal of Enterprise Information Management*, vol. 26, pp. 21–49, Feb. 2013.

[34] S. Parragh, *Ambulance Routing Problems with Rich Constraints and Multiple Objectives*. PhD thesis, uniwien, 2009.

[35] N. A. M. Nordin, Z. A. Zaharudin, M. A. Maasar, and N. A. Nordin, "Finding shortest path of the ambulance routing: Interface of A* algorithm using C Sharp programming," in *Humanities, Science and Engineering Research (SHUSER), 2012 IEEE Symposium On*, pp. 1569–1573, IEEE, 2012.

[36] L. Talarico, F. Meisel, and K. Sörensen, "Ambulance routing for disaster response with patient groups," *Computers & Operations Research*, vol. 56, pp. 120–133, 2015.

[37] S. Panahi and M. Delavar, "Dynamic Shortest Path in Ambulance Routing Based on GIS.," *International Journal of Geoinformatics*, vol. 5, no. 1, 2009.

[38] N. A. M. Nordin, N. Kadir, Z. A. Zaharudin, and N. A. Nordin, "An application of the A* algorithm on the ambulance routing," in *Humanities, Science and Engineering (CHUSER), 2011 IEEE Colloquium On*, pp. 855–859, IEEE, 2011.

[39] R. C. Hunt, L. H. Brown, E. S. Cabinum, T. W. Whitley, N. H. Prasad, C. F. Owens, and C. E. Mayo, "Is Ambulance Transport Time With Lights and Siren Faster Than That Without?," *Annals of Emergency Medicine*, vol. 25, pp. 507–511, Apr. 1995.

[40] M. Rehn, G. Davies, P. Smith, and D. Lockey, "Emergency versus standard response: Time efficacy of London's Air Ambulance rapid response vehicle," *Emergency medicine journal: EMJ*, vol. 34, pp. 806–809, Dec. 2017.

[41] K. Petzäll, J. Petzäll, J. Jansson, and G. Nordström, "Time saved with high speed driving of ambulances," *Accident Analysis & Prevention*, vol. 43, pp. 818–822, May 2011.

[42] H. Højgaard and S. Mikkelsen, "Emergency lights and sirens secures fast arrival with predictable gain in transportation time," *Resuscitation*, vol. 118, p. e25, Sept. 2017.

[43] B. Murray and R. Kue, "The Use of Emergency Lights and Sirens by Ambulances and Their Effect on Patient Outcomes and Public Safety: A Comprehensive Review of the Literature," *Prehospital and Disaster Medicine*, vol. 32, pp. 209–216, Apr. 2017.

[44] Ordnance Survey, "Getting there quicker with Geotracker — Business and government — Ordnance Survey." https://bit.ly/2DggFKV, 2018.

[45] V. T. N. Nha, S. Djahel, and J. Murphy, "A Comparative Study of Vehicles' Routing Algorithms for Route Planning in Smart Cities," in *2012 First International*

Workshop on Vehicular Traffic Management for Smart Cities (VTM), pp. 1–6, Nov. 2012.

[46] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, Dec. 1959.

[47] B. Zhan, "Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures : JGIDA vol.1, no.1, pp. 69-82." http://publish.uwo.ca/~jmalczew/gida_1/Zhan/Zhan.htm, 1997.

[48] R. W. Floyd, "Algorithm 97: Shortest Path," *Commun. ACM*, vol. 5, pp. 345–, June 1962.

[49] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, pp. 100–107, July 1968.

[50] W. Zeng and R. L. Church, "Finding shortest paths on real road networks: The case for A*," *International Journal of Geographical Information Science*, vol. 23, pp. 531–543, Apr. 2009.

[51] A. Stentz and I. C. Mellon, "Optimal and Efficient Path Planning for Unknown and Dynamic Environments," *International Journal of Robotics and Automation*, vol. 10, pp. 89–100, 1993.

[52] A. Stentz, "The Focussed D* Algorithm for Real-Time Replanning," in *In Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1652–1659, 1995.

[53] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong Planning A_*," *Artificial Intelligence*, vol. 155, pp. 93–146, May 2004.

[54] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Transactions on Robotics*, vol. 21, pp. 354–363, June 2005.

[55] S. Wang, S. Djahel, J. McManis, C. McKenna, and L. Murphy, "Comprehensive Performance Analysis and Comparison of Vehicles Routing Algorithms in Smart Cities," in *Global Information Infrastructure Symposium - GIIS 2013*, pp. 1–8, Oct. 2013.

[56] G. Musolino, A. Polimeni, C. Rindone, and A. Vitetta, "Travel Time Forecasting and Dynamic Routes Design for Emergency Vehicles," *Procedia - Social and Behavioral Sciences*, vol. 87, pp. 193–202, Oct. 2013.

[57] W. Stone, L. Stenneth, and J. alowibdi, "Reducing Travel Time by Incident Reporting via CrowdSourcing," in *The 2011 International Conference on Internet Computing, ICOMP'11*, 2011.

[58] A. Janecek, K. A. Hummel, D. Valerio, F. Ricciato, and H. Hlavacs, "Cellular Data Meet Vehicular Traffic Theory: Location Area Updates and Cell Transitions for Travel Time Estimation," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, (New York, NY, USA), pp. 361–370, ACM, 2012.

[59] GraphHopper, "GraphHopper Directions API with Route Optimization." https://www.graphhopper.com/, 2018.

[60] Stefan, "Time-Dependent Route Optimization API," Nov. 2017.

[61] Google, "Google Maps Distance Matrix API." https://developers.google.com/maps/documentation/distance-matrix/, Nov. 2017.

[62] Waze, "Live Traffic Jam News & Updates Near You, Waze Maps." https://www.waze.com/en-GB/livemap, Nov. 2017.

[63] OSRM, "Project OSRM." http://project-osrm.org/, 2018.

[64] "TomTom Maps - Live Traffic." https://www.tomtom.com/licensing/livetraffic/.

[65] R. J. Fleischman, M. Lundquist, J. Jui, C. D. Newgard, and C. Warden, "Predicting Ambulance Time of Arrival to the Emergency Department Using Global Positioning System and Google Maps," *Prehospital emergency care : official journal of the National Association of EMS Physicians and the National Association of State EMS Directors*, vol. 17, no. 4, pp. 458–465, 2013.

[66] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, "Map-Matching for Low-Sampling-Rate GPS Trajectories," in *ACM SIGSPATIAL GIS 2009*, Association for Computing Machinery, Inc., Nov. 2009.

[67] X. Zhang, Q. Wang, and D. Wan, "The relationship among vehicle positioning performance, map quality, and sensitivities and feasibilities of map-matching algorithms," in *IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No.03TH8683)*, pp. 468–473, June 2003.

[68] W. Ochieng, M. Quddus, and R. Noland, "Map-matching in complex urban road networks," *Brazilian Journal of Cartography*, vol. 55, no. 2, pp. 1–18, 2004.

[69] L. Li, M. Quddus, and L. Zhao, "High Accuracy Tightly-Coupled Integrity Monitoring Algorithm for Map-Matching," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 13–26, 2013.

[70] G. R. Jagadeesh and T. Srikanthan, "Robust Real-Time Route Inference from Sparse Vehicle Position Data," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 296–301, 2014.

[71] G. R. Jagadeesh and T. Srikanthan, "Online Map-Matching of Noisy and Sparse Location Data With Hidden Markov and Route Choice Models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, pp. 2423–2434, Sept. 2017.

[72] R. Mohamed, H. Aly, and M. Youssef, "Accurate Real-Time Map Matching for Challenging Environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–11, 2016.

[73] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk, "On map-matching vehicle tracking data," in *In Proc. 31st VLDB Conference*, pp. 853–864, 2005.

[74] M. Hashemi and H. A. Karimi, "A Critical Review of Real-Time Map-Matching Algorithms: Current Issues and Future Directions," *Computers, Environment and Urban Systems*, vol. 48, pp. 153–165, Nov. 2014.

[75] J. Kim, "Node Based Map Matching System for Car Navigation Systems," in *International Symposium on Automotive Technology & Automation (29th : 1996 : Florence, Italy). Global Deployment of Advanced Transportation Telematics/ITS*, 1996/00/00.

[76] D. Bernstein and A. Kornhauser, "An Introduction to Map Matching for Personal Navigation Assistants," tech. rep., New Jersey Institute of Technology, 1996.

[77] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 15, pp. 312–328, Oct. 2007.

[78] J. Huang, C. Liu, and J. Qie, "Developing Map Matching Algorithm for Transportation Data Center," in *2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pp. 167–170, Nov. 2014.

[79] B. Phuyal, "Method and Use of Aggregated Dead Reckoning Sensor and GPS Data For Map Matching,," in *E Institute of Navigation (ION) Annual Conference*, 2002.

[80] N. R. Velaga, M. A. Quddus, and A. L. Bristow, "Developing an enhanced weight-based topological map-matching algorithm for intelligent transport sys-

tems," *Transportation Research Part C: Emerging Technologies*, vol. 17, pp. 672–683, Dec. 2009.

[81] H. Yang, S. Cheng, H. Jiang, and S. An, "An Enhanced Weight-based Topological Map Matching Algorithm for Intricate Urban Road Network," *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 1670–1678, Nov. 2013.

[82] M. A. Quddus, W. Y. Ochieng, L. Zhao, and R. B. Noland, "A general map matching algorithm for transport telematics applications," *GPS Solutions*, vol. 7, pp. 157–167, Dec. 2003.

[83] G. Hu, J. Shao, F. Liu, Y. Wang, and H. T. Shen, "Towards Accurate Map-Matching with Information Fusion," in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pp. 9–10, Apr. 2017.

[84] K. E. Mokhtari, S. Reboul, M. Azmani, J. B. Choquel, S. Hamdoune, B. Amami, and M. Benjelloun, "A Map Matching Algorithm Based on a Particle Filter," in *2014 International Conference on Multimedia Computing and Systems (ICMCS)*, pp. 723–727, Apr. 2014.

[85] V. Fox, J. Hightower, a. D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *IEEE Pervasive Computing*, vol. 2, pp. 24–33, July 2003.

[86] L. Zhao, W. Y. Ochieng, M. Quddus, and R. Noland, "An Extended Kalman Filter Algorithm for Integrating GPS and Low Cost Dead Reckoning System Data for Vehicle Performance and Emissions Monitoring," *Journal of Navigation*, vol. 56, May 2003.

[87] E. J. Krakiwsky, C. B. Harris, and R. V. C. Wong, "A Kalman filter for integrating dead reckoning, map matching and GPS positioning," in *IEEE PLANS '88.,Position Location and Navigation Symposium, Record. 'Navigation into the 21st Century'.*, pp. 39–46, Nov. 1988.

[88] M. E. El Najjar and P. Bonnifait, "A Road-Matching Method for Precise Vehicle Localization Using Belief Theory and Kalman Filtering," *Autonomous Robots*, vol. 19, pp. 173–191, Sept. 2005.

[89] C. Wang, H. Liang, X. Geng, and M. Zhu, "Multi-sensor fusion method using kalman filter to improve localization accuracy based on android smart phone," in *2014 IEEE International Conference on Vehicular Electronics and Safety*, pp. 180–184, Dec. 2014.

[90] B. Yu, L. Dong, D. Xue, H. Zhu, X. Geng, R. Huang, and J. Wang, "A hybrid dead reckoning error correction scheme based on extended Kalman filter and map matching for vehicle self-localization," *Journal of Intelligent Transportation Systems*, vol. 0, pp. 1–15, Nov. 2018.

[91] Y. Choe, C. G. Park, and J. W. Song, "Importance sampling Kalman filter for urban canyon navigation," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pp. 1264–1269, Apr. 2018.

[92] E. A. Wan and R. V. D. Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, pp. 153–158, Oct. 2000.

[93] M. Grewal and A. Andrews, *Kalman Filtering: Theory and Practice with MATLAB, 4e.* 2015.

[94] F. Pereira, H. Costa, and N. Martinho Pereira, "An off-line map-matching algorithm for incomplete map databases," *European Transport Research Review*, vol. 1, pp. 107–124, Oct. 2009.

[95] P. Newson and J. Krumm, "Hidden Markov Map Matching Through Noise and Sparseness," in *Proceedings of the 17th ACM SIGSPATIAL International Confer-*

ence on Advances in Geographic Information Systems, GIS '09, (New York, NY, USA), pp. 336–343, ACM, 2009.

[96] R. Raymond, T. Morimura, T. Osogami, and N. Hirosue, "Map Matching with Hidden Markov Model on Sampled Road Network," in *2012 21st International Conference on Pattern Recognition (ICPR)*, pp. 2242–2245, Nov. 2012.

[97] H. Wei, Y. Wang, G. Forman, Y. Zhu, and H. Guan, "Fast Viterbi Map Matching with Tunable Weight Functions," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '12, (New York, NY, USA), pp. 613–616, ACM, 2012.

[98] H. Koller, P. Widhalm, M. Dragaschnig, and A. Graser, "Fast Hidden Markov Model Map-Matching for Sparse and Noisy Trajectories," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 2557–2561, Sept. 2015.

[99] M. Ali, T. Rautman, J. Krumm, and A. Teredesai, "ACM SIGSPATIAL GIS Cup 2012," *Microsoft Research*, Dec. 2016.

[100] A. Luo, S. Chen, and B. Xv, "Enhanced Map-Matching Algorithm with a Hidden Markov Model for Mobile Phone Positioning," *ISPRS International Journal of Geo-Information*, vol. 6, p. 327, Oct. 2017.

[101] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260–269, Apr. 1967.

[102] P. Del Moral, "Non Linear Filtering: Interacting Particle Solution," *Markov Processes and Related Fields*, vol. Volume 2 Number 4, 1996.

[103] S. Thrun, "Particle Filters in Robotics," *Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)*, p. 9, 2002.

[104] T. Driver, "Long-Term Prediction of GPS Accuracy: Understanding the Fundamentals," in *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation*, 2007.

[105] K. Par and O. Tosun, "Parallelization of Particle Filter Based Localization and Map Matching Algorithms on Multicore/Manycore Architectures," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 820–826, June 2011.

[106] F. Kuhnt, R. Kohlhaas, R. Jordan, T. Gußner, T. Gumpp, T. Schamm, and J. M. Zöllner, "Particle Filter Map Matching and Trajectory Prediction Using a Spline Based Intersection Model," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1892–1893, Oct. 2014.

[107] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund, "Particle Filters for Positioning, Navigation, and Tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 425–437, Feb. 2002.

[108] J. C. P. Davidson, "Application of Particle Filters to Map-Matching Algorithm," *17th Saint Petersburg International Conference on Integrated Navigation Systems, ICINS 2010 - Proceedings*, vol. 2, no. 4, pp. 285–292, 2011.

[109] F. G. Toro, D. E. D. Fuentes, D. Lu, U. Becker, H. Manz, and B. Cai, "Particle Filter Technique for Position Estimation in GNSS-Based Localisation Systems," in *2015 International Association of Institutes of Navigation World Congress (IAIN)*, pp. 1–8, Oct. 2015.

[110] K. Kempinska, T. Davies, and J. Shawe-Taylor, "Probabilistic Map-Matching Using Particle Filters," *arXiv:1611.09706 [stat]*, Nov. 2016.

[111] N. Drawil, H. Amar, and O. Basir, "A Solution to the Ill-Conditioned GPS Accuracy Classification Problem: Context Based Classifier," in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*, pp. 1077–1082, Dec. 2011.

[112] US Government, "GPS Accuracy." https://bit.ly/1qqPzCJ, Mar. 2017.

[113] J. Newman, J. Chen, and M. Bierlaire, "Generating probabilistic path observation from GPS data for route choice modeling," in *Proceedings of the European Transport Conference*, Jan. 2009.

[114] Diggelen, "GNSS Accuracy: Lies, Damn Lies and Statistics," *GPS World*, 2007.

[115] J. Krumm, E. Horvitz, and J. Letchner, "Map Matching with Travel Time Constraints," in *SAE World Congress & Exhibition*, Apr. 2007.

[116] A. Oran and P. Jaillet, "An HMM-based map matching method with cumulative proximity-weight formulation," in *2013 International Conference on Connected Vehicles and Expo (ICCVE)*, pp. 480–485, Dec. 2013.

[117] M. Rahmani, E. Jenelius, and H. N. Koutsopoulos, "Route travel time estimation using low-frequency floating car data," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp. 2292–2297, Oct. 2013.

[118] D. Taş, N. Dellaert, T. van Woensel, and T. de Kok, "The time-dependent vehicle routing problem with soft time windows and stochastic travel times," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 66–83, Nov. 2014.

[119] M. Schilde, K. F. Doerner, and R. F. Hartl, "Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem," *European Journal of Operational Research*, vol. 238, pp. 18–30, Oct. 2014.

[120] M. Figliozzi, "The Impacts of Congestion on Commercial Vehicle Tours Characteristics and Costs.," *Civil and Environmental Engineering Faculty Publications and Presentations*, Dec. 2007.

[121] A. R. Güner, A. Murat, and R. B. Chinnam, "Dynamic routing under recurrent and non-recurrent congestion using real-time ITS information," *Computers & Operations Research*, vol. 39, pp. 358–373, Feb. 2012.

[122] L. Fu and L. R. Rilett, "Expected shortest paths in dynamic and stochastic traffic networks," *Transportation Research Part B: Methodological*, vol. 32, pp. 499–516, Sept. 1998.

[123] S. Budge, A. Ingolfsson, and D. Zerom, "Empirical Analysis of Ambulance Travel Times: The Case of Calgary Emergency Medical Services," *Management Science*, vol. 56, pp. 716–723, Apr. 2010.

[124] B. S. Westgate, D. B. Woodard, D. S. Matteson, and S. G. Henderson, "Large-Network Travel Time Distribution Estimation for Ambulances," *European Journal of Operational Research*, vol. 252, pp. 322–333, July 2016.

[125] H. Wang, Z. Li, Y.-H. Kuo, and D. Kifer, "A Simple Baseline for Travel Time Estimation Using Large-Scale Trip Data," *arXiv:1512.08580 [cs]*, Dec. 2015.

[126] G. Kim, "Travel time estimation in vehicle routing problem," in *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 1004–1008, Dec. 2017.

[127] P. Groß, M. Geisinger, J. F. Ehmke, and D. C. Mattfeld, "Interval Travel Times for More Reliable Routing in City Logistics," *Transportation Research Procedia*, vol. 12, pp. 239–251, Jan. 2016.

[128] Y. Guessous, M. Aron, N. Bhouri, and S. Cohen, "Estimating Travel Time Distribution under Different Traffic Conditions," *Transportation Research Procedia*, vol. 3, pp. 339–348, Jan. 2014.

[129] S. K. Singh and G. S. Maddala, "A Function for Size Distribution of Incomes," *Econometrica*, vol. 44, no. 5, pp. 963–970, 1976.

[130] M. R. Daya, R. H. Schmicker, D. M. Zive, T. D. Rea, G. Nichol, J. E. Buick, S. Brooks, J. Christenson, R. MacPhee, A. Craig, J. C. Rittenberger, D. P.

Davis, S. May, J. Wigginton, and H. Wang, "Out-of-Hospital Cardiac Arrest Survival Improving over Time: Results from the Resuscitation Outcomes Consortium (ROC)," *Resuscitation*, vol. 91, pp. 108–115, June 2015.

[131] R. Graham, M. A. McCoy, A. M. Schultz, C. o. t. T. o. C. A. C. S. Directions, Future, B. o. H. S. Policy, and I. of Medicine, *Emergency Medical Services Response to Cardiac Arrest*. National Academies Press (US), Sept. 2015.

[132] M. L. Weisfeldt and L. B. Becker, "Resuscitation after Cardiac Arrest: A 3-Phase Time-Sensitive Model," *JAMA*, vol. 288, pp. 3035–3038, Dec. 2002.

[133] NHS England, "Ambulance Quality Indicators Data 2017-18." https://bit.ly/2FGwJF9, 2018.

[134] LAA, "London's Air Ambulance." https://londonsairambulance.co.uk, 2018.

[135] BT, "Enhanced Information Service for Emergency Calls," Oct. 2018.

[136] J. Hunt, "The ambulance service in England:Written statement - HCWS201." https://bit.ly/2PSfKqH, 2015.

[137] K. Willet, "Clinical review of ambulance responses in England: Advice to Secretary of State." https://www.england.nhs.uk/wp-content/uploads/2015/01/clincl-rec-amblncs.pdf, Jan. 2015.

[138] NHS England, "Ambulance Response Programme Review." https://www.england.nhs.uk/wp-content/uploads/2018/10/ambulance-response-programme-review.pdf, Nov. 2018.

[139] OpenStreetMap, "OpenStreetMap." https://www.openstreetmap.org./, 2018.

[140] Here, "Location-Based Services for the Autonomous Future." https://www.here.com, 2018.

[141] TomTom, "TomTom MultiNet." https://www.tomtom.com, June 2018.

[142] Ordnance Survey, "Britain's mapping agency." https://www.ordnancesurvey.co.uk/, 2018.

[143] R. Lott, "Geographic information - Well-known text representation of coordinate reference systems." http://docs.opengeospatial.org/is/12-063r5/12-063r5.html, May 2015.

[144] Ordnance Survey, "The National Grid." https://bit.ly/2rStMxX, Oct. 2018.

[145] R. Alanis, A. Ingolfsson, and B. Kolfal, "A Markov Chain Model for an EMS System with Repositioning," *Production and Operations Management*, vol. 22, pp. 216–231, Jan. 2013.

[146] R. Finkel and J. Bentley, "Quad trees a data structure for retrieval on composite keys," *Informatica*, Mar. 1974.

[147] R Core Team, *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2017.

[148] A. C. Cullen and H. C. Frey, *Probabilistic Techniques in Exposure Assessment: A Handbook for Dealing with Variability and Uncertainty in Models and Inputs*. Springer Science & Business Media, 1999.

[149] Ripley, "Maximum-likelihood Fitting of Univariate Distributions." http://astrostatistics.psu.edu/su07/R/html/MASS/html/fitdistr.html, Apr. 2018.

[150] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, pp. 53–82, July 2010.

[151] T. Li, S. Sun, T. P. Sattar, and J. M. Corchado, "Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches," *Expert Systems with Applications*, vol. 41, pp. 3944–3954, June 2014.

[152] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings F - Radar and Signal Processing*, vol. 140, pp. 107–113, Apr. 1993.

[153] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 50, no. 2, p. 15, 2002.

[154] G. B. Dantzig and J. H. Ramser, "The Truck Dispatching Problem," *Management Science*, vol. 6, pp. 80–91, Oct. 1959.

[155] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, pp. 308–313, Jan. 1965.

[156] M. L. Fredman and R. E. Tarjan, "Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms," *J. ACM*, vol. 34, pp. 596–615, July 1987.