Sequence modelling using deep learning approaches for spatiotemporal public transport data

Thilo Reich

Doctor of Philosophy

Bournemouth University

2022

This thesis is dedicated to everyone who is still waiting for their bus.

Abstract

Encouraging the use of public transport is essential to combat congestion and pollution in an urban environment. To achieve this, the reliability of public transport arrival time prediction should be improved, as this is often requested by passengers. This will make the use of urban bus networks more convenient for passengers and, thus, will play a crucial role in shifting traffic to public transport. Ultimately, this will alleviate pollution and congestion and save a substantial amount of cost to society associated with the use of private cars. Here, the overarching objective was to investigate novel prediction methods and improve predictions for urban bus networks with a focus on short-horizon predictions.

ETA predictions are unreliable due to the lack of good quality historical data, while 'live' positions in mobile apps suffer from delays in data transmission. The assessment of different of data quality regimes on the next-step prediction accuracy of Recurrent Neural Networks (RNN) showed that that without data cleaning, model predictions can give false confidence if mean errors are used, highlighting the importance of a holistic assessment of the results. It was demonstrated that noisy data is a problem and simple but effective approaches to address these issues are discussed. It became apparent that RNNs are exceptionally good at predicting stationary positions at either end of a journey. The maximum model improvement of the Sharpe ratio compared to noisy data was 4.71%. This provides insight into the value of addressing data quality issues in urban transport data to enable better predictions and improve the passenger experience.

Furthermore, a comparison of different target representations was tested by encoding targets as unconstrained geographical coordinates, progress along a known trajectory, or ETA at the next two stops. The target representation was shown to affect the accuracy of the prediction by constraining the prediction space and reduced the prediction error from 244.8 to 142.3 m for the Long Short-Term Memory (LSTM) network. This error was further reduced if an ETA was predicted and if a distance is estimated from the ETA error resulted in a a reduction to 4.5 and 14.5 m for the next 2 stops on the route.

Due to the observed lack of data quality, a method was to developed for synthesising data, using a reference curve approach derived from very limited real-world data without a reliable ground truth. This approach allows the controlled introduction of artefacts and noise to simulate their impact on prediction accuracy. To illustrate these impacts, a RNN next-step prediction was used to compare different scenarios in two different UK cities. Two model architectures were used as comparison: a Gated Unit and a LSTM model. Hybrid data was generated where

real-world and synthetic data was mixed. When compared to the inference of a model trained purley on synthetic data, the error was reduced from 53.5 to 47.4 m for the LSTM and from 53.4 to 44.0 m for the GRU. The results show that realistic data synthesis is possible, allowing controlled testing of predictive algorithms.

Urban traffic networks are interconnected systems that behave in complex ways to any disturbance. As urban buses operate in such networks and are influenced by traffic within this system, estimated arrival time (ETA) predictions can be challenging and are often inaccurate. To enable the use of network-wide data, a novel model architecture was developed. This attention-mechanism based predictor incorporated the states of other vehicles in the network by encoding their positions using gated recurrent units (GRU) of the individual bus line to encode their current state. By muting specific parts of the imputed information, their impact on prediction accuracy were estimated on a subset of the available data. The results showed that a network-based predictor outperforms models based on a single vehicle or all vehicles of a single line.

However, a model limited to vehicles of the same line ahead of the target was the best performing model, suggesting that the incorporation of additional data can have a negative impact on the prediction accuracy if it does not add any useful information. This could be caused by poor data quality, but also by a lack of interaction between the included lines and the target line. The technical aspects of this architecture are challenging and resulted in a very inefficient training procedure. It can be expected that if a more efficient training regime is developed or the model is trained for a longer time, usable predictive accuracy can be achieved.

Acknowledgements

First and foremost, I would like to thank my supervisors Prof Marcin Budka and David Hulbert for their guidance support and inspiration, and their input in my many drafts. Thank you for your advice over the last few years. My special thanks go to Marcin who made it possible for me to start my adventure into computer science and who guided me through my development process. Thank you also to Passenger Ltd. for match funding this work.

I would like to thank my parents, who always supported me and encouraged me during my very long and changing educational journey.

I also want to thank Olivia for all the support, keeping me sane and making sure that I did not forget to go outside.

Declaration

I Thilo Reich (TR) confirm that the research presented within this thesis is my own. The following chapters were, however, published or prepared for publication in collaboration with Marcin Budka (MB) and David Hulbert (DH):

Chapter 3:
Thilo Reich, Marcin Budka, and David Hulbert. “Impact of Data Quality and Target Representation on Predictions for Urban Bus Networks,” in 2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020. IEEE, 12 2020, pp. 2843–2852.
TR, MB and DH conceived and designed the investigation. TR analysed the data and developed the algorithms. TR wrote the paper. TR, MB and DH revised the paper.

Chapter 4:
Thilo Reich, Marcin Budka, and David Hulbert. "Bus journey simulation to develop public transport predictive algorithms", Soft Computing Letters, 3:100029, 2021.
TR, MB and DH conceived and designed the investigation. TR analysed the data and developed the method. TR wrote the paper. TR, MB and DH revised the paper.

Chapter 5:
Thilo Reich, Marcin Budka, and David Hulbert. "A Model Architecture for Public Transport Networks Using a Combination of a Recurrent Neural Network Encoder Library and a Attention Mechanism", Algorithms, 2022.
TR, MB and DH conceived and designed the investigation. TR analysed the data and developed the method. TR wrote the paper. TR, MB and DH revised the paper.

Contents

Figures and Tables

Figures

Tables

Chapter 1

Introduction

1.1 Background motivation

Anyone who has ever used any type of public transport will appreciate the usefulness of accurately estimated arrival time (ETA) predictions. This is highlighted in surveys in which passengers rated accurate ETAs as one of the most important areas of improvement in the public transport sector [1]. Previous work in the same geographical area where the data for this study were collected showed that the bus ETA predictions were substandard and that there is room for improvement in accurate ETAs [2]. Although this information is not publicly available due to the proprietary setup of British bus networks, it is believed from conversations with commercial partners of this study that currently very simple methods are used to make ETA predictions, such as adding the current delay to the timetabled arrival times. The motivation of this doctoral thesis was based on this information and the intuition that more sophisticated deep learning-based methods should allow one to improve on these currently applied simplistic methods, especially if the entire transport network is considered in the prediction. The choice of target was based on the observation that the available "live" position showing vehicle locations on the operator's website and a mobile phone app was, in fact, delayed by approximately 30-40 s. Therefore, the next-step prediction was chosen as one prediction target to pose a directly practical application of any algorithm developed in bringing the currently delayed "live" position closer to a true live location. This would have a direct positive impact on the customer experience. Furthermore, as discussed, accurate ETA predictions are often requested by customers and therefore are used as a secondary target and compared to the next-step predictions.

The literature review in Chapter 2 gives an in-depth background on the current literature and highlights the fact that urban bus prediction methods lack reproducible and a standardised systematic approach to compare algorithms developed in different studies. It was observed that from 2018 the overall quality of the reviewed publications increased, suggesting a maturation of the research area. However, the fact that most publications have single or only very few lines as test problems and that in some cases the data are specifically collected, as presumably routine data are not available, suggests that gaining access to high-quality data is challenging.

In theory, this should be a straightforward prediction problem; however, in reality, the data lack information and integrity which has also been reported in other studies [3]. The data standards that should be adhered to in any UK bus network are outlined in a UK national Extensible Markup Language (XML) based standard called TransXChange [4]. The guideline provides information on how and what data can be made available by bus operators. This includes bus schedules, departure frequencies, and many operational details. This aims to make the data from buses accessible and exchangeable. Other standards are also defined by the government, for example, access nodes to public transport such as bus stops and train stations in a comprehensive and constantly updated list, the National Public Transport Access Nodes (NaPTAN) [5].

1.2 The bus data structure

In the following paragraphs the ideal data structure and availability is outlined, which should be available according to the government defined standards. For each bus line, the stop locations of the NaPTAN schema should be described with descriptions that contain the coordinates as well as the name and a specific route ID. This refers to an entire route and should allow linking this route to any related data and background information.

Each route and vehicle should be linked to the following data fields:

- A Service
- A Timetable
- A Journey Pattern
- A Route Section
- A Vehicle Journey

These data fields and how they are linked to each other will be discussed in the following paragraphs; a flow chart of the XML schema and how these data are linked is shown in Figure 1.1.

A **Service** describes a specific service, for example, the outbound line 1. This also describes the operating profile, for example, whether a service runs on public holidays, weekends etc. This links to the timetable which allows to look up the run times of each service as well as bus stop codes that can be linked to their name as well as coordinates.

The service also references a **Journey Pattern**, this pattern allows to read the runtimes for an individual **Route Section** each defining the route section between several stops with a unique ID. This route section is divided into **Route Links**, shows the origin and destination stops of each of the links using their unique stop code and give a distance in meters between these locations. By combining all the distances, the total distance for each segment can be calculated. To get the arrival times at each of the bus stops, the route links can be back-referenced to the route section, which will give the runtimes for each of the sections.

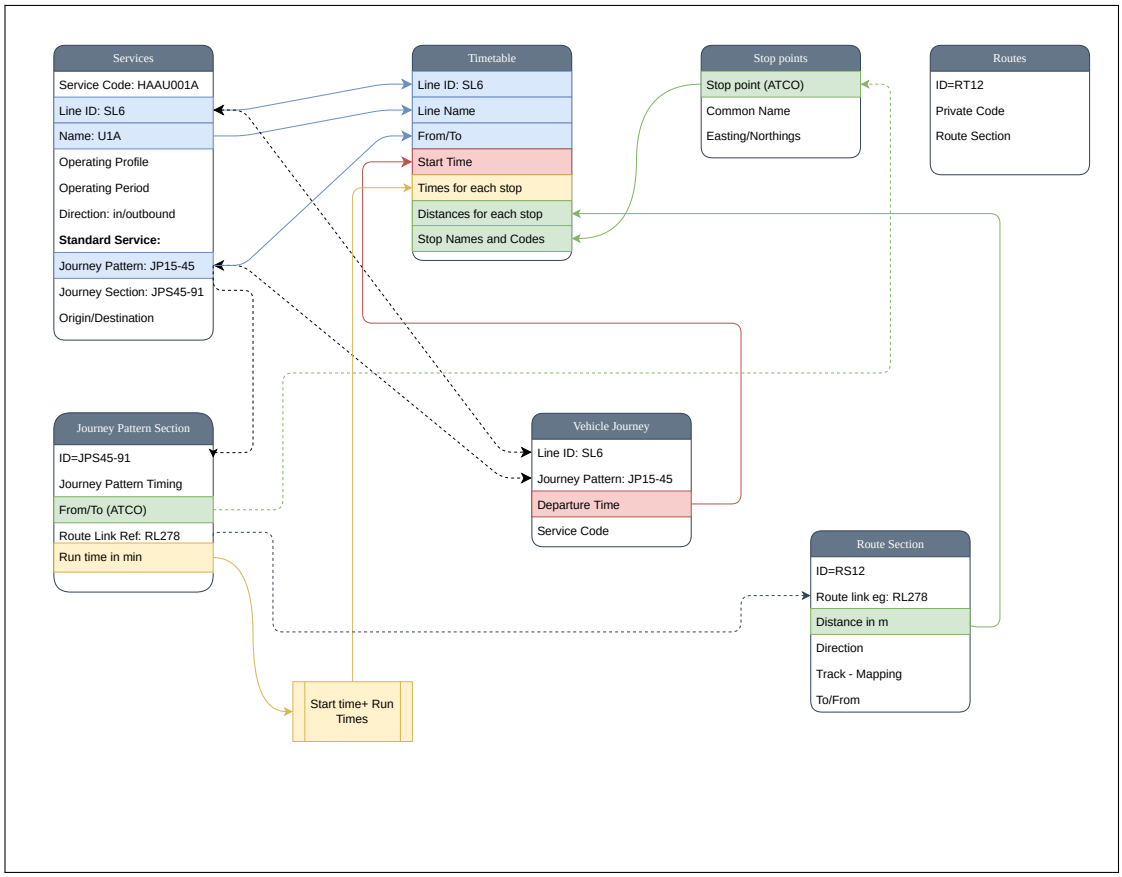


Figure 1.1: Overview of the data structure according to the TransXChange standard.

The journey pattern also allows one to link to the **Track** corresponding to each Route Link and section that contains the coordinates and eastings and northings of the road section between the two stops. The coordinate resolution differs depending on whether the road section is straight or winding.

This information should allow to get all necessary information about a bus service within a city or region. The final step is to link this information to an active vehicle that is operating in a network. To achieve this, each bus should identify itself with a vehicle number or license plate, the line name, and crucially a **Journey Reference**. This Journey Reference should be a unique reference code linking a vehicle to a service, and thus its timetable and journey pattern, for example, the 8 am journey of line number 1. In reality, this identifier was not found to be unique in the XML schema for the areas available in this study, making the reliable linking using this method impossible. A second approach that used the ID of the ticket machine, which is the device used to sell bus tickets but also houses the Automatic Vehicle Location (AVL) system. This device ID is also reported for each vehicle journey in the XML schema, as long as the correct vehicle is serving a specific journey and no short-term operational changes have been made.

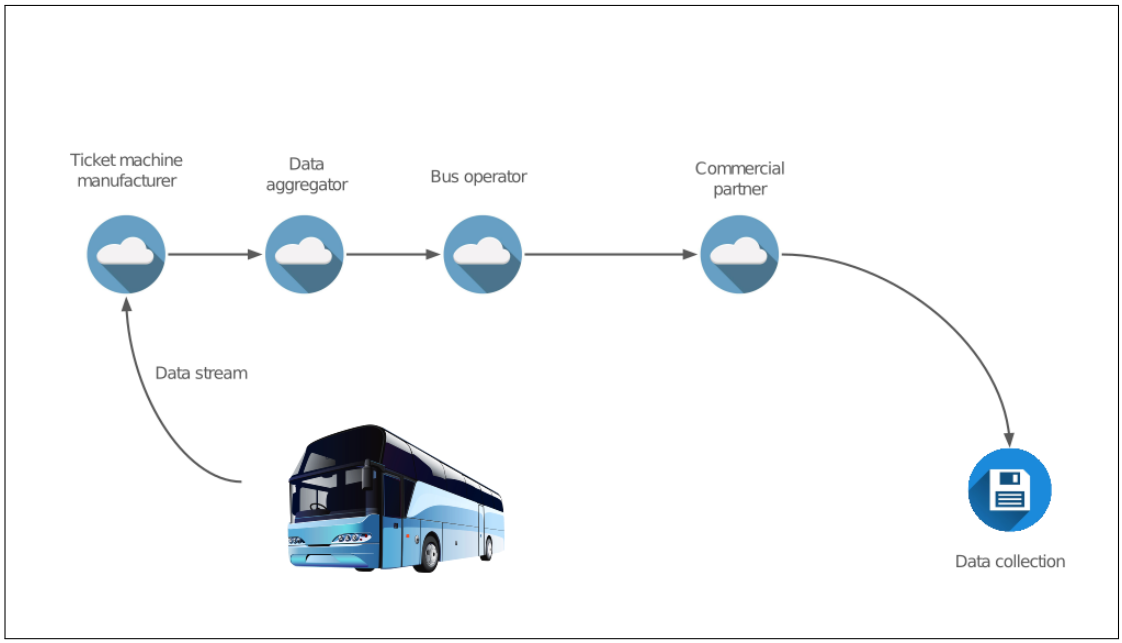


Figure 1.2: The path of the data from the transmitting vehicle via several data brokers to the point where it was recorded. During this process, significant amounts of information were lost, making the matching of the vehicle with a schedule or route pattern impossible.

Although complicated, this data structure should allow one to report, interpret, and use any data from bus transport networks in the government-defined framework. Another technical guideline by the Department of Transport defines how real-time data should be transmitted by any connected vehicle. The Service Interface for Real-Time Information (SIRI) is a European standard for the exchange of information on the planned, current, or projected performance of real-time public transport operations between different computer systems [6]. This live feed should contain the reference for the trip and the reference for the ticket machine, crucial for linking a vehicle with other information about the bus service. This is also the data stream that was made available by the commercial partners for this study. However, since the data stream was passed through the hands of several data brokers, most of the information was not available for this study (Figure 1.2). This is also the hypothesised reason for the observed transmission latency of the "live" locations. The available data stream did not contain the journey reference nor the ticket machine ID making it impossible to link a vehicle to any part of the XML schema. To make things more difficult, the vehicles did not reliably report line number nor direction of a line, effectively leaving only GPS coordinates as usable data. This problem has also been mentioned by [3]. To limit the data, the reported line number had to be used in combination with a heuristic approach to exclude any vehicles that were running on entirely different routes identifying with the wrong line number. However, this, of course, does not pose a good start to collect high-quality data. In addition, a bus line in all the cities investigated can have many different journey patterns, which differ in the exact route a vehicle

takes, the number of stops it serves and the length of the route it serves. This effectively means that each line has many sub-lines all identifying with the same line name but which will behave very differently. If a vehicle could have been linked to a journey pattern, high quality data could have been collected which would not only have linked a vehicle to the timetable and journey patterns but would have also allowed to gather information about delays and arrival times at bus stops. As this however, was not the case the collected data significantly suffered from quality issues.

In an attempt to collect the best possible data, live data streams of a total of 8 UK bus operators were considered in the following geographical areas:

- Blackpool
- Bournemouth
- Cardiff
- Isle of Wight
- Reading
- Salisbury
- Southampton
- Swindon

None of these operators transmitted sufficient data to allow matching any of the journeys, thus not allowing to address the data quality issues. For the studies presented in this thesis, two cities were selected, Bournemouth and Reading. These were selected because the data transmission rate was high, thus allowing one to collect data at a higher frequency compared to the other cities.

This challenge of collecting high-quality data is described in Chapter 3, where the impact of extremely low-quality data is discussed. Through the corporate partners of this study, the available data was very limited due to the fragmentation of the British public transport network, where each step of the data transmission is compartmentalised and delivered by different commercial entities with differing commercial goals. As a result, a heuristic approach had to be developed to extract basic information such as when a journey might have started or ended. This, of course, is a sub-optimal starting point to build any predictive algorithm upon. Furthermore, erroneous artefacts that can only be explained by software versions of the ticketing machine, which includes the GPS unit, were observed in data from some vehicles. The manufacturer was approached several times for comment but never responded. Due to the limited data a ground truth could not be established but this chapter demonstrates some improvements of a deep learning ETA prediction method in comparison to a naive average speed prediction specifically at stationary positions of a vehicle.

Building onto these discovered challenges a synthetic data generator is described in Chapter 4 to partially circumvent the data quality issues. Naturally, to make a data-informed data synthesiser, the low-quality data is the initial position upon which the data generation has to be built. This approach was chosen as the aim of this thesis is to explore practically useful methods instead of purely theoretical frameworks. Therefore, the described data generator uses minimal prior-assumptions and through a number of heuristic steps generates the best possible approximation of the available data based on observed historical behaviour. Several combinations of data were tested and it is shown that the combination of real-world data and synthetic data can improve the inference capabilities on real-world data; thus, demonstrating a practical application of this study in scenarios where no high quality data are available. Alongside this main subject, Chapter 4 also demonstrates the effect of limiting the predictive space in order to improve the prediction accuracy. This can simply be achieved by limiting the possible predictions to positions along the route ahead of the vehicle as buses should not be changing direction before the end of their journey.

Finally, Chapter 5 describes a predictive framework leveraging the state of the entire transport network, by combining data from several lines that were deemed to interact within the network and training several different recurrent neural networks for each of the individual lines. These multidimensional outputs are then combined using an attention mechanism to make the final prediction. This technically challenging approach gives some interesting insights into the possible benefits of using a network-state based approach by showing that for ultra-short horizon predictions the network state is not relevant and by focusing on vehicles of the same line ahead of the target the best predictions can be made. When it comes to predictions of the ETA at the final stop the network state became relevant in the explorative results. This study highlights the need for the development of a more computationally efficient implementation of this method. This can become an interesting avenue to explore in future research if the data quality issues are resolved. As will be demonstrated throughout the following experimental chapters, the available data was of low quality and thus contained very limited real information [7]. Even though the method demonstrated in Chapter 5 intuitively should improve any long horizon prediction, the colloquial concept of Garbage in, Garbage Out (GIGO) [8] inhibits the practical use and evaluation [9].

In this light, the final conclusion in Chapter 6 highlights the limitations of the individual chapters and brings their results into context with recommendations for further work once high-quality and reliable data become available.

1.3 Aims and Objectives

This thesis aims (i) to assess the potential of the available data, by removing some of the known artefacts within the data and (ii) assess the impact of those on any next step prediction made. Furthermore, (iii) comparison of the different effects of varying target representations on the predictive performance was made. Secondly (iv), a method to synthetically generate data was developed to simulate data and enhance predictive models by training these on hybrid data through the mixing of synthetic and real-world data. Finally (v), a method was developed to make next step and ETA predictions based on network-wide data by incorporating several vehicles operating within the bus network into the model data and, thus, make the predictor aware of the overall state of the public transport system.

1.4 Contributions

The contributions of this thesis are the demonstration of a heuristic approach to recover information from low quality public transport data records. Data processed with this heuristic approach was then systematically analysed in an ablation study to compare the effects of different data cleaning regimes. The insights from this study give a thorough overview of data issues faced by public transport data and their effect on prediction accuracies in short horizon position predictions. Furthermore, building onto the insights gained from this a method to synthetically generate data as close to reality as possible is demonstrated. This method allows to introduce artefacts in a controlled manner and could be a useful tool for other researchers. Furthermore, it was demonstrated that including some synthetic data into low-quality real world data can improve overall generalisation capabilities of a short horizon predictive algorithm. Finally a method is proposed, which uses the network state of a urban bus network to make short horizon positional predictions as well as ETA predictions. This could be further developed and could in the future outperform the current state of the art.

1.5 Organisation of the thesis

This thesis is presented in an integrated format, in which the material is incorporated in a suitable style for submission and publication in a peer-reviewed journal. Thus, the experimental chapters (Chapters 3 to 4) are presented as original, complete, and published research. The review of the literature and the final experimental chapter (Chapter 2 & 5) are presented as pre-submission manuscripts. This thesis format has been chosen as it provides flexibility around the types, numbers, and content of papers included in the thesis. The final chapter (Chapter 6) discusses the implications of this research and concludes the thesis. A complete list of references is provided at the end of the thesis to improve readability.

Each experimental chapter begins with an introduction and an in-depth description of the background literature. Chapter 2 consists of a dedicated systematic review of the literature that introduces the reader to the topic and the current state of research. To save the reader from quintuple repetition of the introduction to the literature, this thesis introduction is kept brief and will focus on laying out the structure of the thesis and linking the chapters, which are again linked retrospectively in the final discussion (Chapter 6). Furthermore, it will give an introduction to the technical aspects underlying this study that are specific to the bus networks of the operators' data, which was available for this study.

Chapter 2

Survey of ETA prediction methods in public transport networks

Chapter overview

The majority of public transport vehicles are fitted with Automatic Vehicle Location (AVL) systems generating a continuous stream of data. The availability of this data has led to a substantial body of literature addressing the development of algorithms to predict Estimated Times of Arrival (ETA). Here, research literature reporting the development of ETA prediction systems specific to buses is reviewed to give an overview of the state of the art. Generally, reviews in this area categorise publications according to the type of algorithm used, which does not allow an objective comparison. Therefore, this survey will categorise the reviewed publications according to the input data used to develop the algorithm. This review highlights inconsistencies in reporting standards of the literature. The inconsistencies were found in the varying measurements of accuracy preventing any comparison and the frequent omission of a benchmark algorithm. Furthermore, some publications were lacking in overall quality. Due to these highlighted issues, any objective comparison of prediction accuracies is impossible. The bus ETA research field therefore requires a universal set of standards to ensure the quality of reported algorithms. This could be achieved by using benchmark datasets or algorithms and ensuring the publication of any code developed.

2.1 Introduction

The UK has seen a constant rise in vehicles on its roads since personal cars have become available. This resulted in a 7-fold increase in traffic on British roads between 1950 and 2016 [10]. This has naturally led to an increase in congestion felt by all road users. In a recent report, it was estimated that UK travellers spent 10% of their driving time in gridlock [11]. A reduction of congestion has become a key priority as it will have a positive impact on the environment, the economy and will reduce commute times. This has been recognised for example in the UK government's 'Road to Zero' strategy aiming to tackle emissions from

road usage. The biggest environmental and societal impact can be achieved if the public is encouraged to use alternative modes of travel instead of private cars [12]. This review is focused on public buses as 4.44 billion bus journeys are made annually in the UK. Despite this, the patronage is declining and better Estimated Time of Arrival (ETA) predictions could play a role in slowing down this trend. It has been shown that even small changes in traffic can have a significant impact on the overall congestion of a city as highlighted by the fact that reducing daily commutes from specific neighbourhoods by only 1% can cut delays for all road users by as much 18% [13]. Even if cancelled commutes are randomly selected, delays can still be reduced by up to 3%. To encourage road users to change their mode of transportation, public transport has to be convenient and reliable. Punctuality and timeliness of the journey have the biggest impact on passenger satisfaction [14]. Non-surprisingly, the most frequently requested improvements by passengers are accurate travel times both pre-trip and during the journey, especially for passengers using public transport to commute [1].

To provide this punctuality, buses should ideally adhere to a timetable that has been carefully designed to allow the bus to meet it without introducing too many buffer times to lengthening the journey unnecessarily. However, this is often difficult and, therefore, it is crucial to accurately predict the arrival times of vehicles. This will improve passenger satisfaction even if the vehicle is late as passengers, in general, do not mind waiting as long as they know how long the expected delay is [15]. Furthermore, reliable real-time travel information provided to passengers reduces the perceived waiting time for bus passengers, as well as the actual waiting time as passengers can arrive closer to the departure time [16]. Furthermore, it will allow developing new smart applications allowing to offer personalised journey suggestions to the traveller. Because buses are affected by a large number of external influences such as weather, traffic conditions, passenger loads [17] and other types of disruptions, predicting their arrival is challenging and therefore currently not very accurate [2]. Methods for predicting ETA can include simple historical averages or statistical models. Therefore, such techniques applied to bus ETA predictions can be expected to drastically improve current performance. However, due to the complexity of the ETA prediction machine learning methods have become increasingly popular [18]. In recent years, Artificial Neural Networks (NN) have revolutionised a number of other domains. Therefore, NNs should be expected to have the same potential when applied to bus ETA prediction problems. A comprehensive review that specifically investigated the applications of NN in public transport [19] found that only 16% (12) addressed the ETA of buses, while the rest of the studies applied the technique to other modes of transport. This suggests that the area of prediction of ETA for buses using NNs might be underrepresented in the context of public transport research. Today, most buses have onboard Automatic Vehicle Location (AVL) systems, which are equipped with GPS sensors and transmit the location of the bus at frequent intervals, typically between 20 and 60 s. The availability of vehicle locations is the basis for any ETA prediction and should be readily accessible through the AVL systems without any additional investment in static sensors. The

general approach of published reviews of ETA prediction methods is to categorise studies by area of application, by the technique used as in [19] or by the applied algorithm [20, 18]. This review will evaluate the current literature on ETA prediction for buses. In doing so, it will demonstrate a more informative categorisation than commonly used to review the literature and address shortcomings of the reporting standards.

2.2 Search strategy

This review includes studies that focus on the prediction of bus ETA in public transport networks, with the aim of predicting the arrival at a bus stop or a point along the journey. The search terms used were "bus, prediction, arrival time, public, algorithm" in combination with any of the following terms: "Algorithm, Neural Networks or NN, Long Short Term Memory or LSTM, Deep learning or DL, Recurrent Neural Network or RNN". The initial search was conducted in 2018 spanning publication dates from 1999-2018. The search was repeated in March 2022 and the publication dates from 2018 onward were included in the second review. For both searches, the 50 highest-ranked publications were screened for context initially by their title followed by a second screening based on the abstract. Additional articles were included from a snowball search based on the reference of the included studies. Based on these inclusion criteria, studies that aim to predict ETA, for example, at junctions to streamline bus signal timings, were excluded [21]. Furthermore, studies that predicted other metrics related to public transport, such as passenger load [22] or operational predictions such as reliability of buses [23] and financial predictions [24] were also excluded. The search was carried out using mySearch (a proprietary library search system used at Bournemouth University). Moreover, we exclude our own publications. As this review aims to establish an overview of practical applications, two studies based on simulated data were excluded [25, 26].

2.3 Categorisation of ETA prediction algorithms

ETA prediction methods are commonly reported as categorised reviews of the literature based on the type of algorithm used as suggested in [18, 27]. This categorisation is not necessarily informative to the reader, as the algorithms can be developed based on different background information – different input features such as locations, speed, and passenger load of the vehicle are used to develop the algorithm, which prevents any meaningful comparison. Therefore, approaches that were developed using only AVL data should in most cases not be compared to methods that also account for passenger load and weather conditions, even if they might be based on the same algorithm. Typically, AVL data include vehicle position, schedule, and route identifiers, but can include more information depending on the provider. This would compare algorithms that rely on an entirely different amount of information, thus

preventing a meaningful interpretation. As this review's focus lies on the prediction of bus ETA, the reviewed studies are categorised based on the nature of the input features used. The most basic requirement of input features to predict ETAs are sequences of time-stamped GPS coordinates recorded by AVL systems (n=25). These features were used by all 53 reviewed publications (also see Supplementary 1). The different feature sources were found to be external data, such as traffic or weather information (n = 9), passenger information such as load and embarking and disembarking numbers (n=4), and a combination of the three sources mentioned above (n = 3). A separate group of studies used AVL information from the bus to be predicted in combination with AVL data of other buses serving the same route to calculate the headway (n=7). And finally, some studies used trajectories either as a prediction base or to extract input features from (n=5).

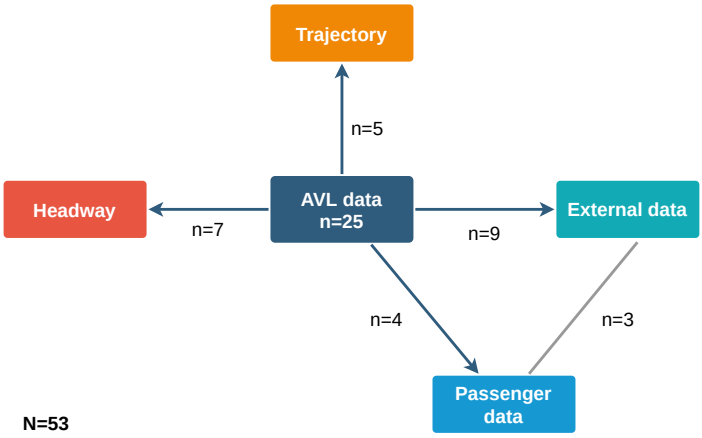


Figure 2.1: Categories are used to review the literature on the basis of feature types.

Finally, 4 studies used AVL data to choose a historical bus trajectory that best represented the current bus progress. One study [39] used trajectories to extract other features. These 5 different types of input features were used as categories in this review rather than the common classes based on the type of algorithm (Figure 2.1 and Table 2.1). This will allow a more objective comparison of the individual categories and will include all types of algorithms. By doing so, we have examined methods that benefit from related input information and thus have arguably more similarity than algorithms that are not based on the same data inputs.

Publication	AVL	External	Trajectories	Passenger	Headway
Amita et al. (2015) [28]	•				
Bai et al. (2015) [29]	•				•
Celan et al. (2018) [30]	•				
Chen (2004a) [31]	•				
Chen et al. (2004b) [32]	•	•		•	
Chen (2018) [33]	•				
Chien et al. (2002) [34]	•	•		•	
Dailey et al. (2001) [35]	•				
Deng et al. (2013) [36]	•	•			
Dong et al. (2013) [37]	•		•		
Gal (2017) [38]	•				•
He et al. (2019) [39]	•		•		
Heghedus (2017) [40]	•				
Hua et al. (2017) [41]	•				•
Jalaney et al. (2021) [42]	•	•		•	
Jeong & Rilett (2004) [43]	•				
Julio et al. (2016) [44]	•				
Junyou et al. (2018) [45]	•	•			
Kee et al. (2017) [46]	•				
Khosharavi et al. (2011) [47]	•				
Kumar et al. (2017) [48]	•		•		
Li (2018) [49]	•	•			
Lin & Zeng (1999) [50]	•				
Lin et al. (2013) [51]	•				•
Liu et al. (2020) [52]	•				
Maiti et al. (2014) [53]	•				
Meng et al. (2017) [54]	•				
Nappiah et al. (2009) [55]	•				
Nedeeshan et al. (2021) [56]	•	•			
Nimpanomprasert et al. (2022) [57]	•	•			
Padmanaban et al. (2009) [58]	•				
Pan et al. (2012) [59]	•				
Petersen et al. (2019) [60]	•				
Shalaby & Farhan (2003) [61]	•			•	
Shalaby & Farhan (2004) [62]	•			•	
Sinn et al. (2012) [63]	•		•		
Toparia et al. (2021) [64]	•				
Treethidtaphat et al. (2017) [27]	•				
Vanajakshi et al. (2009) [65]	•				
Wang et al. (2014) [66]	•			•	
Wu et al. (2020) [67]	•				
Xinghao et al. (2013) [17]	•	•			
Xu (2017) [68]	•		•		
Ye et al. (2021) [69]	•				
Yin et al. (2017) [70]	•				•
Yu et al. (2010) [71]	•				
Yu et al. (2011) [72]	•				•
Yu et al. (2017) [73]	•				•
Zaki et al (2013) [74]	•	•			
Zhang et al. (2015) [75]	•[1]			•	
Zeng et al. (2019) [76]	•				

[1] authors used a modified smartphone instead of a commercial AVL system.

Table 2.1: The input features used by each publication indicated as points.

2.3.1 AVL as sole data source

A minimum requirement to allow any ETA prediction is knowledge about the position of a vehicle; hence, most reviewed studies used AVL data from onboard devices. The only exception was [75], where the locations were recorded using a modified mobile phone, as the buses were not equipped with a GPS system. The reviewed studies used data, including time-stamped bus positions, and in some cases additional information was explicitly calculated, such as average speeds [50] or dwell times [31]. Therefore, this central group of features was the most common, and thus also includes the widest range of applied techniques. The simplest ETA prediction based solely on AVL data are historical methods using the average speed from historical records to predict the arrival time at a destination [50]. Naturally, these cannot account for any fluctuations and thus perform with up to 9.3% lower accuracy compared to more intricate methods such as Kalman Filters (KF) [48]. Attempts to improve simple historical mean-based algorithms, such as accounting for timed stops at which the timetable has deliberate waiting times, reduce the prediction deviation by 0.8% [50]. Another approach was used in which the prediction was made using the historical average updated with exponential smoothing for several short sections of the route, which are then combined to give the total travel time [54]. In the search for an algorithm with better performance and lowest computational impact [53], compared a historical average method, Artificial Neural Networks (NN), and Support Vector Machines (SVM). The results suggest that the NN outperformed historical methods with a minuscule advantage, although the exact value of the improvement is not reported. The authors' conclusion is that as the NN and the historical method perform similarly, yet the NN requires more intensive training and longer prediction times, the historical method is superior [53]. However, the general consensus of the literature on historical methods is that their performance is low [61, 62, 65].

Kalman Filters (KF) are a statistical method that has been applied to bus arrival times [35, 77, 58] and was found to perform with better accuracy compared to historical methods (maximum relative error of 0.543 of the historical approach and 0.087 for the Kalman Filter) [61, 62, 65]. The autoregressive integrated moving average (ARIMA) exploits the information contained in the timeseries and was used in one example with acceptable results compared to the ground truth (MAPE = 3.88-6.42% depending on direction). Unfortunately, it was not compared to any other method, making it difficult to objectively put this method into context [55]. A direct comparison of historical methods with linear regression (LR) in [61, 62] showed that LR performed with up to 6.7 times lower error than historical methods. However, KF performed up to 3.95 times better than LR. This study is the only example of a direct comparison of KF and LR. Compared to regression models, NNs generally perform with higher accuracy when trained on the same dataset [28]. Historical and regression methods do not cope well with fluctuations [27] and variations of travel times are highly likely at peak times in the urban environment. Therefore, nonlinear methods such as NNs should intuitively perform better

when used with more complex data with higher variation. Pan et al. [59] used a NN to predict the average speed for the remaining distance to the destination, improving the accuracy compared to a historical algorithm by 5.7%. Similarly, in Houston (US), an NN outperformed historical and regression models [43]. Interestingly, this study also found that the improvement, although drastic compared to the historical algorithm, was less pronounced in the suburban areas presumably due to congestion. This also materialises from the findings by [44], that overall NNs performed significantly better. An exception was heavy congestion where historical approaches were more accurate than NNs. Further investigations found that the NN overestimated speeds in slow conditions and underestimated travel times at high speeds. Surprisingly, the information whether a bus was currently in a bus lane did not influence this behaviour. Generally, ETA predictions are made by estimating the absolute number of minutes until arrival or travel speed. In a unique approach, [46] treated the estimation as a classification problem by predicting the arrival in 15 min time slots. In their experiments, an NN-based approach performed 8% better than Decision Trees, Random Forests (RF) and a Naive Bayes approach. An ensemble approach was also used to combine several NNs where parameters such as the number of layers and the number of neurons were randomly assigned and the best performing was included in the final ensemble [33]. Unfortunately, the authors do not report the exact architecture of the final NNs. As the number of layers could have ranged between 1-5, this could be an example of a deep neural network if this information was known.

In the years up to 2018 the relative absence of deep learning approaches is striking. A reason could be the reported behaviour that NNs with a single hidden layer outperformed NNs with two or three layers, suggesting that shallow NNs might be sufficient or even desirable to predict bus ETAs [40]. However, as ETA prediction is a sequential problem, it can be expected that Recurrent Neural Networks (RNN) and their derivatives will perform better. The reason for this is the design specifically tailored to sequential data, where the depth of the network is linked to the length of the sequence [78]. A similar conclusion was reached by [31] who found in a comparison of NN architectures that the more hidden layers a network had, the less likely it was to generalise. On the contrary [27] used a NN with 4 hidden layers that reported excellent performance compared to ordinary least squares regression. As this study does not report on any NNs with different depths, the results are difficult to interpret. Generally, arrival times are predicted for designated bus stops; however, in some public transport systems, buses can be flagged down anywhere on their route. In a study in Bangkok (Thailand), a 4 layer deep neural network was used to improve arrival prediction compared to a regression model, resulting in an error reduction of 55% [27]. The dilemma of choosing a suitable NN architecture has led [47] to use a genetic algorithm (GA) to select the best performing architecture. As it is unlikely that any model will be able to perform with the same accuracy under every condition, some authors have tried to overcome this limitation by using hybrid methods [71]. Such an

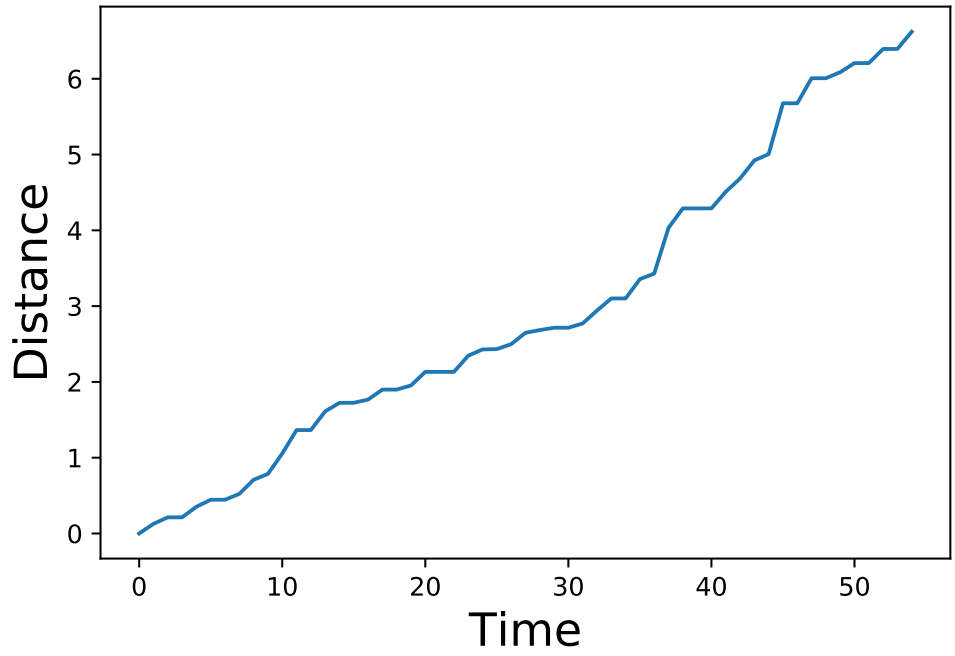


Figure 2.2: Example of a bus trajectory illustrating the travelled distance over time.

example is a combination of an SVM and a KF by [71], where the SVM predicted the baseline values used for the KF prediction. The SVM-KF hybrid achieved 11.1% higher accuracy than a NN-KF hybrid. However, the most commonly used hybrid methods in the context of ETA prediction are NN combinations.

More recent publications have used more complex models to address ETA predictions due to the sequential nature of the problem. LSTMs feature more in the proposed solutions [76]. However, much more complex methods can be found in the recent literature since 2018. These tend to use different types of models for different aspects of the prediction task such as [79] who use a hybrid LSTM based model (LSTM & NN). Another example using different models for each type of input metric [60] (a simple NN for Seasonal features, Exponential smoothing for dwell times and a convolutional LSTM for link run-time prediction). One example employs NNs trained using the artificial bee colony algorithm and reports better performance compared to a simple NN [67].

2.3.2 Trajectory based methods

Trajectory-based methods use historical trajectories of a bus line i.e. the distance traveled by a bus over time (see Figure 2.2 for an example). The estimate is made by comparing the current trajectory of a bus with those of the past and using the most similar trajectory as a prediction. The choice of an appropriate trajectory is made by different algorithms.

One such example is the work described by [37], who select the most similar trajectory using a k-Nearest Neighbour (kNN) algorithm. In this study, the kNN algorithm was found to outperform the NN approach for long-term prediction. Interestingly, this approach did not perform well on short distances below 3 km, and the authors reverted to using the average speed of all buses travelling on the same road segment as a prediction. Similarly, [80] used a kNN classifier to select historical trajectories which were then fed into a KF to predict the bus travel time. In a modification [68] grouped the trajectories into categories according to the road segments and the time of day. The prediction was then made by comparing the progress of the current bus with the historical trajectories corresponding to the time and section of the route the bus is currently travelling on. This approach was used to reduce computational cost and was shown to outperform the SVM and NN trajectory matching approaches. In a comparison of different methods applied to the trajectories, the kernel regression was superior to both the LR and kNN methods [63]. An example also developed a prediction framework that combined features extracted from trajectories with historical data, which were then used in a LSTM to make the final prediction [39].

2.3.3 AVL and headway information

As the progress of a bus is naturally dependent on traffic flow, information about the state of the forward traffic should improve the accuracy of any algorithm. As all the reviewed methods used AVL data, this allows the use of these data from previous buses as an indication of the traffic ahead. The distance or time to the preceding buses is called headway and was used in 7 of the 53 reviewed studies. An example specifically looking at bus stops served by multiple routes showed that the best accuracy could be achieved if not only the weighted headway to preceding buses of the same route but also those to buses of other lines were included. This was true when the prediction was made using an SVM; interestingly, excluding the running time of the same line resulted in the best prediction using a NN, but was still outperformed by the SVM [72].

In contrast, when taking into account the travel times of the previous buses on a virtual road, a NN solution was found to perform better than a SVM [41]. However, [41] used different features as well as 2 hidden layers instead of 1 making a comparison difficult. A further study found that an SVM had slightly better accuracy than NN models and KFs. The error was nearly halved if a KF was used upstream of either model to account for dynamic changes. Also, in this case, the SVM-KF model was slightly superior to the NN-KF approach [29].

The authors of [70] found that overall both NNs and SVMs resulted in a prediction error of around 10% although with minimal variations over the course of the day and in different city environments. A genetic algorithm was used to determine the best architecture for a NN, resulting in an NN with 1 hidden layer and 5 hidden units. This used the same structure as [72] and a similar structures as [29] who used 6 hidden units instead of 5. The described works are very consistent in the selection of network depth, as well as their findings.

An improvement from a simple NN was presented by [51], who used a hierarchical NN. This approach trained sub-NNs for clusters based on the day of the data collection as well as the delay level at the time of collection. These were then combined into a hierarchical NN that performed better than conventional NNs and KFs. Other hierarchical methods are Random Forests which surpassed SVM, kNN and LR. The error was further reduced by 1.3% if the RF was trained on pre-selected datasets using a kNN approach accounting for the intuition that under similar circumstances the travel time will be similar [73]. The methods described in this section use headways as additional inputs to AVL data. However, one method instead used queuing theory. The so-called snapshot method simply uses the travel time of the last bus traversing the same segment as a prediction. To minimise the effect of outliers on this approach, different RF based methods were used to obtain the final prediction based on the snapshot design [38].

2.3.4 AVL and external data

As any road user knows, the progress of traffic depends on many external influences, such as weather or traffic volume. This is also true for buses and has been addressed in a number of studies. The weather conditions have been taken into account in two studies. A basic example that included weather influences used a SVM to make ETA predictions based on data from the last 30 days. These predictions are stored and used as predictions for all journeys the next day. Naturally, this will not account for any sudden changes in external conditions. Unfortunately, this study does not compare the method with any other approaches, making it impossible to objectively evaluate it [49]. Similarly, [45] used an SVM to predict ETAs based on the last four days to predict the fifth. An interesting approach used cameras on overhead bridges to count not only bus traffic but also the speed of taxis as these can use the same routes as buses and unsurprisingly found that their speed is the same in heavy traffic. Furthermore, it was found that the prediction solely based on the information from the static cameras identifying the bus was more accurate than if it were using only GPS recordings. The authors did not combine both to investigate whether this would improve overall performance, although this would have been an insightful addition to their research [17]. Again, these methods were not compared with any alternative approaches. A combination of weather and traffic status was used in a hybrid method. The reasoning is that NNs are often poor at accounting for disruptions; therefore, a system was used, employing a NN for traffic situations that appear to be ‘normal’

in the sense that the system has encountered similar conditions before. If it appears to be an unseen condition, the prediction is made using a KF. This improves performance compared to an NN that is used for all conditions by 0.2 min error for the entire route (37 min) [74]. This highlights the crux that it is unlikely that one method will always perform best and it can be anticipated that different conditions will affect a model's performance.

A preliminary report [40] describes attempts to use LSTMs to predict bus ETAs and includes both traffic and weather data, but the full results have not yet been published. Weather is a more common external feature in the more recent literature. In more recent examples, variations of LSTM predictors are used, such as a bidirectional LSTM [56], a convolutional LSTM [81]. One hybrid method used a genetic approach for hyperparameter tuning of a LSTM model combined with a KF to adjust the model to the latest operational data [57].

2.3.5 AVL and Passenger data

As public transport's purpose is to convey passengers, the customers themselves affect the progress of any bus. The number of passengers that board will influence the dwell time as well as the frequency of stops made by the vehicle.

An interesting sensitivity analysis [31] showed that the impact of dwell time on ETA of a bus has an effect of 45% whereas day of the week played a 25% role. In practice, it is difficult to include the exact number of passengers, as this information often not collected automatically since tickets do not necessarily have information about the destination. However, if these data could be made available, it should give information about future dwell times as more passengers require longer to disembark.

Therefore, passenger numbers boarding and disembarking were included in an NN model that performed significantly better than LR with the same inputs [66]. Due to the difficulty of assessing the number of passengers, an imaginative way used the microphone of a mobile phone installed on the bus to count the sound made when a smart card was swiped at the terminal by a passenger. This information was used to record the number of boarding passengers without any information about the number disembarking [75]. In a comparison, [61] found that a KF performed better if data were used that included the location and the load of passengers. This outperformed a time-lagged NN, as well as LR and a historical model. The same study was republished [62]. This model was later replicated and found to perform with the lowest accuracy compared to NNs and Hierarchical NNs [51]. This illustrates the replication problem found in the current literature, inhibiting any objective comparison of the proposed methods.

2.3.6 AVL and passenger and external data

To account for as many external influences as possible, several studies combined data from external sources such as weather and traffic, and information about passengers.

A NN-KF hybrid where the NN feeds into the KF was developed using features including weather (and more specifically precipitation), passenger loads, boarding and disembarking as well as AVL information. The hybrid performed better than a conventional NN [32]. Generally, two methods of segmentation of a route exist: (1) the stop based segmentation where the travel time between two stops is predicted, and (2) the link based prediction where the travel time of a link consisting of several stop-to-stop segments is estimated. The travel time can either be predicted using a stop-based approach where the time needed from one stop to the next is predicted or a link-based method where the route between two stops is split into several shorter links and each link is predicted separately. In a comparison of stop-based and link-based ETA predictions using AVL data and traffic flow data as features, it was found that the stop-based method performed with up to 2.7 times smaller error [34]. A combination method using an extreme learning machine for non-linear parameters and a SVM for linear parameters also used both passenger load, traffic density, as well as weather data. The final prediction was then made using the weighted sum of both outputs [42].

2.4 Discussion

The feature-based categorisation used in this review allowed a better understanding of the methods applied to predict bus ETAs. The analysis highlighted several flaws in current research that make interpretation of the results challenging. A reliable comparison of the methods was not possible because the measures used to report algorithm performance were inconsistent. Furthermore, some of the reviewed papers presented an algorithm without any comparison to other methods [35, 36, 45, 54, 55, 58], thus preventing any objective assessment. Lastly, the quality of the reporting of some articles was inadequate [33, 66]. These points will be discussed individually.

2.4.1 Reproducibility

As the accuracy and performance of any prediction model is of crucial importance, this has to be reported in a way that allows to replicate and compare the results. However, this is not possible in all cases as some authors report relative errors [45, 82, 51, 54] and no consistency in the parameters reported can be distinguished. The precondition that any machine learning algorithm developed must satisfy is verifiability and has been highlighted in a Royal Society report as one of central importance [83]. This has also been recognised in the healthcare sector where guidelines exist for the development and reporting of predictive models [84]. The

difference in standards could be explained because ETA predictions do not affect the health or safety of a passenger, and a spurious algorithm could most likely cause inconvenience rather than physical harm. However, for an operating company, this might cause a loss of revenue because patronage might decline. Furthermore, the society as a whole could be subject to more congestion, which could be simply reduced by providing accurate ETA predictions. Furthermore, the doctrine of science is replicability. The reproducibility crisis is most prominently known from psychological research [85] however, due to its notoriety, it is actively being addressed [86]. It has also been identified as a problem in 'harder' sciences such as biomedicine [87] and also artificial intelligence [88]. Although results gained from machine learning techniques might be considered to be hard evidence, because the final model is based on mathematical concepts, they suffer from similar problems as seen in psychology where the research is often subjective to the researcher. The similarities between the two fields are that the findings cannot usually be explained due to the 'black-box' effect. The field of psychology has now begun to apply lessons from the problems seen in machine learning research [86]. A suggested way of addressing such problems is meta-science that could shed light on the true accuracy of the findings [89]. However, this is based on comparable precision measurements, which was not found in a large proportion of the reviewed literature. Therefore, comprehensive reporting standards are urgently needed in the field of predictive bus transportation research. As this review revisited the same topic in the space of three years an improvement was noticed in the more recent literature, which all have reported several metrics allowing a better comparison between studies. However, due to the lack of publicly available datasets reproducibility is still not reached.

2.4.2 Comparison

Leading on from the reproducibility problems is the lack of comparison with other methods found in a large body of research. This would not be a major issue if the same prediction measurements were described, however, as this is not the case, such reports only allow limited comparison between the studies. The findings cannot be compared to other researcher's work and therefore can only be considered standalone reports of a method applied to a certain problem. Such studies do not even provide information about any possible relative improvements to other methods currently employed. If the researchers had directly compared their approach to a preexisting or commonly used algorithm, the value of the findings would increase. Comparison to other methods is the only way to establish a benchmark to which any improvement can be compared. Again, a maturation of the recent literature can be observed in which all articles published after 2018 compare several methods and even bus lines in different cities [30].

2.4.3 Quality

The third issue is related to reporting standards and in one extreme example a study did not make it clear what architectures were used in the final algorithm [33] leaving leeway in the interpretation of its findings, by not explaining graphs or figures or because of discrepancies between values in the description compared to the presented figures.

2.4.4 Future challenges

This up-to-date review of the literature highlighted some shortfalls of current research in the field of bus ETAs in public transport networks. On the basis of the presented overview of current practice, several challenges for the field can be identified. Firstly, a framework needs to be developed that allows a comparatively development and testing of ETA prediction algorithms, which will be addressed in our future work. Furthermore, the field would benefit from making any algorithms available for the community to cross-validate, which would most likely speed up the development of this research. In addition, a standard dataset that can be used as a benchmark like in other fields such as image recognition [90, 91] is needed. Alongside these general challenges, some technical challenges can be identified. Mainly the move from predictions based on one bus route to systems incorporating network-wide information. An example impressively demonstrating the capabilities of such an approach for traffic speed prediction based on data from personal cars has been published by [92]. A similar network-wide approach could drastically improve current ETA predictions for public transport systems.

2.4.5 Conclusion

This review highlighted some shortcomings in the current literature on ETA prediction of buses. Overall NNs predominated the methods. In addition, deep learning approaches with more than 2 hidden layers were underrepresented in the publications prior to 2018. However, in one approach an iterative selection of layer numbers and units was applied, but the final layer number was not reported [33].

It was telling that several studies found different algorithms that performed better in different settings, suggesting that there will not be one superior algorithm for all cases. Unfortunately, due to the highlighted shortcomings, it is not possible to identify the ‘best’ method for each of the categories. Considering the popularity of NNs it appears to be the most widely used method, suggesting that it is the best performing and/or most universal method.

Interestingly, deep learning approaches have lagged behind in this research area, but are since 2018 the most common method applied to ETA prediction problems. This suggests that other methods have reached their limits and also that some researchers have access to more and better data required for such approaches. In general, the input features used consisted of

data from one bus line and several variables directly linked to this line, such as other vehicles travelling on the same route. It would be expected that deep learning approaches will be more successful in generalising more complex datasets, for example, if the entire network state is considered, including information about all vehicles in the network.

In conclusion, it can be said that research on bus ETAs lacks consistency and uniform standards. Ideally, an approach similar to image classification or other research areas could be used where a standard reference dataset is made available and used as a benchmark performance test. Alternatively, if the used data were published alongside the used code, this would help increase the comparability. Furthermore, it became clear that an industry-wide standard for reporting prediction accuracy is urgently needed.

2.4.6 Supplementary 1

Publication	Type	Architecture	Evaluation	Accuracy	Features	Comparison
Amita et al. (2015) [28]	NN	NN 3-5-1 and 3-15-1, output = travel time	RMSE, MAPE, R2	NN: MAPE = 6.527%, LR: MAPE = 16.234%	dwell time, delays, distance between stops	NN <LR
Bai et al. (2015) [29]	SVM / NN adjusted with KF	NN-KF: 8-6-1, output= travel time	MAE, MAPE, RMSE	MAPE for best road segment: NN and SVM: ~10%, NN-KF and SVM-KF: 4%, KF: 10.68	Time, Road segment, weighted average of travel time of other buses of other lines, travel time of preceding bus.	NN-KF / SVM-KF <NN / SVM <KF
Celan et al. (2018) [30]	historic		MAE, Mean relative error (MRE)	MRE 5-18%	AVL, georeference, route sections	<Two locations were comapred
Chen et al. (2004a) [31]	NN	Compared activation functions: hyperbolic tangent, tanh linear, sigmoid, sigmoid linear, sigmoid sigmoid: NN1; hidden layers= 1-2, number of neurons are not reported	Average error	prediction varies within 15% of travel time.	Cumulative dwell time, day of week, trip pattern.	<comparison of different activation function combinations

Publication	Type	Architecture	Evaluation	Accuracy	Features	Comparison
Chen et al. (2004b) [32]	NN	NN; 18-[4-6 neurons]-1 , output travel time between two points. NN is dynamically adjusted by KF incorporating the latest current arrival times for unique schedule patterns .	MSE, RMSE	MSE for best pattern; NN-KF: 0.009, NN: 0.016.	precipitation, time of day door opening/closing, stop sequence Ids, trip status, Coordinates, dwell time, stop distance number of passengers boarding disembarking, travel time between two consecutive stops, arrive passenger load and leave passenger load	NN with dynamic adjustment <NN
Chen (2018) [33]	NN potentially DNN	Generated 10 NNs with random number of hidden layers between 1-5, and random number of neurons up to 7. Trained separate NN for urban and rural traffic and used 9 NN ensemble.	Average accuracy	Average Accuracy: NN ensemble (architecture not reported): 94.75%, NN: 94.65%, LR: 94.42%, Statistical mean: 94.08% or LR.	Historical stop to stop travel time.	NN ensemble <NN <LR <Statistical mean

Publication	Type	Architecture	Evaluation	Accuracy	Features	Comparison
Chien et al (2002) [34]	NN	NN link based: 4-6-1, NN stop based: 6-7-1	RMSE, sum of squares errors (SSE)	SE for; link based NN: 0.0965, stop based: 0.041	Link based NN: distance on link, traffic volume, link speed, link delay, link queue, passenger demand. Stop based NN: distance between stops, mean traffic volume, std of volume, mean link speed, std of link speed, mean link delay, std link delay intersections, demand.	NN stop and link <NN stop base <NN link based
Dailey et al. (2001) [35]	KF		Kologormonov - smirnov		Locations	No comparison
Deng & He (2013) [36]	Bayesian network	Used traffic state (average speed for a section) as parent and arrival as child node.	MAPE, MAE, RMSE	MAPE=0.195, MAE=39.09, RMSE=49.14	Road state	No comparison
Dong et al. (2013) [37]	kNN	Average speed of all buses that passed a point over the last 10 mins for predictions below 3km. NN: 18-37-15 where the output applies to all bus stops for distances >3km.	APE (Long distance), AE (short distance)	Better performance on long distances of kNN than NN no values reported. KNN: APE <12 % mean 7%.	Trajectories	kNN <NN

Publication	Type	Architecture	Evaluation	Accuracy	Features	Comparison
Gal et al. (2017 [38])	Snapshot method	The Snapshot method uses the travel time of previous bus on the same route as prediction in heavy traffic. Random Forrest, Extreme random Forrest, AdaBoost, Gradient Tree Optimisation, and combinations with snapshot methods were tested. Optimised versions use the absolute deviation error instead of mean quadratic error.	RMSE, MARE, MdARE	Snapshot method improved the accuracy MARE(%): snapshot-optimised gradient-boost =19.06, optimised gradient-Boost =19.38 snapshot gradient-boost =19.95, gradient-boost =20.46, extreme RF 22.05, snapshot =23.37, RF =24.11, snapshot-adaBoost =26.38, adaBoost =27.08	Travel time of last bus for the same segment. Headway to last bus. Day of the week. Time of day.	snapshot-optimised gradient-boost<optimised gradient-boost <Snapshot gradient-boost <gradient boost <extreme RF <snapshot <RF <snapshot-adaBoost <adaBoost
He et al. (2019) [39]	LSTM	this is a frame work for implemenation	MAE, MAPE min/km	MAPE 5.098	Locations	comparison to historic average, KNN, Tensorflow timeseries, LR, SVR, NN

Publication	Type	Architecture	Evaluation	Accuracy	Features	Comparison
Heghedus (2017) [40]	LSTM	NN: 1-3 hidden layers with 10 & 20 neurons in all combinations only best reported. CNN: 2 convolutional, 2 pooling, 1 fully connected, 1 dropout. Filter 1X1. LSTM: 10 LSTM cells and two activation functions tanh and sigmoid.	MSE, Brier score	Values not shown.	Arrival time, departure time, distance between stops.	LSTM <CNN <NNs of different depth
Hua et al. (2017) [41]	NN	SVM with RBF kernel, NN: 2 hidden layers with 10 and 5 hidden units.	MAE, RMSE, MAPE	RMSE smallest for NN, LR 10% worse than SVM and NN.	preceding travel time of other routes, weighted average travel time of preceding buses, total travel time including on real and virtual road. Including all features above was best.	NN <SVM <LR
Jalney et al. (2021) [42]	hybrid	Extreme learning machine for non-linear features and SVM for linear features combined by weighted sum	MAE, MAPE	MAPE 6.5 and 8.9 depending on route	distance, weather, waiting time, passenger numbers, traffic density, speed, road type, rush hour, red signal duration	two lines were compared

Publication	Type	Architecture	Evaluation	Accuracy	Features	Comparison
Jeong & Rilett (2004) [43]	NN	NN: 1 Hidden layer, hidden units variable depending on route up to 15, out=ETA	MAPE	Average improvement of models 54.2% downtown and 48.61% in north area (Houston) in Comparison to LR. And 71.01% downtown and 76.53% north compared to LR.	dwell time, schedule adherence, distance	NN <LR
Julio et al (2016) [44]	NN	NN: 3-6-5-1, Mixed Model: SVM to cluster the speed categories, second SVM for low speed or NN for higher speeds were used to predict the travel speed.	RMSE, MSE, MAPE	MAPE improvement: NN 7.9-44.7% compared to algorithm which uses current speed as the speed for the next 15-30 min. Bayesian Networks performed so poorly that results were excluded.	3 real time 10 min previous cell speeds, Binary if the to be predicted cell is in a corridor and 4 historical speeds for cells	NN <mixed model <SVR historical <current speed for next section (with exceptions)
Junyou et al (2018), [45]	SVM	Based on four days and predict the 5^{th}. Using RBF.	Relative error.	+/- 0.5 relative error	Data was collected for 4 days to predict travel time of 5^{th} day: traffic flow, average speed, flow density and lane occupacy.	No comparison

Publication	Type	Architecture	Evaluation	Accuracy	Features	Comparison
Kee et al (2017), [46]	Ensemble NN	NN: 24-50-1, ensemble of 10 NN, output: Binary quarter of an hour	Hamming loss, Accuracy, Precision, Recall, F1	Hamming loss: NN <23% . Ensemble up to 8% better than other methods.	Historical arrival times, peak hour, public holiday, weekday, deployment frequency, arrival times from previous hours	NN ensemble <NN <DT <RF <Naïve Bayes
Khosharavi et al (2011) [47]	NN	GA used to select numbers of neurons for each of the 2 hidden NN layers. Number of neurons between 1-10. 500 NNs were trained.	PICP, MPIL, NMPIL, CLC, R2	NN R^2: urban 25.42-46.29, freeway: 83.73	Weekday travel times, and time of day in one direction of the route for 1800 trips over 6 months.	different architectures of NNs
Kumar et al. (2017) [48]	KNN-Kalman	Used KNN to identify similar trajectories and KF to predict the ETA based on the identified trajectory.	MAPE, MAE	MAPE; KNN-KF: 11.6-26.6%, average speed model 13.49-47.58% to 11.6-26.6%.	Trajectories	KNN <average speed
Li et al. (2018) [49]	SVM	Based on the last 30 days to reduce computational cost. SVR with radial bias function.	Absolute error, relative error	Average error 30s.	Time period, Weather, Holiday, Position	No comparison

Publication	Type	Architecture	Evaluation	Accuracy	Features	Comparison
Lin & Zeng (1999) [50]	Historical	Adds average time for the next section to the time at current bus stop.	standard least square method, maximum deviation, fluctuations	Overall deviation 2.0	Locations	<comparison to different version of algorithm
Lin et al (2013) [51]	NN	NN: 11-16-1 for 4 different conditions combined to hierarchical NN.	Relative average error, Relative variance error, Relative prediction error	Hierarchical NN error of 0.2min, other methods 1 min.	Arrival time at stop, departure time from stop, travel time between stop, headway of previous buses, time index, index of delay.	Hiereacrhical NN (better for short distances) = NN (better for long distances) <Kalman (Shalaby & Farhan 2004)
Liu et al. (2021) [93]	hybrid	LSTM and NN combination	MAE, RMSE, MAPE	MAPE: short-term 0.27, long-term 0.04	AVL, station sequence , time, position, speed coordinates	

Publication	Type	Architecture	Evaluation	Accuracy	Features	Comparison
Maiti et al (2014) [53]	NN	NN: 4-7-1, output:arrival at next stop	Percentage error, RMSE	Only graphs no exact values. NN has the lowest percentage error followed by historical model and SVM.	Bus arrival time in previous bus stop, location (latitude and longitude) of previous and target bus stops.	NN <Historical <SVM
Meng et al. (2017) [54]	Historical	Uses current average speed and Historical values for the same section updates the historical data.	minutes	shows actual predictions	Real time location, average speed in section.	No comparison
Nadeeshan et al. (2021) [56]	bidirectional LSTM	predicts intercity travel	RMSE, MAPE, MAE	MAE 24.2	AVL data, weather, wait times, temperature, wind, cloud	<
Napiah & Kamaruddin (2009) [55]	ARIMA		Mean average relative error MARE, MAPPE	MAPPE 3.88-6.42 %	arrival time and departure time, location of stop points, name of location , road network map, timetable information	No comparison to other methods
Nimpanomprasert et al. (2022) [57]	Hybrid	Fetaure selection using genetic algorithm, LSTM KF combination	RMSE	depending on scenario%.	AVL datam precipitation, historic trips	
Padmanaban et al. (2009) [58]	Historical	Model based on 3 days	MAPE	MAPE= 16%.	Travel time, Dwell time, running time for subsections	No comparison

Publication	Type	Architecture	Evaluation	Accuracy	Features	Comparison
Pan et al. (2012) [59]	NN	NN: 1 Hidden layer, input=10 nodes, Hidden layer= 13 units, output predicted speed.	Average prediction error	5.7% improvement compared to historical algorithm.	Stop location, distance, speed	NN <Historical data
Petersen et al. (2019) [60]	Ensemble	Seasonal features (NN), Dwell time (exponential smoothing), link travel time (convolutional LSTM)	RMSE	RMSE 4.38-4.53 depending on horizon	time information, AVL data	<
Shalaby & Farhan (2003) [61]	KF	KF with output travel time	RMSE, Mean Relative Error, Maximum relative error	RMSE; KF =0.36 -0.109, TLRNN =0.075 -0.166, Regression =0.76 -0.220, Historical average =0.181 -0.543.	GPS, Passengers boarding / leaving	KF <TLRNN <Regression <Historical
Shalaby & Farhan (2004) [62]	KF	KF with output travel time	RMSE, Mean Relative Error, Maximum relative error	RMSE; KF =0.36 -0.109, TLRNN =0.075 -0.166, Regression =0.76 -0.220, Historical average =0.181 -0.543.	GPS, Passengers boarding / leaving	KF <TLRNN <Regression <Historical

Publication	Type	Architecture	Evaluation	Accuracy	Features	Comparison
Sinn et al. (2012) [63]	Kernel regression	Kernel regression with Gaussian kernel	Absolute error in min	For a time horizon of 50m in absolute error <10%	Trajectories	Kernel regression <KNN <Linear regression <Delay based
Taparia et. al (2021) [64]	LSTM		MAE, MAPE, RMSE	ETA: MAPE 0.8, run time MAPE: 4.8	AVL, route, stop infomration, journey pattern, delay, arrival at stop distance from city	<Lr, XGboost, historical
Treethidtaphat et al. (2017), [27]	DNN	DNN: 11-7-7-7-7-1	MAE, RMSE	MAPE of DNN 55% lower than OLS	Current location, Target location, Distance, Instantaneous speed, GPS point average speed, Hour, Day	DNN <Ordinary least square regression
Vanajakshi et al. (2009) [65]	KF	KF with output travel time	APE	APE: KF 9.3% better than 7d average	Coordinates, speed, time	KF <average
Wang et al (2014) [66]	RBF-NN	NN: 1 hidden exact architecture not shown	MAPE	Compared to multiple linear regression, and NN.	Travel time, dwell time, distance to next stop, passengers getting on/off, delay, speed to next stop. Online system live speeds, variability.	RBFNN online <RBFNN offline <LR
Wu et al. (2020) [67]	Artifical bee colony algorithm		MSE, RMSE, MAE	22.16	AVL data	<NN

Publication	Type	Architecture	Evaluation	Accuracy	Features	Comparison
Xie et al. (2021) [81]	convolutional LSTM		RMSE, MSE, MAE	MAE 17.08	AVL data, weather, stop arrival/departure, holidays, working days, direction, driver ID	<
Xinghao et al (2013) [17]	Exponential smoothing	Exponential smoothing method	MAE, MAPE	Algorithm based on simulated RFID 14% better MAPE than only GPS data.	length of link, departure time of previous stop, arrival time of previous bus stop, number of intersections on link, delay delay at intersection, distance passed when decelerating or accelerating, acceleration time from static to running speed, the deceleration time from running speed to static.	comparison of different input features
Xu & Ying (2017) [68]	Clustering		MAPE, MAE, RMSE	Time dependent graph performed better than NN and SVM. Only graphs shown	Trajectories	trajectory <NN <SVM
Ye et al. (2021) [69]	SVR		MAPE, MAE, RMSE, R2	R2: SVR=0.995, ARIMAX=0.74	AVL data, dwell times, arrival times at stops	SVR, ARIMAX

Publication	Type	Architecture	Evaluation	Accuracy	Features	Comparison
Yin et al. (2017) [70]	SVM/NN	SVM:3-3-1 output=travel; NN: 3-5-1	MAE, RMSE	Both performed similarly. All predictions with MAPE around 10%.	weighted travel time of preceding bus of same number, weighted travel time of preceding bus of different number, average speed of objective bus	Compared different feature numbers
Yu et al. (2010) [71]	SVM-KF hybrid	Use SVM to predict the baseline travel times and the KF to predict the arrival time.	RMSE	SVM-KF hybrid performance better than actual timetable, The SVM-KF by 11.1% better than NN-KF	Latest bus arrival time, with estimated baseline travel times	SVM-KF <NN-KF
Yu et al. (2011) [72]	SVM	NN 4-5-1 SVM radial bias function.	MAE, MAPE, RMSE, R	R: SVM0.9, NN=0.87, kNN=0.85, LR=0.84	Headway to same line and last bus at stop, running time between stops of same line and last preceding bus.	SVM <NN <k-NN <LR
Yu et al. (2017) [73]	RFNN	Random Forrest based on Nearest Neighbour RFNN	MAE, RMSE, MAPE	MAPE for route with better performance; RFNN: 6.9%, RF: 8.24%, SVM: 11.16%, KNN: 17..33%, LR: 16.41%	dwell time, running condition on the current route segment of 3 preceding buses, running condition of next segment of 3 preceding buses, for each segment, speed variance, average speed,	RFNN <SVM <KNN <LR

Publication	Type	Architecture	Evaluation	Accuracy	Features	Comparison
Zaki et al. (2013) [74]	Hybrid NN-KF	NN: 7-10-3-1	MSE	NN 1.2 min MSE on route, KF 1 min on whole route. Has the MSE been confused with RMSE?.	Day, Direction, Stations, Days Category, Weather, Avg. speed, traffic status	NN-KF <NN
Zeng et al. (2019) [68]	LSTM	Hidden=2, Cells=100	MAPE, RMSE	MAPE: peak=0.06, off peak=0.05	AVL data peak times, dwell times, route links	
Zhang et al. (2015) [75]	Historical	Use the historical travel time between stops and the historical dwell time at the stops to predict overall travel time.	APE	No comparison if further away than 5 stops error of 60s when it gets closer it becomes smaller. Maximum error 300s.	Number of passengers boarding	No comparison

Chapter 3

Impact of Data Quality and Target Representation on Predictions for Urban Bus Networks

Abstract

Passengers using urban bus networks often rely on forecasts of Estimated Times of Arrival (ETA) and live vehicle locations to plan their journeys. ETA predictions are unreliable due to the lack of good quality historical data, while 'live' positions in mobile apps suffer from delays in data transmission. This study uses deep neural networks to predict the next position of a bus under various vehicle location data quality regimes. Additionally, we assess the effect of the target representation on predictions by encoding it either as unconstrained geographical coordinates, progress along known trajectory, or ETA at the next two stops. We demonstrate that without data cleaning, predictions give false confidence if mean errors are used, highlighting the importance of a holistic assessment of the results. We show that the target representation affects the prediction accuracy by constraining the prediction space. The literature is vague about quality issues in public transport data. Here, we show that noisy data is a problem and discuss simple but effective approaches to address these issues. Research generally focuses only on a single method of target representation. Therefore, the comparison of several methods is a useful addition to the literature. This provides insight into the value of addressing data quality issues in urban transport data to enable better predictions and improve the experience of passengers. We show that 'rephrasing' the prediction problem by changing the target representation can yield massively improved predictions. Our findings enable researchers using deep learning approaches in public transport to make more informed decisions about essential data cleaning steps and problem representation for improved results.

3.1 Introduction

Bus passengers increasingly rely on Real-Time Passenger Information (RTPI) systems at bus stops, online and in mobile apps. Current RTPI systems attempt to account for deviations from the timetable but are often unreliable [2]. This affects the convenience of bus passengers and is reflected in customer surveys as the area of improvement most frequently requested [1]. In general, passengers assign different importance to certain aspects of public transport. Reliability and safety are considered the most important [94]. This highlights the importance of accurate Estimated Time of Arrival (ETA) predictions to improve customer experience [15] and increase public transport use.

Many cities suffer from severe congestion due to the increase in the number of cars [95], making travelling a challenge. In a recent report, it was estimated that in the UK, travellers spent 10% of their driving time in gridlock, costing the economy £38 billion [11]. The same report ranked Bournemouth as the 8^{th} most congested city in the UK. Prospective studies suggest that the greatest environmental and social impact can be achieved if the public is encouraged to change from private cars to public transport, thus reducing air pollution and congestion [12]. This was illustrated by a study suggesting that cancelling just 1% of daily commutes from specific neighbourhoods in the Boston (US) area can reduce the delays of all road users by up to 18% [13].

To encourage such a shift, it is important to address the passengers' desire for reliability. As delays in bus services are inevitable, it is crucial to keep passengers informed. As many public transport apps give 'live' positions of vehicles, these are often used by passengers to decide when to leave to catch their bus without having to wait too long at a bus stop. However, due to the latency of this information caused by delays in wireless network infrastructure and passing data through a number of 3^{rd} party systems, these data are delayed, suggesting that the vehicle is further away than it is in reality. In the Bournemouth area, for example, the latency of the internet-based 'live position' is approximately 30s and could be the difference between a passenger catching a bus or missing it. Therefore, a reliable short-horizon prediction to tackle this delay would undoubtedly be useful.

The infrastructure required to allow such predictions is already in place in the form of Automatic Vehicle Location (AVL) systems [96]. As AVL systems stream data continuously, in theory they could easily be leveraged to develop better data-driven solutions. However, the AVL data suffers from serious quality issues. These include the lack of clear journey identification linkable to the timetable, artefacts such as gaps in recordings, falsely reported line numbers, and directions. The biggest positive impact for passengers can be achieved by improving not only the delay seen in 'live locations, but also the ETA predictions at bus stops. To this end, this study uses one bus line from the city of Bournemouth (UK) as an example and addresses: (1) the data quality issues encountered, (2) their impact on prediction using Recurrent Neural Networks (RNN) and measures to overcome the identified issues, and (3) the impact of target representation on ETA prediction accuracy where we compare the accuracy of two types of output – the position in the next 40s, which is the equivalent to a next-step prediction based on the sample rate of our dataset and the arrival time at the next two bus stops.

3.2 Related work

Urban bus networks generate highly multidimensional data. This includes not only the geographic and temporal aspects but also the data generated by several vehicles serving the same line with different timetables or directions. This large data source can be easily affected by quality issues. The importance of data quality has been highlighted in the literature in the context of bus travel, for example, for pattern analysis [97] but also to allow general improvement of public transport services [98]. Other authors have proposed methods to tackle these issues [99]. However, it is notable that few literature examples directly address data quality. This problem should be much more prevalent considering the strive to include big data in urban transport predictions from novel data sources, especially via crowdsourcing [100, 101, 102]. The assumption that cleaner data will allow better predictions is examined in this paper.

The second question is how to best represent the prediction problems. Reducing the complexity of the input data can have beneficial effects on prediction tasks [103]. This is generally applied to the input. Furthermore, the representation learning technique suggests that there is a right way to pose a question to a machine learning algorithm [104]. More well-known examples that highlight the importance of target representation come from medical image classification, where algorithms have been found to use confounding clues such as visible cables in an image to make a prediction [105]. Therefore, an empirical approach will be used to compare the quality difference of three target representations of similar prediction problems specific to public transport.

3.3 Methods

3.3.1 Data collection

The data used was collected from one of two bus operators in the city of Bournemouth (UK). Vehicles transmit their position approximately every 40 s which is collected by the company providing the Electronic Ticketing Machines (ETMs) with the integrated AVL-system. Due to the involvement of several companies handling the data, only a limited amount of information is transmitted. The available data are as follows.

- Timestamp
- Position (latitude and longitude)
- Line number
- Direction (outbound or inbound)

It became apparent that neither the direction nor the line numbers are reliable. The transmitted direction is often incorrect, and so are the line numbers when a vehicle changes its line during an operational run. This becomes evident when observing data identified as one line but serving another as well as vehicles travelling in the opposite direction from their transmitted data. This suggests that although the coordinates are updated continuously, the additional information is not always updated after a vehicle starts its journey. On the basis of this limited information, it is typically not possible to match a vehicle to a timetable corresponding to the journey it is currently serving. A journey is a specific trip found in the timetable of a bus line, for example, the outbound 9 AM service 1. In contrast, a route pattern is

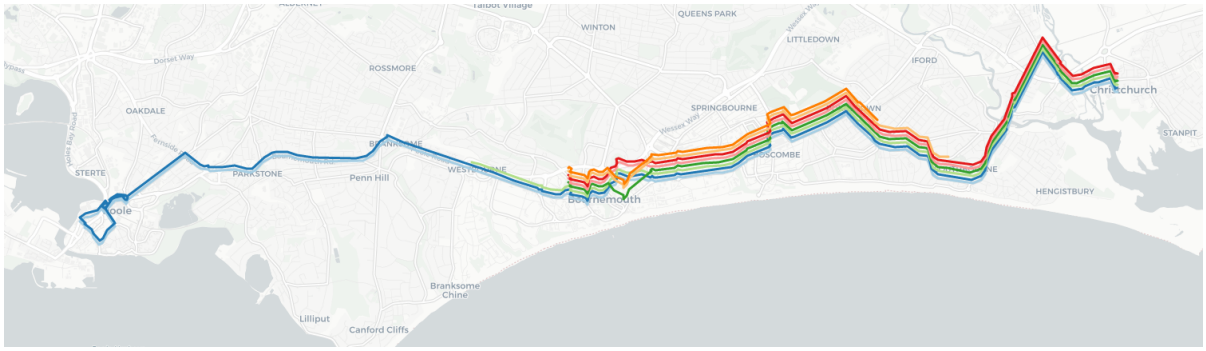


Figure 3.1: Map showing different route patterns associated with line 1 in Bournemouth (UK). Overall this line has 12 more or less distinct patterns (4 inbound and 8 in the outbound direction). For clarity each shape was offset by 0.0005° northwards to prevent overlapping.

the route as travelled on the road which can vary slightly for each journey for the same bus service. In the example of line 1 in Bournemouth, each line has several patterns, which can include different start points along the route, resulting in shorter overall journeys or slightly different routes – see Fig. 3.1 for examples. Matching a vehicle directly to a specific route pattern is not possible, as no specific identifiers are transmitted.

Therefore, a specific route pattern was selected for the proof of concept. This route pattern is line 1 in Bournemouth in the outbound direction from the city centre (Triangle) to the final destination (Christchurch). The reason for this choice is that the start point for all inbound journeys is the same, making these journeys indistinguishable. The inbound journeys, however, do not always have the same destination, so the outbound direction was chosen to allow better identification of journeys.

Identification of individual journeys

As the data lack an explicit indication of the progress of the journey (e.g., bus stops already visited), it is not self-evident when a journey ended and a subsequent journey started. An observation made was that between two timetabled journeys, the vehicle generally goes briefly offline. Thus, once it comes online again, a gap in the recordings can be detected. A new journey was defined as a time gap of more than 15 min. If such a gap is detected, it is assumed that a new journey has started.

3.3.2 Representation of a journey as trajectory

All buses should follow a predefined route which can be represented as a trajectory. Trajectories are the distances a vehicle has travelled along a route over time. This means that the trajectory will always be different for each journey. The transmitted coordinates are simply projected onto a route pattern by assuming that the closest point on the route to the current coordinates represents the position of the vehicle (see Section 3.3.3 for filtering approaches). These trajectories are used as one target representation as well as for benchmarking as described below. As the route is known, the positions along the trajectories can be converted back to the coordinates along the route (Fig. 3.2).

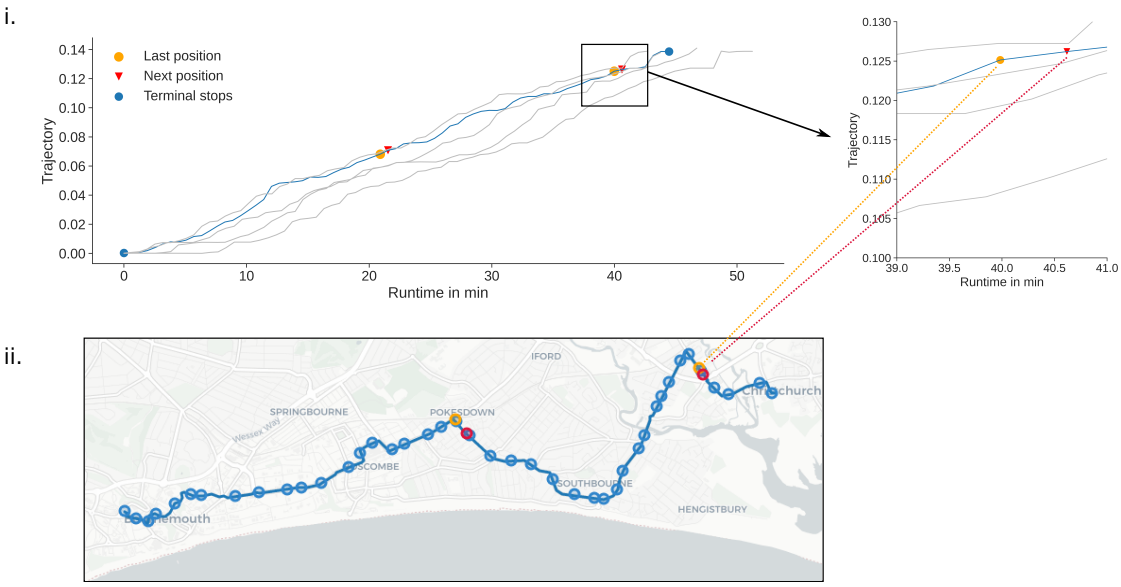


Figure 3.2: i. The trajectory representation of several journeys, where the progress along the route is represented over time. The difference between several vehicles travelling on the same route is illustrated. As an example, one journey has been highlighted in blue with examples of the input position in yellow and the target position in red. **ii.** The route of the bus line with stops indicated as blue circles. The highlighted trajectory positions are shown as coloured circles on the route

3.3.3 Data pre-processing

As described previously, the data suffered from quality issues. Problems encountered were misreported identifiers and positions that resulted in physically impossible position changes between recordings, such as vehicles travelling over 60 mph. To combat these, several filtering procedures were applied within the ablation study to clean the data and assess the influence these cleaning steps have on the final results (Table 3.1). These data cleaning steps are referred to as 'sets' and correspond to the experiments described in Section 3.4:

- *Set 1.0 – minimal processing.* As the line number of a vehicle was unreliable, all vehicles identifying themselves as line 1 were selected. To ensure that these are following the correct route, the journeys were filtered by excluding those with reported positions further than 2 x the mean distance from the route. Furthermore, any vehicle which appeared to travel faster than 62 mph (100 km/h) was also removed as this is legally and physically not possible within a city environment. This represents the dataset with minimal prepossessing and thus has the most data points. To ensure a fair comparison with the more heavily processed datasets, a randomly selected subset of 1476 journeys of this dataset was used.
- *Set 2.0 – filtering of direction.* As the direction was found to be reported incorrectly, the outbound direction was filtered by ensuring that each vehicle was within 100 m of the first outbound stop at the beginning of its journey. If this was not the case, these journeys were removed.

- *Set 2.1 – removing repetitions from the end.* In practice, a vehicle will stop at the beginning and end of the journey for operational reasons. Therefore, these positions will be repeated until the vehicle starts the next journey. These repetitions were removed from the end of the journey once the bus has reached the closest point to the final destination.
- *Set 2.2 – removing repetitions from the start.* Stationary repetitions were removed from the beginning of each journey, assuming that a vehicle had started its journey once it had moved more than 10 m between recordings. This removes positions where the bus has arrived at the beginning of the route but is waiting for the timetabled journey start.
- *Set 3.0 – removing all repetitions.* The final set combines all above-described filters and is, therefore, the most heavily processed dataset.

Set	Outbound only	End truncation	Start truncation
1.0			
2.0	X		
2.1	X	X	
2.2	X		X
3.0	X	X	X

Table 3.1: Ablation study setup.

3.3.4 Benchmarks

The literature on ETA prediction in public transport often lacks comparative benchmarks, making it difficult to objectively compare different approaches. In other areas of machine learning, it has become the norm to use benchmarks and standard datasets. As there is no appropriate publicly available benchmark dataset available for public buses in urban areas, this study uses benchmarks that can be easily implemented on any dataset. This allows other researchers to compare their solutions to this publication, but also gives a threshold to assess any results against. The benchmarks are as follows:

1. *Average speed.* This method uses the average speed of a vehicle since the start of its current journey. Thus, it does not reflect any short-term speed variations. The calculated speed is used to interpolate the position of the vehicle from the trajectory of its journey pattern for the next 40 s.
2. *Current speed.* This method uses the last three transmitted positions of a vehicle to calculate its current speed. The prediction is made by interpolating the position for the next 40 s from the journey trajectory. This method will account for temporary speed variations.
3. *ETA benchmarks.* To calculate the ETA benchmarks both speed-based methods are used to interpolate the arrival time at the next two stops for the ETA based benchmarks. For further details see Section 3.3.5.

3.3.5 Target representation

To investigate differences in accuracy, three different target representations were used. All of these use the same data as input but represent the prediction target differently:

1. *Unconstrained coordinates.* The raw data of bus locations are affected by inaccuracies due to interference of the GPS signal. Therefore, the positions of vehicles are not always directly on the route. This represents the raw target where no preprocessing of the target was applied. The only constraint used was a bounding box framing the city. This approach predicts two normalised values representing coordinates within the bounding box.
2. *Trajectory.* The raw coordinates can be projected onto the route-pattern of a journey by simply using the closest point on the route as position once a journey is successfully matched to a route-pattern. This ensures that inaccuracies locating a vehicle off-route are removed. The route-matched positions can be turned into a trajectory by plotting the distance along the route over time as demonstrated in Fig. 3.2. In practice, this method predicts a number representing the progress along the trajectory with a max of 1, which is the final destination.
3. *ETA.* This approach predicts the arrival time at the next bus stop instead of the position of a vehicle. As the next stop could be very close to the vehicle, we predict the next two stops instead. The prediction itself is in seconds to the corresponding stop. As we aim to compare different target representations, to make the ETA predictions more comparable to the position based approaches, the error in seconds was translated into an approximate margin of error in meters based on the travel speed, assuming the bus travels at a constant speed from its current position to the two stops. The distance-based errors are approximations for comparison only.

3.3.6 Model training and evaluation

All models were trained on an Nvidia GeForce RTX 2060 GPU using the *fastai* library. The experimental setup was the following.

1. *Input features.* The features included were coordinates normalised to a bounding box, the bearing reported by the AVL system, the time-delta between consecutive recordings, the elapsed time from the start of the journey and time embeddings as described below. The input features were min-max normalised unless stated otherwise.
2. *Time embeddings.* The time information was split into its components to make it possible for the algorithms to learn seasonal patterns. To achieve this the timestamp was translated into minute of the day, hour of the day, day of the week, day of the month and month of the year. These were embedded in a multidimensional space as detailed in the architecture description.
3. *Architecture.* Two neural network models were used with identical architecture (Fig. 3.3) except for the Recurrent Neural Network (RNN) module which was either a Gated Recurrent Unit (GRU) or a Long Short Term Memory (LSTM) network. The time embeddings were learned by the network in a multidimensional space. The dimensions were chosen as half of the possible number of values for each embedded variable. As an example, the hour of the day was embedded in 12 dimensions as the maximum number of hours is 24. These embeddings with a total of 52 dimensions were fed into a linear layer to reduce their dimensions back to the original number

of time based features. The output of the linear layer was concatenated with the remaining input features and fed into either a GRU or LSTM layer followed sequentially by a 1D Batchnorm, a linear layer, a leaky ReLU, a second Batchnorm and a final linear layer. To ensure the outputs were bounded, a sigmoid function was also applied.

4. *Hyper-parameters.* To allow for direct comparison between the models all training hyper-parameters were kept constant. It is appreciated that this might not in all cases yield the best performance but will illustrate the influence of the modifications made on the performance. The used variables were chosen through empirical exploration. Each model was trained for 50 epochs using the one-cycle policy [106] with a maximum learning rate of 10^{-3}. Networks with unconstrained coordinate targets used the haversine distance between target and prediction as loss-function, while all other networks were trained using the Mean Average Error (MAE).

3.3.7 Evaluation

Predictions from approaches that produce positional outputs, including coordinate- and trajectory-based predictions, were converted to denormalised coordinates. The errors were evaluated by the haversine distance between the target and the prediction.

As the ETA-based approach does not give any location-based prediction, the error in meters was estimated. This was done using the error in seconds to calculate the number of meters travelled in this time, based on the average speed between the current position and the target stop. This assumes that the vehicle travels at a constant speed and, therefore, is not used as a loss function, but rather as a comparison.

3.4 Results and Discussion

3.4.1 Data cleaning

The dataset spans 144 days (12-Oct-2019 to 04-Mar-2020) with an overall number of 1,909,861 instances (bus location records). These correspond to 4080 individual journeys as it can be seen in Fig. 3.4. This excludes 0.9312% of journeys due to speeds above 62 mph. Filtering by direction, as discussed in Section 3.4.4, leaves 1486 (36.42%) of the overall number of journeys.

3.4.2 Benchmarking

The purpose of the benchmark is to give a baseline to interpret subsequent results. Fig. 3.5 (a) shows the distribution of errors for each benchmark in meters.

The peak at 200 m stands out in the mean-speed benchmark. This error occurs when a vehicle remains stationary as this method does not allow for a stationary prediction. The average distance travelled along the trajectory corresponds to ~200 m (the error is calculated as the straight line distance, therefore corners or loops will cause smaller errors than the same distance along a straight part of

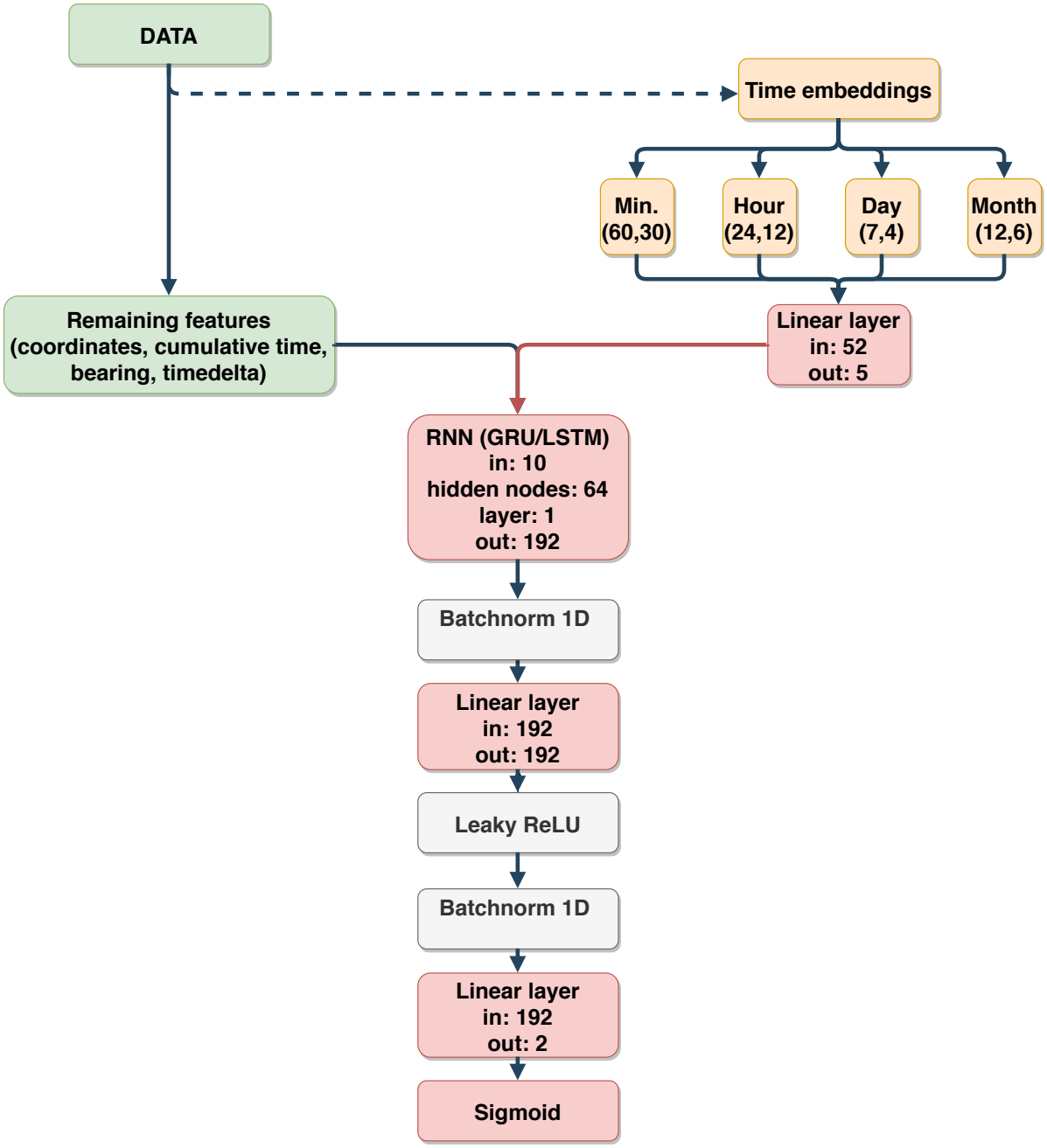


Figure 3.3: Network architecture for the two RNN approaches – GRU or LSTM without any other changes to the network.

the route). Plotting the errors along the route gives a more detailed overview of the performance of the benchmark as shown in Fig. 3.5(b). This confirms the hypothesis that in general, the benchmark will perform poorly at stationary positions, which is especially evident at the start and the end of the journey when some vehicles remain stationary for an extended period of time.

Interestingly, set 1.0, which is the least processed and thus affected most by noise, had the best results. This is further discussed in Section 3.4.4. The mean-speed predictions will be used as a baseline from hereon.

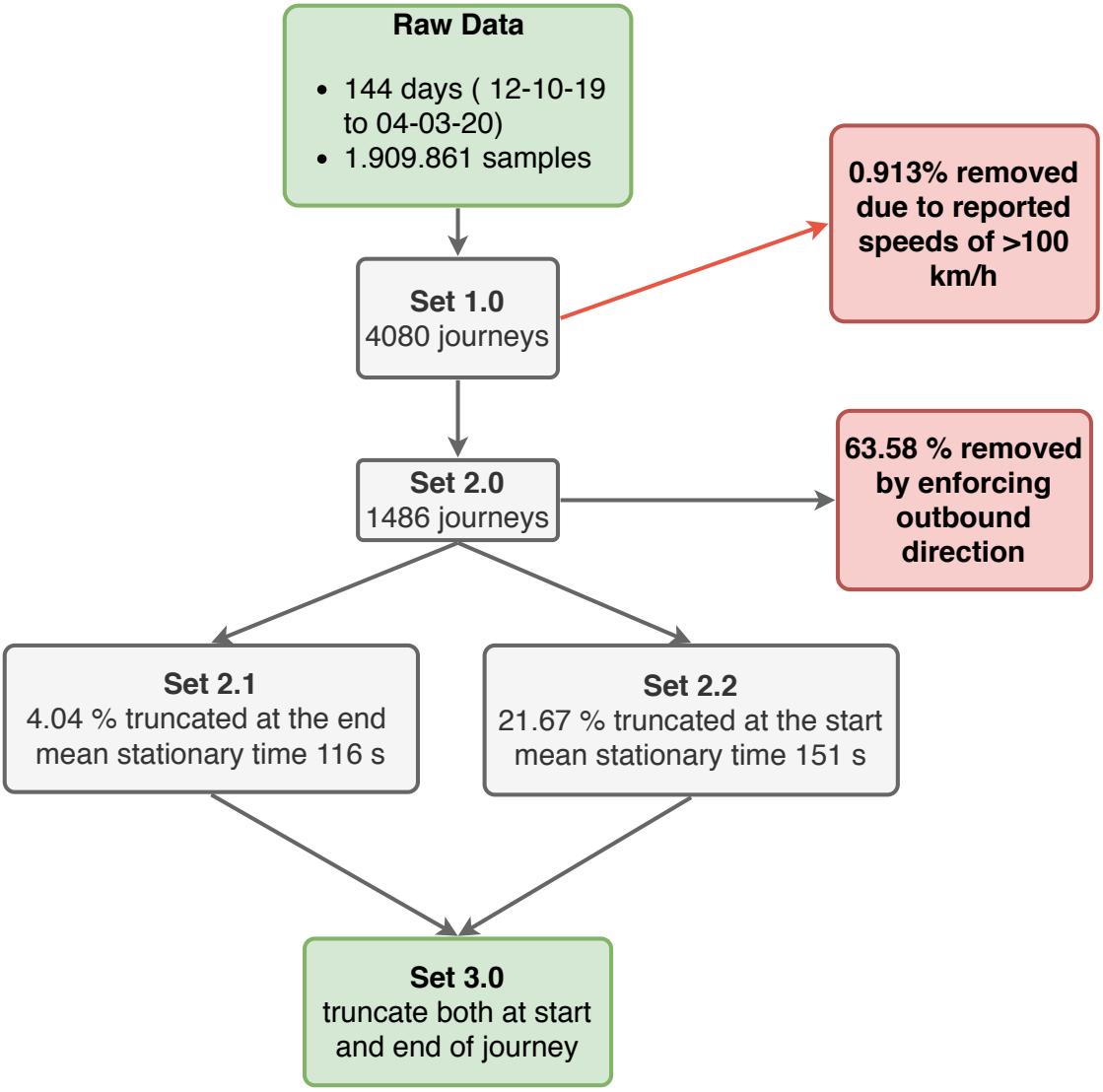
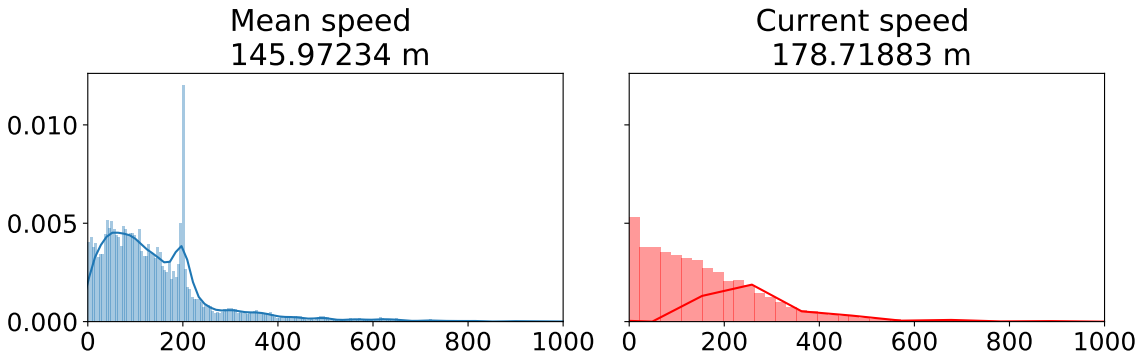


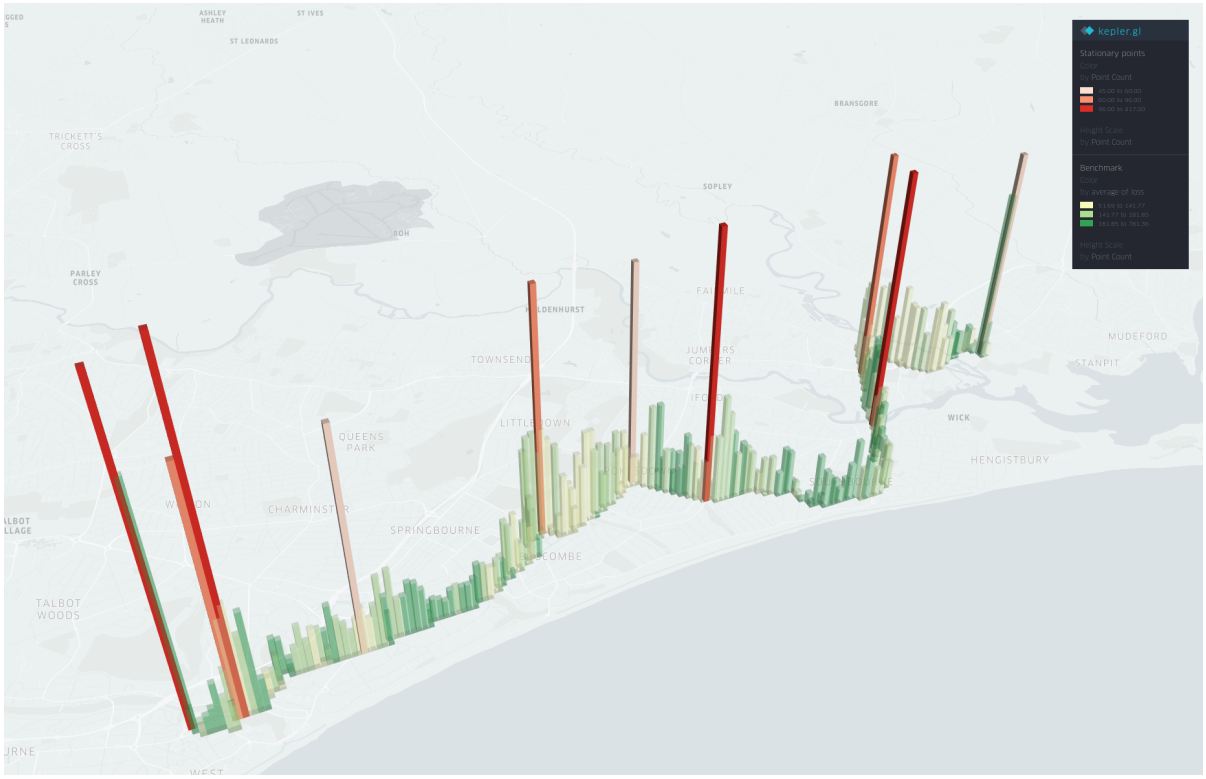
Figure 3.4: Step-wise data cleaning sequence.

3.4.3 Data quality

In addition to the aforementioned problems with the data quality, the collected data contained characteristic circular patterns (Fig. 3.6i). These occur at bus stops only and are not explainable by artefacts from GPS interference. An empirical investigation showed that the origin of this phenomenon is most likely a side effect of geofencing that the AVL-system uses to determine if a vehicle has arrived at the stop. Unless the bus has been very close to the stop, the AVL-system 'snaps' the real position of the vehicle to the geofence boundary (Fig. 3.6 iii). By choosing this exclusion zone to be 10 m in radius, it was possible to simulate data mimicking the artefact seen in the real-life data (Fig. 3.6 ii). The issue requires further investigation to verify the exact rules this artefact is following.



(a) Benchmarks for set 2.0. The 200 m peak visible in the left plot occurs when a vehicle stops either at traffic lights, pedestrian crossings or a bus stop. This peak is not found in the current speed benchmark as it naturally compensates for variations in speed. Overall the global average does result in lower haversine mean-error.



(b) Performance evaluation of the mean-speed benchmark on set 2.0. The average error is shown in meters as green bars (colour and height indicate the average error along the route). The number of repeated positions are shown in red. The bars show the points at which more than 90% of repeated positions occur, generally at bus stops.

Figure 3.5: Assessment of the benchmarks.

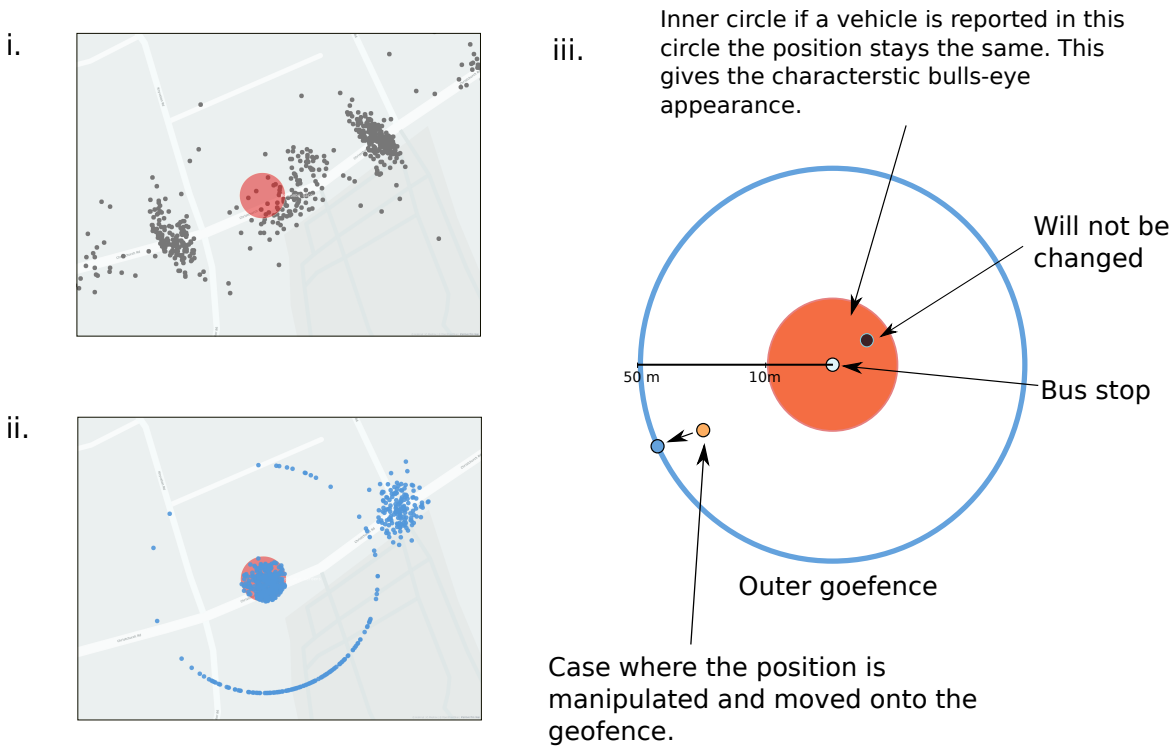


Figure 3.6: i. The circular artefact recorded from real-life data. The red circle denotes the bus stop. **ii.** The simulated data generated closely resembles the artefact recorded. **iii.** The underlying process used to simulate the data.

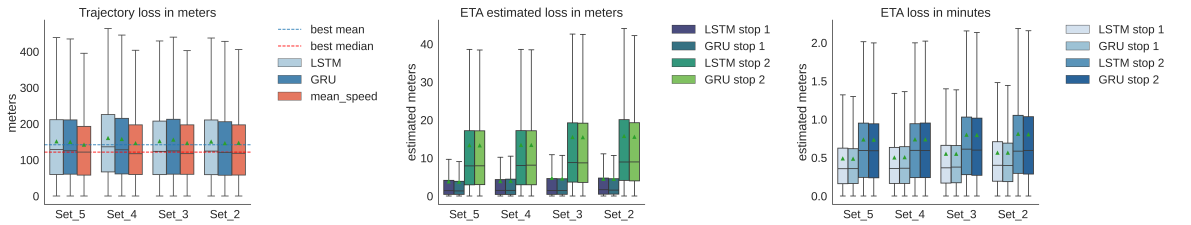


Figure 3.7: Left: Boxplot showing error in meters for GRU, LSTM and the mean-speed benchmark. Outliers have been removed. Green triangles represent the mean and the median is represented as a horizontal black line. The best benchmark's (based on Sharpe ratio set 1.0) median and mean are shown as red and blue dashed lines respectively. **Middle:** Boxplot showing the estimated error in meters for the ETA prediction. Both networks are shown and errors are given for the first and second stop. Boxplots showing the errors in minutes for the ETA prediction for either network in comparison to the benchmark. The prediction is more accurate for the immediately next stop and the error increases for the second stops. Note the difference in error magnitude. **Right:** Boxplot showing the ETA loss in minutes.

3.4.4 Effects of data cleaning

To evaluate the effect of the cleaning steps on prediction quality, two target representations were tested: the prediction of unconstrained coordinates and the trajectory-based prediction. As the mean error of the trajectory was ~100 m lower when compared to unconstrained coordinates, only the trajectory-based predictions are discussed for clarity.

Errors for both model types are shown in Fig. 3.7. The performance for all model types is similar in general. Interestingly, the benchmark is very robust and only in set 2.2, the GRU has a slight advantage over the MAE of the benchmark (GRU: 145.86 m, mean-speed: 146.85 m) but the difference is negligible. In practice, the mean error is not the best metric for model evaluation, as it does not account for the spread of the errors. To better assess the performance, the Sharpe ratio [107] was used (Eq. 3.1). Widely used in finance, it accounts for the standard deviation of the errors or their volatility. This gives a different picture and the benchmark is outperformed except in the case of set 2.1 (see Fig. 3.9 and Table 3.2).

$$S = \frac{MAE - r}{\sigma} \tag{3.1}$$

Sharpe ratio (S), where r is the risk-free rate which here translates to the best expected error (assumed to be perfect at 0m) and σ is standard deviation of errors.

Comparing the error distributions of the network with the lowest MAE (GRU using set 2.2) to the benchmark (Fig. 3.8) it becomes apparent that the GRU's distribution is skewed toward smaller errors and is not bi-modal like the benchmark. This confirms that in practice the GRU will deliver a more reliable prediction even though the mean error is similar.

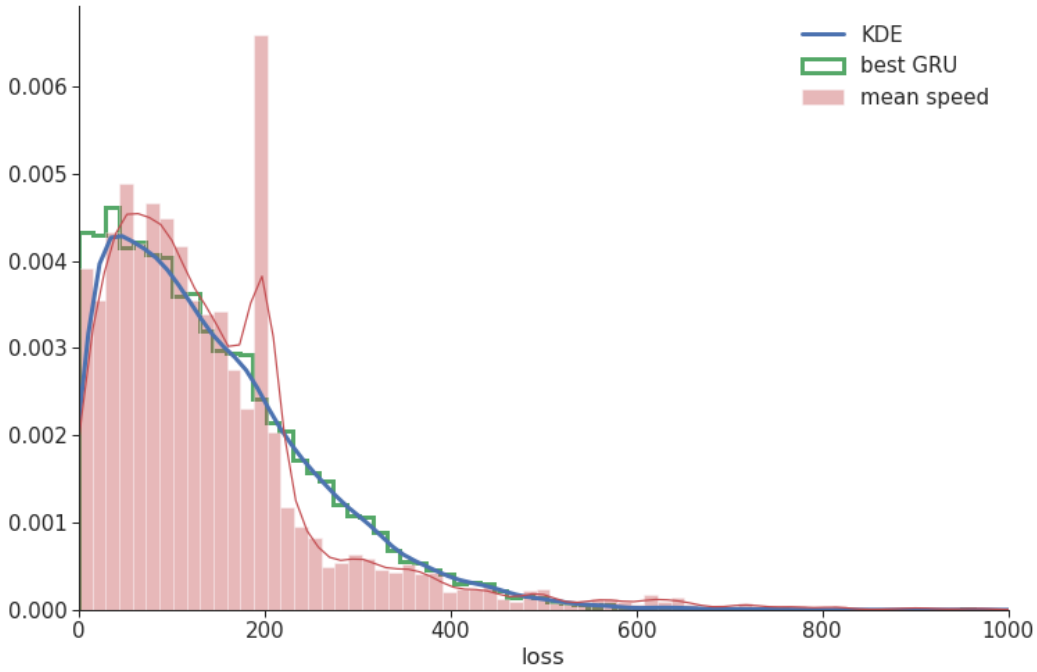


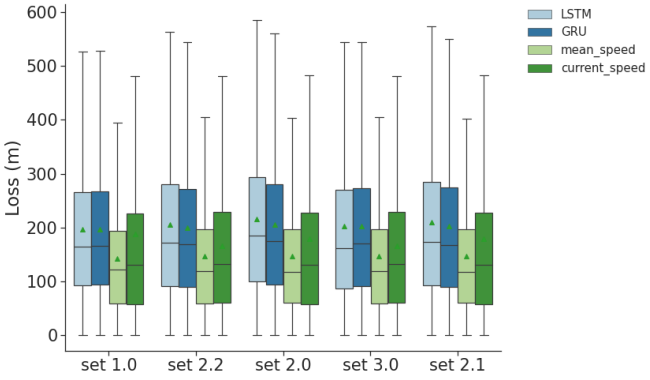
Figure 3.8: The error distribution for the best performing GRU and the mean-speed benchmark. The 200 m peak caused by stationary vehicles is apparent. The outline of the GRU errors suggests that the model makes more reliable predictions.

	set	Sharpe ratio	Delta benchmark	Delta set 1.0
GRU	set 1.0	1.26	9.18	0.00
	set 2.0	1.08	-4.29	-14.62
	set 2.1	1.03	-8.69	-18.41
	set 2.2	1.25	12.78	0.89
	set 3.0	1.13	1.49	-10.66
LSTM	set 1.0	1.27	9.84	0.00
	set 2.0	1.27	12.98	0.19
	set 2.1	1.11	-1.77	-12.76
	set 2.2	1.21	9.33	4.71
	set 3.0	1.21	8.80	-4.80

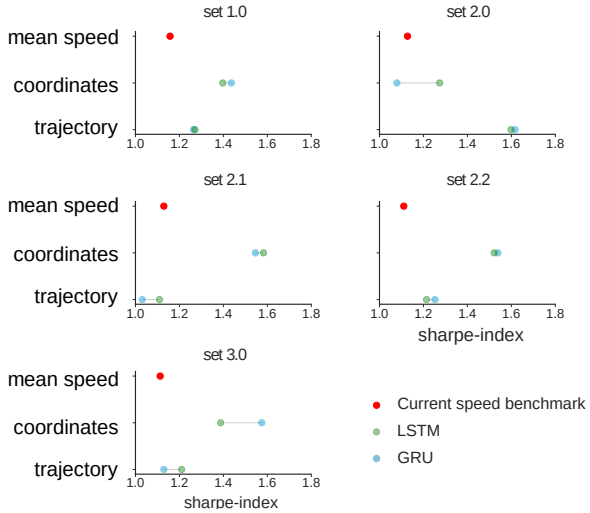
Table 3.2: Table showing the Sharpe ratios for the two different model architectures as well as the different sets. The **delta benchmark** shows the difference of the respective model compared to the benchmark of the respective set in %. The **delta set** describes the inter model change of Sharpe ratio compared to the model's Sharpe ratio in set 1.0 in %.

Set 2.0 filter direction

The evaluation of the cleaning steps shows an interesting behaviour. The first cleaned dataset 2.0 shows an increase in the mean error of both RNNs and the mean speed benchmark Fig. 3.7. It would be expected that limiting the data to a single direction should improve predictability. When assessing the Sharpe ratio, the findings are different and the LSTM shows an improvement of 0.189% whereas the other methods decrease in performance with large errors (Fig. 3.9).



(a) The error of the networks and benchmark in meters. Using this evaluation metric the benchmark cannot be outperformed.



(b) Cleveland plot showing the Sharpe ratio (higher number = better performance). Using this more holistic metric compared to MAE the benchmark is generally outperformed.

Figure 3.9: Comparison of different ranking approaches.

Set 2.1 removal of end repetitions

At the end of a journey, the vehicle will in some cases repeat transmission from the same position. This occurred in 4.04% of the journeys with an average stationary time of 116 s. As the transmission frequency is 40 s this represents ~3 repeated positions. This only affects a small portion of the journeys but still worsened the performance compared to the raw dataset (set 1.0) both when assessed using MAE and causing a reduction of -12.755% of the Sharpe ratio. This suggests that both LSTM and GRU had an advantage in those few stationary cases. When assessing the error map of the LSTM for this dataset (Fig. 3.5) it becomes apparent that the error of the set 2.0 at the final data-points is ~90 m, whereas upon removing repetitions the error in the same area ranges from 283-818 m. This suggests that the LSTM became exceptionally good at predicting the final position if the vehicle remained stationary, thus appearing to improve the overall performance.

Set 2.2 removal of start repetitions

A large proportion of vehicles idle at the start of a journey, which affects 21.67% of the journeys with an average idle time of 151 s corresponding to ~4 repeated positions. This means a much larger proportion of the vehicles will arrive early at the start of their journey compared to those that remain stationary once the journey is finished. Also this cleaning step reduced the Sharpe ratio by -4.499% for the LSTM when compared to set 2.0. Interestingly, the GRU performed best in this scenario, whereas in all other cleaning steps the LSTM outperformed the GRU. In addition, under this scenario, the best overall MAE was achieved by the GRU of 145.86 m. This suggests that the GRU suffered most from repeated starting points.

Set 3.0 removal of repetitions at the start and end

The final dataset 3.0 is the most processed and reflects the journey most closely with the lowest influence from artefacts. Interestingly, the models trained on this dataset did not achieve the overall best performance. The LSTM lies -4.8% below the Sharpe ratio of set 1.0. This might appear counter-intuitive. However, on closer inspection, this behaviour can be explained by the fact that the RNNs are most accurate when a vehicle is stationary. This causes the reduction of the Sharpe ratio due to over 20% of the journeys having repetitions at the start or the end, making these cases easy to predict. This causes the paradox that the cleanest dataset appears to perform worse, yet the reason for this is that the model is no longer able to predict the artefacts of repeated positions at either end of the journey.

Interpretation of data cleaning results

It is important to look at error locations and where exceptional performance is achieved. It is clearly easier to predict the position of a bus when it finished a journey because it will not move again, whereas at the beginning the difficulty is to predict when the vehicle will start moving. This could be overcome if the timetable corresponding to a journey was known. It can hence be concluded that although the overall metrics have not improved after data cleaning, it is still beneficial to prevent falsely reduced metrics due to accurate prediction of artefacts.

As mentioned before, the mean-speed benchmark fails when a vehicle comes to a stop as it assumes that all vehicles travel at a constant speed. Therefore, the benefit of RNNs is that it will learn locations where this is likely to happen, such as bus stops, junctions, or pedestrian crossings. This is indeed seen in the data, in areas with the most stationary data points, the accuracy of the best RNN is lower than the average accuracy, meaning that the neural network has learned the areas where a vehicle is most likely to stop for some time (Fig. 3.10).

3.4.5 The influence of target representation

The prediction of unconstrained coordinates is not an effective approach and results in large errors and therefore is not shown here. This is not a surprising finding as this approach has a very large prediction space spanning the entire city. In reality, the possible predictions are along the route, thus making this an unnecessarily difficult approach.

To combat this complexity issue, the trajectory approach was used with much better results. However, this requires linking the data to a specific route pattern, which in practice can be challenging. This again is an expected finding as limiting the prediction space to the route which is the only possible space the vehicle should be travelling on simplifies the problem dramatically, reducing the errors approximately by half from 244.8 m (LSTM) to 141.3 m (LSTM).

The third approach, predicting the ETA at the two subsequent bus stops might be the most important method. To make this approach more comparable, the error given in seconds was translated into rough estimates of a distance error.

All models trained to predict ETAs used the last 3 positions to interpolate ETA and performed better than the benchmark (Fig. 3.7). The benchmarking method using the current speed was chosen as it performed substantially better than the overall-speed based method (current speed MAE: 1.2 min, mean speed MAE: 10.7 min). As would be expected, the first stop is predicted with higher accuracy compared to the second stop. Both the GRU and LSTM perform better than the benchmark when comparing the MAE. Both the mean and Sharpe ratio are reduced by the data cleaning steps.

In all sets, the performance of the ETA prediction method is ~10 fold more accurate with estimated mean distance errors of ~4.2 m for the first stop and 14.5 m for the second stop for both RNNs. This is a substantial improvement compared to the position-based method (the best scenario has an error of 145.8 m) and could be further improved by making additional information such as the distance to the next stop available to the neural network. Furthermore, the network could easily be changed to predict the arrival times for all the following stops. The drawback of this method is that for the data used the actual ETA is not known. Therefore, the ETA is an approximation and might not fully represent reality. However, to collect accurate ETA information, the current technology would need to be upgraded, e.g., with proximity sensors at each bus stop as the sample interval is insufficient.

In light of the findings representing short-term prediction targets as ETA problem rather than a position-based target gives by far the best results with the caveat that the ground truth used to compare the ETA against is an estimation.

Figure 3.10: The number of repeated positions generally seen at main bus stops, junctions and crossings shown in red. The error of the best GRU trained on set 2.2 in turquoise. The error is generally low if a vehicle is more likely to stop in an area.

3.5 Conclusions

Bus travel is a well-established mode of public transport and the vehicles are mostly equipped with modern telemetry systems. However, we highlighted data quality issues, which complicate any data-driven solutions. Unreliable or omitted information about the route and timetable a vehicle is following, most likely inhibiting the performance of the developed prediction models. Improving the availability and quality of such data would allow to further advance ETA predictions. Additionally, in this study, the ambiguity of line numbers might have resulted in the loss of some journeys. Circular artefacts were also discovered that can be explained using a geofencing method that moves the vehicle position onto the geofence boundary unless it has arrived at a stop. Such manipulation of the data stream could hamper prediction efforts, although the assessment is difficult without a ground truth.

This study used benchmarks to make the findings easily comparable to other studies. We have shown that a simple metric such as mean error cannot be used to objectively compare algorithms. To make an informed decision, it is crucial to use several metrics. The Sharpe ratio was used to account for the standard deviation in addition to the mean error, which proved to be a better measure than a simple MAE. Furthermore, the importance of assessing the error distribution was highlighted where it was possible to see that, for example, the mean speed benchmark performed especially poorly if the vehicle was stationary, which is impossible to deduce from simpler metrics.

The extracted journey data was affected by artefacts such as repeated position records at either end of the journey. Therefore, it was necessary to remove such artefacts and assess their impact on the final prediction. Unexpectedly, the step-wise cleaning approach did not improve the overall MAE of the predictions compared to the raw data. This can be explained by the fact that the RNNs perform especially well when predicting the stationary positions of buses at stops or idling at either end of the route – over 20% of the journeys have repeated positions at the start. This is a large

number of predictions that can be made with exceptional accuracy, thus giving the appearance that the predictions are more improved the noisier the data is. In other words, the more stationary points a dataset contains, the better will be the overall prediction accuracy. Such a model is naturally not very useful in an operational context where the emphasis is on predicting vehicles in motion. Therefore, it is crucial to assess each developed algorithm in depth by examining errors along the route and focusing on any patterns that might be contained in such data. Even though the overall prediction rate did not improve, the RNNs did perform better than the benchmark at stops along the route and with a general better accuracy along the route while the vehicle is in motion.

Using alternative representations of the same target, considerable improvements in accuracy have been made (66 m between the trajectory and unconstrained coordinates). The intuition is that by simplifying the problem and reducing the prediction space, the model will achieve better results. In practice, this meant that predicting unconstrained coordinates did not perform well, whereas limiting the prediction space to the trajectory and subsequently transforming the problem to an ETA prediction improved the results 10 fold. The overall winner was the ETA prediction. Operationally, this could be considered the most important algorithm as ETAs, for example, displayed at bus stops or in mobile apps, could be considered more important than short-horizon predictions at all points along the route. However, a short-horizon prediction compensating for transmission delays in 'live' location representations on the web or mobile apps, will make the user experience better, benefiting those passengers who rely on such features.

Overall, this study highlighted the urgency to make all available data accessible to develop the best data-driven solutions in public transport. It furthermore illustrates the importance of not only relying on mean-based metrics but using a selection of different metrics in combination with geographical error representation to objectively assess any prediction algorithms. Additionally, even though in theory modern deep learning methods should learn to predict a target in any format, in practice they perform best if faced with the most simple representation of the task. As a conclusion and suggestion for further work, it is necessary to address the highlighted lack of data, as well as the lack of benchmark datasets. Furthermore, it is worth to consider the development of an evaluation framework specifically tailored to public transport prediction methods, consisting of a collection of different metrics and a formula to assess the geographical variation of errors.

Chapter 4

Bus Journey Simulation to Develop Public Transport Predictive Algorithms

Abstract

Encouraging the use of public transport is essential to combat congestion and pollution in an urban environment. To achieve this, the reliability of arrival time prediction should be improved as this is one area of improvement that passengers frequently request. The development of accurate predictive algorithms requires good-quality data, which is often not available. Here we demonstrate a method to synthesise data using a reference curve approach derived from very limited real-world data without a reliable ground truth. This approach allows the controlled introduction of artefacts and noise to simulate their impact on prediction accuracy. To illustrate these impacts, a recurrent neural network next-step prediction is used to compare different scenarios in two different UK cities. The results show that realistic data synthesis is possible, allowing for controlled testing of predictive algorithms. It also highlights the importance of reliable data transmission to gain such data from real-world sources. Our main contribution is the demonstration of a synthetic data generator for public transport data, which can be used to compensate for low data quality. We further show that this data generator can be used to develop and enhance predictive algorithms in the context of urban bus networks if high-quality data is limited, by mixing synthetic and real data.

4.1 Introduction

Cities around the world are trying to shift personal traffic to public transport to reduce congestion and environmental impact. A crucial part of such a strategy is to make public transport as convenient as possible. Bus passengers often rely on Real-Time Passenger Information (RTPI) systems at bus stops, online, and in mobile apps. These RTPI systems can be unreliable [2] which is inconvenient for passengers. In general, passengers assign different priorities to certain aspects of public transport. Reliability and safety are considered the two most important [94].

The importance of making especially buses as attractive as possible in comparison to private vehicles is highlighted in historical records. In the UK, 4.8 billion bus trips were made in 2018/19, representing 58% of all public transport journeys [108]. These journeys amounted to 27.4 billion km travelled and saved approximately 96 million tonnes of CO_2 [109]. However, since 1985, bus travel has been steadily decreasing by a total of 0.7 billion. As other public transport modes, such as trains in most areas,

cannot be a replacement for local bus services, this suggests that a larger share of passengers opt for private vehicles. This is reflected in the continuous upward trend of car traffic on British roads [108]. To encourage potential passengers to use public transport, it is crucial to make it as attractive as possible to reverse these trends. This will ultimately have a positive impact on the environment and congestion levels in urban settings. However, the mentioned data are pre-pandemic, thus the long-term impact of the pandemic on public transport cannot currently be anticipated.

Other studies also highlighted the importance of accurate Estimated Time of Arrival (ETA) predictions to improve customer experience [15]. Many public transport providers have developed mobile apps, which give 'live' positions of vehicles. Passengers can use this technology to decide when to leave the house to catch a bus without having long waiting times at a bus stop. However, we previously noted the latency of this information caused by delays of wireless network infrastructure and the fact that data in our operational area passes through a number of 3rd party systems [110]. Therefore, the RTPI system might suggest that a vehicle is further away than it is in reality. This could cause a passenger to miss a bus and thus unnecessarily inconvenience them. In Bournemouth, one of the two cities used as an example in this study, the latency of the internet-based 'live position' is approximately 30-40 s. To alleviate this issue, we have proposed a short-horizon prediction which will be useful in the further development of ETA and long-term predictions and in bringing the 'live' locations closer to reality. The commonly deployed Automatic Vehicle Location (AVL) systems [96], could provide data for such approaches.

To compare any potential model, the assessment of their performance is of crucial importance, this has to be reported in a way that allows to replicate and compare the results. However, this is not possible in all cases, as some authors report relative errors [45, 82, 54] and no consistency in the reported parameters can be distinguished. The precondition for all machine learning algorithms should be verifiable, and the RoyalSociety's report highlights this as a central feature [83]. This has also been recognised in the healthcare sector, where guidelines exist for the development and reporting of predictive models [84]. The difference in standards might be explained because ETA predictions do not affect the health or safety of a passenger, and a spurious algorithm might at most cause inconvenience rather than physical harm. However, for an operating company, this might cause a loss of revenue through a decline in patronage, and the society as a whole might be subject to more congestion, which could simply be reduced by providing accurate ETA predictions. Furthermore, the doctrine of science is replicability. The reproducibility crisis is most prominently known from psychological research [85] however due to its notoriety, it has been actively addressed [86]. It has also been identified as a problem in 'harder' sciences such as biomedicine [87] and also artificial intelligence [88]. Although results gained from machine learning techniques might be considered hard evidence, because the final model is based on mathematical concepts, they often suffer from similar problems as seen in psychology, where the research is often subjective to the researcher. The similarities between the two fields are that the findings cannot usually be explained due to the 'black-box' effect. The field of psychology has now begun to apply lessons from problems seen in machine learning research [86]. A suggested way to address such problems is meta-science that could shed light on the true accuracy of findings [89]. However, this relies on comparable accuracy measurements, which is not found in a large proportion of the public transport literature. Therefore, comprehensive reporting standards are urgently needed in the field of predictive bus transportation research. This as a consequence

poses the issue that high-quality data are required to develop good predictive models. We and other researchers have highlighted that data quality issues need to be considered in the context of public transport research [110, 111, 112, 113]. Therefore, in this study we demonstrate a method to synthesis bus journeys based on limited and low-quality data. This allows on the one hand to generate a hybrid dataset to develop models from. On the other hand, it has the potential to be used to generate synthetic datasets that can be used for benchmarking in an attempt to combat the highlighted replicability issues faced by public transport research.

In our data, a notable lack of quality hampers the development of predictive algorithms. Quality issues include the lack of clear journey identification, linkable to a timetable, artefacts such as gaps in recordings, falsely reported line numbers, and direction of travel (inbound vs. outbound). These quality issues make it impossible to develop accurate predictive algorithms. Unfortunately, the simplest solution of recording high-quality historical data is not feasible due to closed source data collection by 3rd party companies. To address this issue, this study describes a reference curve-based synthetic data generator, which bases its assumptions on limited real-world data. This allows to test algorithms in a controlled environment and enables the injection of user-defined artefacts into the dataset to test their effect on prediction quality. We also show that mixing real and synthetic data improves the accuracy of the prediction.

4.2 Background

Methods for ETA prediction can include simple historical averages or statistical models. However, due to the complexity of ETA prediction, machine learning methods have become increasingly popular [18]. In recent years, artificial Neural Networks (NN) have revolutionised a number of other domains. Therefore, NNs should be expected to have similar potential when applied to bus ETA prediction problems. A comprehensive review specifically investigating NN applications in public transport [19] found that only 16% (12) addressed ETA of buses, whereas the rest of the studies applied the technique to other modes of transport. This suggests that the area of bus ETA prediction using NNs might be underrepresented in the context of public transport research. This relative absence of NNs to predict bus ETAs is striking as NNs have revolutionised other areas of data science, such as image and speech recognition [114, 115].

The challenge of all machine learning approaches is to fine-tune the model parameters, one solution is to use genetic algorithms [116] to optimise machine learning algorithms inspired by nature. Several innovative variations have been demonstrated in the recent literature, such as an algorithm inspired by the mating of red deer populations [117], or the simplification of parameter search with a simplified metaheuristic [118]. The same authors also demonstrate methods applicable to supply chain management using the Taguchi method to outperform conventional genetic algorithms [119] as well as the potential use of blockchain algorithms in the management of supply chains [120], additionally they show applications to predict photovoltaic electricity generation [121] as well as bioremediation [122].

Nowadays, the majority of buses have onboard AVL systems, which are equipped with GPS sensors and transmit the location of the bus at frequent intervals, typically ranging between 20 and 60s. The availability of vehicle locations are the basis for any ETA prediction and are accessible through the AVL system and do not necessarily need any additional investment in static sensors.

The biggest hurdle in the development of machine learning solutions generally is the difficulty of acquiring enough good-quality data to develop a useful algorithm. In some fields, this has led to the use of simulated data ranging from medicine [123] to geophysics [124]. Regarding public transport journey simulation, the literature is scarce. Some examples related to bus data simulation include bus platooning [125] as well as traffic simulation [126]. However, to the best of our knowledge, no study has investigated the use of simulated data to train a next step prediction model for urban bus networks. In many areas of machine learning research, benchmark datasets are common [127]. These allow researchers to objectively compare algorithms against each other. This is missing in the field of urban bus networks. Therefore, the presented data generator could allow to generate a standardised benchmark dataset that could lay the foundation for further research in public transport.

4.3 Real-world data processing

4.3.1 Data collection

Data is accessible via the infrastructure of our collaborators, and two British cities have been selected with the largest number of vehicles and access to recorded travel data. The AVL data were collected from two different bus operators from Reading (UK) line 17 and Bournemouth (UK) line 1 (Figure 4.1). Each vehicle transmits its position approximately every 40 s, which is recorded by the company providing the Electronic Ticketing Machines (ETMs) with the integrated AVL-system. Due to data handling by several independent entities, only a limited amount of information is transmitted. The available data are as follows.

- Timestamp
- Position (latitude and longitude)
- Line number
- Direction (outbound or inbound)

For the Bournemouth operator, it became apparent that the transmitted directions are often incorrect, and so are the line numbers when a vehicle changes its line during an operational run. The data collected in Reading had better integrity with a reliably transmitted direction, thus simplifying the data processing steps. Based on this limited information, it is not possible to match a vehicle to a timetable corresponding to the journey it is currently serving. A journey is a specific trip found in the timetable of a bus line, e.g., the outbound 9 AM service 1. In contrast, a route pattern (also referred to as ‘shape’) is the route as travelled on the road, which can vary slightly for each journey for the same bus service. In the example of line 1 in Bournemouth, there are several patterns which can include different starting points along the route, resulting in shorter overall journeys or slightly different routes. In both cities,

reliably matching a vehicle directly to a specific route pattern is not possible, as the unique route pattern identifiers were not accessible to us. Therefore, one route pattern for each city was arbitrarily selected and used to generate synthetic data, which is an acceptable approach as in the selected cities the differences between patterns are negligible.

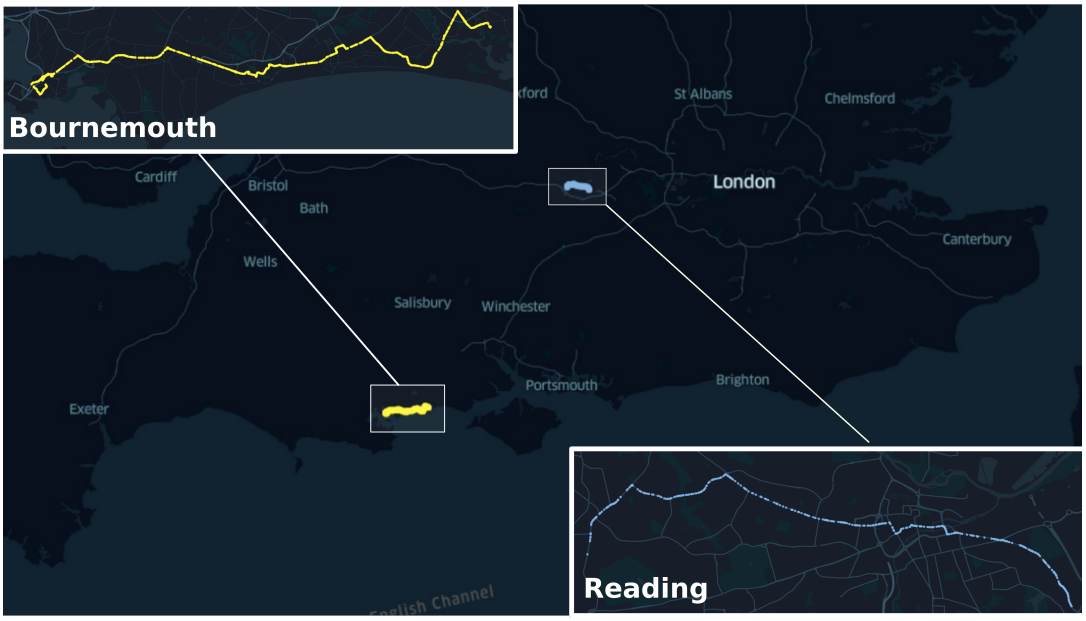


Figure 4.1: Location of both example cities and the journey shape used for all experiments. The line 1 in Bournemouth is shown yellow and the line 17 in Reading in blue.

4.3.2 Identifying route sections for filtering

The bus route used in Bournemouth is line 1, starting in the town centre towards Christchurch (Figure 4.1). The complete route shape includes longer journeys and therefore needs to be truncated. In the second example of Reading line 17 was used, which can have up to 90 different route patterns per direction with different runtimes and minor variations in route shapes (Figure 4.1). Additionally, a complicating factor is that the route follows a one-way system in the city center, meaning that the routes are different depending on the served direction. Therefore, a two-pronged approach was used. To initially filter journeys that were too far away from the shape, all available shapes for both directions were combined to a template shape. Any journey outside a radius of 3 times the mean distance to the template shape was excluded. The final filtering with the ability to enforce the direction was done using an arbitrarily selected route pattern from the many different patterns available for each line covering the entire length of the route. In the case of Reading these route patterns are mostly identical; however, in Bournemouth the patterns can be very different. We have described these issues previously [110].

4.3.3 Identification of individual journeys

Due to the lack of explicit identification of the journey, a heuristic approach was used to separate individual journeys, which will then be used as a basis to generate synthetic data.

Bournemouth operator does not reliably transmit the direction a vehicle currently serves. However, an observation made was that at the end of a journey vehicles stopped transmitting data for a short period of time. Thus, once it reappears in the data stream, a gap in the timestamps can be detected. A new journey was defined as a time gap of more than 15 min. If such a gap is detected, it is assumed that a new journey has started.

Reading operator reliably reports the direction of travel, making the identification of an individual journey easier. Furthermore, vehicles tend to serve the same line and do not change lines between runs, by selecting a single direction, large gaps in transmission timestamps can be observed, making the separation of journeys accurate.

4.3.4 Trajectory generation

It is assumed that the vehicles follow the identified outbound journey shape. This allows us to represent a journey as a trajectory, which is the distance travelled along the route shape. Using such a trajectory, a journey can be represented in two dimensions based on the distance travelled and the run time from the start of the journey.

4.3.5 Additional processing steps

To ensure a clean dataset, repetitions at the start where the vehicle did not move further than 10 m were removed and a journey is assumed to start once the vehicle has moved further than this threshold. The journey was assumed to have ended as soon as it had reached its maximum trajectory.

4.4 Synthetic data generation

The data generation process uses a heuristic data-based approach to generate synthetic journeys. This process is broken down into several sub-steps:

- The interpolation of the route shape as the reported points are not evenly distributed along the route.
- The identification of the normal run time for a journey is based on historical data, which also allows the identification of delays.
- The probability-based simulation of the delays.

The above steps are described in detail in the following subsections.

4.4.1 Interpolating the journey based on the route shape

A synthetic journey is generated based on future timetables. To avoid all vehicles starting at the same point, a time offset is added to the start time of the timetable, which is a random number between 0 and 40 s (the transmission interval). This is added to the scheduled start time. The distance that should be offset is then calculated by multiplying the offset by the average speed observed in the real world data 8 m/s (30 km/h). The timestamps are then interpolated to a user-defined interval – 40 s in the presented example. Calculating the time difference between two subsequent stops on the route segment gives the overall runtime. This can be divided by the transmission frequency of 40 s to give the number of transmissions expected on this route section. By assuming that the vehicle travels at a constant speed, the progress along the shape can be estimated, and the coordinates of the shape at the transmission points can be extracted. However, the coordinates of the reference journey pattern are not equidistant; the distances between consecutive reported locations vary between 5 m and 100 m. Therefore, interpolation solely based on the shape would give very different speeds depending on the shape of the road. This is avoided by generating an interpolation based on the distance along the route. The closest calculated distance of the shape coordinates is used to calculate the difference between the interpolation coordinate and the shape coordinate. If this distance is greater than 5 m, the two neighbouring points on the shape are used to interpolate the positions between these two coordinates to make the data more realistic. This does not account for variations in the speed or the curvature of the earth, but as the distance is at most 100 m, it is a reasonable omission. Furthermore, wider gaps appear on straight road sections and the frequency increases in meandering sections, making the proposed approach a good compromise.

4.4.2 The problem of determining delays

As arrival times at bus stops are not recorded, it cannot be determined whether a vehicle was on time or was delayed. An additional difficulty is that the journey times vary and depend on the time of day and weekdays. This variation in timetabled runtime compensates for the expected traffic status. TomTom, a location technology company, records congestion characteristics for different cities based on consumer GPS data. The data for Bournemouth indicate the percentage of delay that needs to be added to a journey at a certain time of day. The maximum in Bournemouth is on a Wednesday afternoon with an expected 71% increase in travel time (pre-pandemic)[128].

Most times of the day, the timetable overestimates the travel time compared to the expected time based on TomTom's data. However, it should be noted that vehicles travel between Bournemouth and Christchurch and the data only accounts for Bournemouth. Furthermore, stops to let passengers board or debark are not considered in the TomTom dataset. This means the timetable accounts for expected variations in traffic conditions and thus cannot be used to simulate vehicle delays.
Another avenue explored was the use of Google services to predict delays based on consumer data, which was not possible as buses travel in bus lanes, which makes the route very different from a prediction based on Google Maps.

Probability based simulation of delays

By assessing all journeys within the real-world dataset by weekday and hours of the day, a reference trajectory can be derived. This reference trajectory is simply the mean trajectory of all observed journeys (Figure 4.2a). As a result, the outliers are removed, and the reference curve represents the baseline of a 'normal' journey (Figures 4.2b and 4.2c). This allows to calculate the probability that a journey will be delayed or early for every time of each week day. Reference curves were generated using a centered moving 3 hour window except for the first and last hour where a truncated window was used. This gives the advantage that the time dependency of delays is simulated, meaning that a vehicle following a delayed bus will most likely also be delayed, thus approximating the delay propagation along a single line.

Journey generation

To generate a journey, the timetables of one week are queried and used as a template. The reason for this approach is that although the timetables for Bournemouth are available until the end of the current calendar year, this is not the case in Reading where only one week is available. As the timetable normally does not change drastically within the same year, this is a justifiable approach. Subsequently, the reference curve is queried and the following relevant data points are extracted:

- The mean reference trajectory.
- The standard deviation as well as 95% confidence intervals.
- The probabilities of delayed or early arrival with respect to the reference curve (Figure 4.2).

Delays

On the basis of the reference curve, the probability of a journey being delayed or early can be calculated. Whether a journey is delayed is decided by sampling from a normal distribution for each entry of the reference table, a random number *r* is generated and stored in a probability list $\{r_o...r_n\}$. These parameters double as a modification parameter to generate the delay or time gain. To remove variations of the list of probabilities, a Savitzky–Golay filter is applied with a window of 7 and a polynomial order of 3. A decision whether a vehicle will be on time, early, or delayed is made based on the smoothed probability list. A vehicle will arrive early if $r < p_{early}$. If $p_{early} < r < p_{early} + p_{delayed}$ vehicle is delayed. If neither of the conditions is true, the vehicle is assumed to be on time. To simulate the variations in time gained, the initially expected runtime t of the reference curve is calculated as well as the difference of the last position of the reference curve γ. The ratio of expected variation is calculated based on the confidence interval of the reference curve ν. Thus, the progress along the trajectory under the influence of a time gain can be calculated as follows:

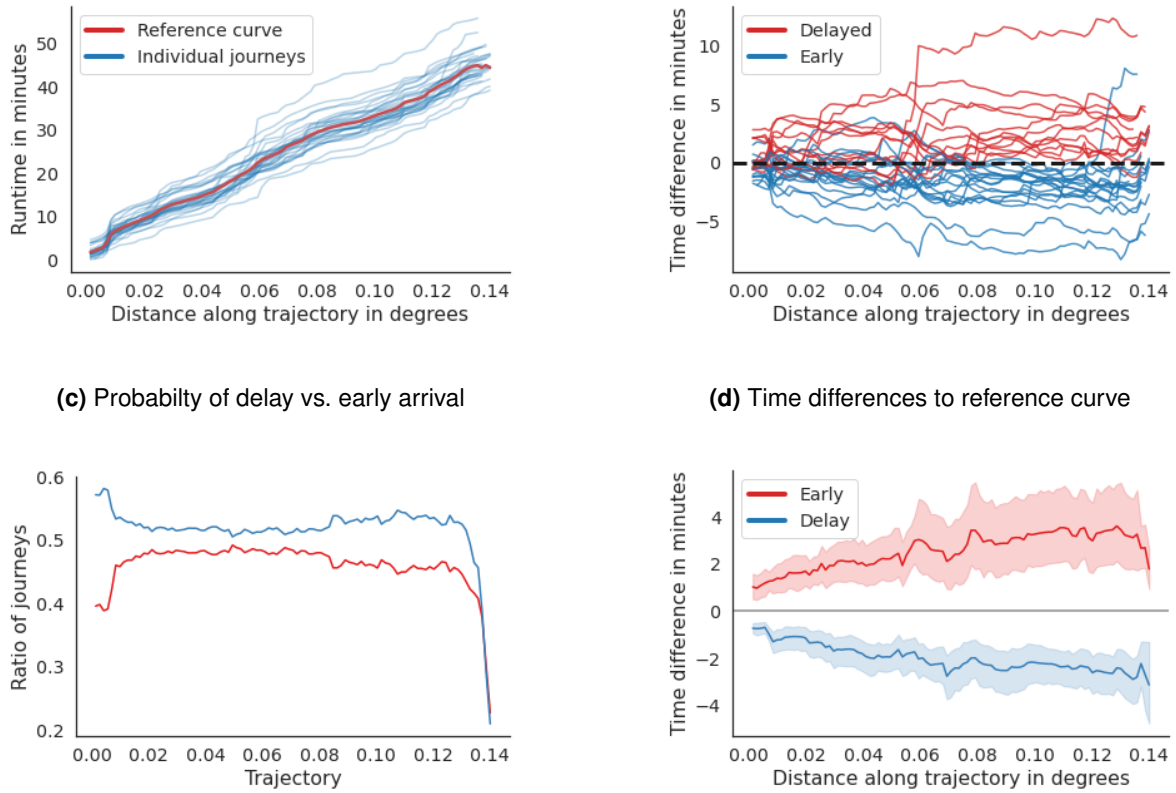
$$\nu = (\sigma_i/\gamma_i) * (R = \{^1_0) \tag{4.1}$$

$$P = P_{i-1} + t - (t \times ((0.9 \times \nu) \times 1.25)) \tag{4.2}$$

Where: ν=volatility, γ=reference, P=position, t =expected time at position

(a) The reference curve

(b) Journeys by delay



(c) Probabilty of delay vs. early arrival

(d) Time differences to reference curve

Figure 4.2: **a.** The historical trajectories of a one day block in Bournemouth (Tuesday 9-12 am). **b.** The relative difference from the reference curve along the trajectory. Journeys delayed at more than 60% of the positions are highlighted in red. **c.** Probability of travelling early or late on the trajectory. The discrepancy in the sum of the two conditions represents the fraction of vehicles that arrive on time. **d.** The average time difference to the reference curve with the uncertainty highlighted.

If the next position will be delayed, a random modification factor m is generated by sampling from a beta continuous random distribution (α=1, β=2). This tailed distribution was chosen as it makes large reductions in delay less likely and a vehicle will in most circumstances make up no or very little time. The delay *volatility* is defined as the ratio of the reference curve standard deviation to the reference curve itself multiplied by m. Additionally, the delay of the previous step d_{i-1} is calculated and subtracted from the current delay to prevent an exponential increase in delay. To account for random major changes outside the 'norm' of delay or time gains observed in the real data, GPS *noise* is generated using a uniformly sampled random number R which also acts as a weight of the additional delay.

Thus, a position with simulated noise can be described as (further explanations can be found appendix A):

$$\eta = \nu \times (R = \{^{1}_{0} + 1) \tag{4.3}$$

$$P = P_{i-1} + (t + [(\nu \times m) - d_{i-1} \pm \eta]) \tag{4.4}$$

Where: η = noise to be added, ν =volatility, P=position, t =expected time at next position

If the bus is most likely on time, the probability p of it being on time is used to generate an adjustment towards the reference curve as follows:

$$P = [P_{i-1} + t] - [p \times t] \tag{4.5}$$

Where: P=position, p = probability a vehicle is on time t =expected time at next position

The generated trajectory is then interpolated to give positions in time intervals of 40 s consistent with the transmission rates of the recorded data.

4.4.3 Injection of artefacts

The original data is affected by artefacts caused by the behaviour of vehicles as well as data collection issues. Three noteworthy artefacts have been incorporated into the simulation of the synthetic data and are described below.

Injection of GPS noise

GPS recordings are affected by noise which can depend on the surrounding environment, such as high-rise buildings. In the cities used in this study, buildings tend to be low, and thus effects due to reflection of the GPS signal are unlikely and have not been observed. To simulate the inaccuracies of the GPS recording, random noise sampled from a normal distribution (mean=0, σ=7) is added to latitude and longitude.

Injection of repeated locations

Due to operational reasons, journeys have scheduled buffers to allow vehicles to catch up with the timetable. This means that the vehicle often repeatedly transmits the same location at the start or end of a journey. At the journey start, 83% of the journeys have repeated locations, whereas end-repetitions are seen in 67% of journeys. The number of repeats varies depending on how long a vehicle is stationary. A skew-normal distribution [129] was fitted to both the start and end repetitions and this reference distribution is used to sample the number of repeats at either end of the journey. This artefact is optional and datasets with as well as without have been generated as in theory it is possible to gather journey data only for the journey itself without buffer times at either end.

Geofencing artefacts

The original data collected contained characteristic circular patterns. We have empirically demonstrated previously [110] that the origin of such characteristic artefacts are the geofencing methods used by some AVL-systems to determine if a vehicle has arrived at a bus stop [110]. Unless the bus has been very close to the stop, the AVL-system 'snaps' the real position of the vehicle to a circular geofencing boundary with a radius of 10 m. As this is an unusual artefact, it is generated optionally.

4.4.4 Data generation

For both cities, datasets were generated for 145 days and for three different conditions:

- a journey only with GPS noise,
- a journey with GPS noise and circular artefacts,
- a journey with GPS noise, and start and end repeats.

Additionally, a hybrid dataset was generated for the city of Reading containing 5000 journeys, of which 50% were synthetically generated and the remaining half were taken from the original dataset.

4.5 Prediction methods

4.5.1 Benchmarks

Two naïve benchmark algorithms were used to compare all models against.
Average speed: This method uses the average speed of a vehicle since the start of its current journey. Thus, it does not reflect any short-term speed variation. The calculated speed is used to interpolate the position of the vehicle from the trajectory of its journey pattern for the next 40 s.
Current speed: This method uses the last three transmitted positions of a vehicle to calculate its current average speed, hence accounting for temporary speed variations. The prediction is made by interpolating the position for the next 40 s from the journey trajectory.

4.5.2 Target representation

The target was represented as a trajectory, by projecting the coordinates onto the route pattern of a journey. This ensures that inaccuracies locating a vehicle off-route are removed. In practice, this method predicts a number representing the progress along the trajectory with a max of 1, which is the final destination. To illustrate the performance of the model, the trajectory can be decoded into coordinates to allow the calculation of a Haversine distance between the predicted and actual location, which is more intuitive than a loss based on the trajectory. Two variations of this target representation were used: **a.** the unconstrained progress along the trajectory, which could lead to a vehicle appearing to move backwards, **b.** the distance travelled in the next time interval added to the last known position, which enforces a forward prediction.

4.5.3 Input features

The features included were: coordinates normalised to a bounding box representing the operational area of the bus company, the time delta between consecutive recordings, the elapsed time from the start of the journey, and time embeddings as described below. The input features were min-max normalised.

4.5.4 Handling of time

The time information was split into its components to make it possible for the algorithms to learn periodic patterns. To achieve this, the timestamp was translated into the minute of the day, the hour of the day, and the day of the week. These were embedded in a multidimensional space as detailed in the architecture description 4.5.6.

4.5.5 Input windows

A moving window was applied to each journey. The window size was a minimum of 10 data points growing by one time step at a time until the end of the journey. This ensures a realistic simulation of the progress of a journey as would be observed in a real-world application.

4.5.6 Architecture

Two neural networks were used with identical architecture except for the Recurrent Neural Network (RNN) module [130], which was either a Gated Recurrent Unit (GRU) [131] or a Long Short Term Memory (LSTM) network [132]. The time embeddings were learned by the network in a multidimensional space. The dimensions were chosen as half of the possible number of values for each embedded variable. As an example, the hour of the day was embedded in 12 dimensions as the maximum number of hours is 24. These embeddings with a total of 52 dimensions were fed into a linear layer to reduce their dimensions back to the original number of time-based features. The output of the linear layer was concatenated with the remaining input features and fed into either a GRU or LSTM layer followed sequentially by a 1D batchnorm, a linear layer, a leaky ReLU, a second batchnorm and a final linear layer. To ensure that the outputs were bounded, a sigmoid function was applied.

4.5.7 Hyper-parameters.

To allow for direct comparison between the models, all training hyper-parameters were kept constant between the two cities. It is appreciated that this might not always yield the best performance, but it will illustrate the influence of the modifications made on the performance. The variables used were chosen through empirical exploration following the recommendations described by [106]. Each model was trained for 50 epochs using the one-cycle policy [106] with a maximum learning rate of 10^{-1} (Bournemouth) and 10^{-2} (Reading). As a loss function, the mean average error (MAE) was used.

4.6 Results and discussion

It is crucial to compare predictive algorithms using several different metrics to ensure a balanced interpretation of the results. Furthermore, it has to be kept in mind that in the presented example the two cities are considerably different. The most striking difference is the practice regarding journey shapes. The idea behind a journey shape is that it gives the exact route along the road of a certain journey. However, this is handled differently by the bus operators. In the example of Reading, each journey has an individual shape amounting to 90 shapes a day. These are mostly very similar or identical. In the example of Bournemouth, fewer shapes are used; however, the shapes are significantly different in length and route, highlighting the need for standardisation of public transport data. As a result, only a subset of the journeys in Bournemouth are similar enough to be simulated in one approach, thus this dataset contains fewer journeys than the dataset generated for Reading (17,115 vs. 7,839 journeys). These differences have to be kept in mind and are crucial for the interpretation of the results. The median accuracies for mean speed benchmarks in Reading are lower in all datasets compared to the current speed benchmark and are shown in Figure 4.3. The current speed benchmark for Bournemouth is comparable to the average speed benchmark. In the example of Reading this is not the case, and the current speed benchmark suffers from higher prediction errors compared to the average speed benchmark (Figure 4.3). An explanation could be that vehicles in Reading are more likely to stop for brief periods, which is reflected in a 13% increase of standard deviation of the travelling speed compared to Bournemouth. Interestingly, the histogram for the Reading benchmarks shows a peak around 80 m for the dataset with repeated start and end (Figure 4.4). This is explained by the benchmarking method, which uses the last three positions to estimate the average speed. Thus, a vehicle's speed can change from stationary to moving within 120 s or vice versa. Taking into account this time frame, 80 m/120 s corresponds to an average speed of 24 km/h, which is a realistic prediction for an urban bus network and is consistent with the estimated speed of the mean speed benchmark (Figures 4.3 & 4.4)).

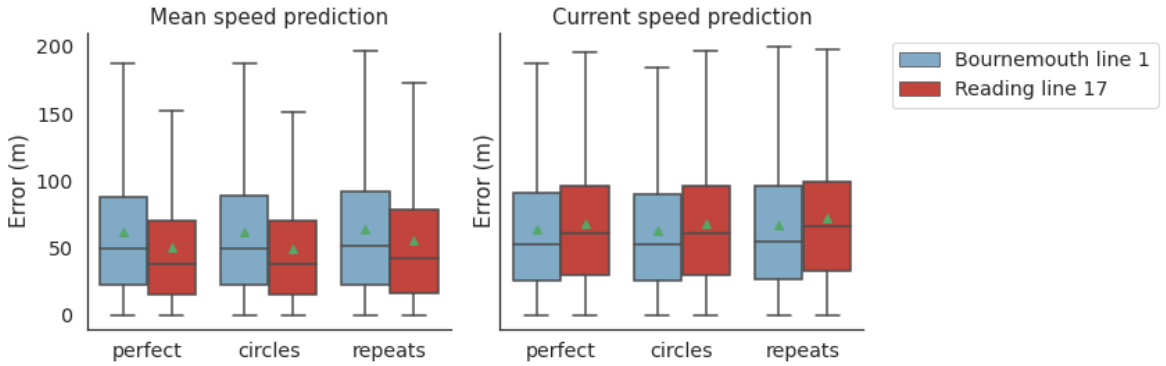


Figure 4.3: Boxplot illustrating the prediction errors of the two nïve benchmark algorithms for both cities.

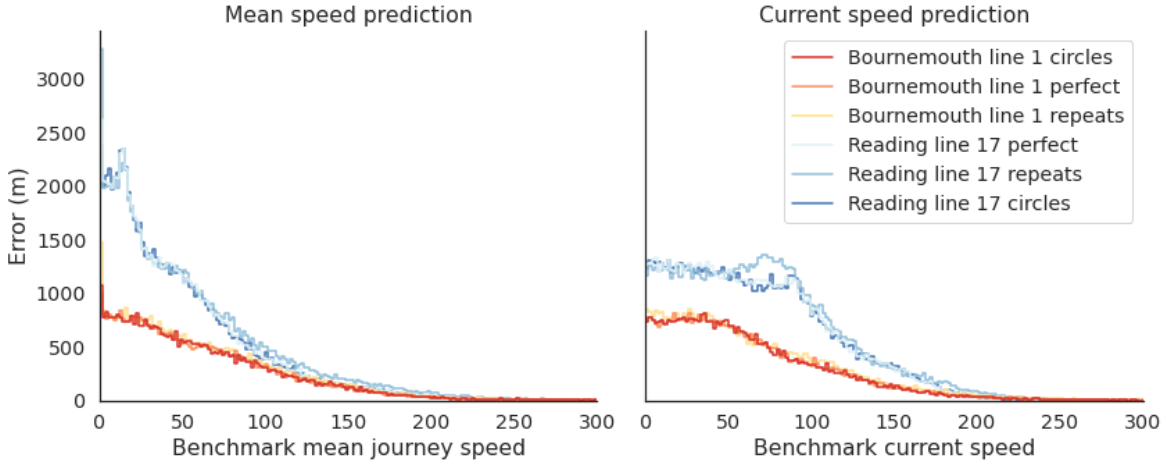


Figure 4.4: Boxplot illustrating the prediction errors of the two nïve benchmark algorithms for both cities.

4.6.1 Perfect journeys

The first set of experiments shows the 'perfect' synthetic journey. These are generated without any of the discussed artefacts and therefore should represent the simplest prediction problem. Poor performance of both architectures can be observed in the Bournemouth dataset. Both architectures perform virtually identical with a mean error of 63.8 m (σ=55 m) (Figure 4.5 a.). This is an accuracy comparable to the benchmarks (current speed: 64.2 m, mean speed: 62.1 m). This underwhelming performance could be explained by the smaller dataset compared to the Reading data, however, a more likely explanation is the variability of the journey shape and routes in Bournemouth, which naturally results in less realistic synthetic data. As a consequence, it is difficult to identify individual journeys from the original data. Furthermore, data generation suffers from the fact that vehicles do not follow a consistent route, which would be expected to cause unrealistic synthetic journeys. In contrast, the prediction for Reading performs well with a mean error of 41.5 m (σ=46.5) and 47.5 m (σ=47.2) for the GRU and LSTM respectively (Figure 4.5 a.). Both models significantly improve the error compared to the benchmark (current speed: 68 m, mean speed 50.7 m). As mentioned previously, this dataset contains more journeys per day, however, the most likely explanation of this performance improvement can be attributed to the uniform journey shape, which will reduce errors in the data generation.

4.6.2 Ticketing machine artefacts

The introduction of characteristic circular artefacts into the dataset would be expected to make any prediction more difficult. This is indeed observed in the predictions for Bournemouth. The average GRU performance was reduced by 2.5 m compared to the artefact free journeys. In particular, the performance of the LSTM did not decrease significantly and remained at 63.9 m (Figure 4.5 a.). Similar findings were observed in Reading where the mean error of the GRU increased by 5 m. Interestingly, the mean error of the LSTM decreased by 2 m.

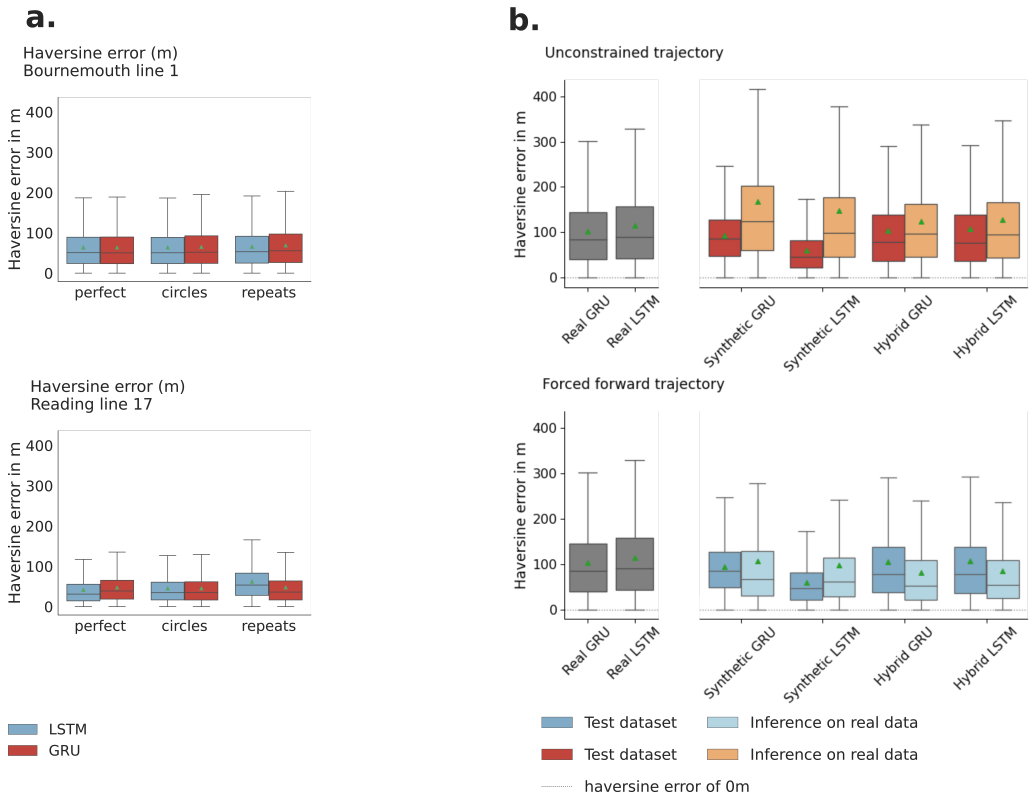


Figure 4.5: **a.** Boxplots for both cities and for each of the dataset and network architecture combinations. It is apparent that the performance in Reading is considerably better and the expected deterioration with the introduction of artefact can be observed. **b. top:** Boxplots showing the error ranges in meters for the unconstrained networks the grey boxes show a network trained on real data as reference. The red boxes show the error of the holdout portion of the synthetic or hybrid dataset the orange boxes show the inference errors on the real dataset. **b. bottom:** Boxplots showing the error ranges in meters for the forced forward networks the grey boxes show a network trained on real data as reference. The dark blue boxes show the error of the holdout portion of the synthetic or hybrid dataset the light blue boxes show the inference errors on the real dataset.

4.6.3 Repeats at start and end

The introduction of repeats at the start and end of the journey had a strong impact on the prediction performance. The mean prediction error in Bournemouth increased by 5 m and 2 m for the GRU and LSTM, respectively. In Reading, the GRU prediction worsened drastically by 24 m, whereas the LSTM was not affected and remained at 47.8 m (Figure 4.5). This is an intuitive response of the LSTM which, due to its ability to forget irrelevant information, is able to focus on the data relevant for the next step prediction.

4.6.4 Using hybrid data to improve predictions

The described hybrid dataset was used to demonstrate a possible application. As an intuition, it was assumed that the addition of synthetic data, which are cleaner and not affected by uncontrollable artefacts, should improve the overall prediction. When using an unconstrained prediction along the trajectory, however, this is not observed, and a model trained on purely synthetic or hybrid data performs worse on inference on real data (Figure 4.5). However, this is not the case if the prediction is forced forward as described in Section 4.4.4. If the prediction space is limited, an improvement in the inference accuracy of networks trained on both the real-world dataset can be observed both in the purely synthetic and the hybrid dataset. The greatest improvement can be observed if hybrid data were used for training (Figure 4.5 b).

4.6.5 Discussion of results

The results of this study show that the addition of synthetic data can improve predictive algorithms, which suffer from data quality issues. The use of synthetic data is used in many settings [133], such as healthcare settings to preserve privacy [134] but is also used in the assessment of algorithms such as feature selection methods where the control of features is important [135]. Some authors have also used synthetic data to estimate the upper theoretical limits of predictive algorithms [136]. The generation of hybrid datasets consisting of both real and synthetic data is less common, but examples such as from computer vision exist [137] or for classification problems with heavily unbalanced data [138]. Furthermore, some studies used synthetic data to augment small datasets for example to improve pandemic datasets and the associated machine learning models [139]. Examples from the field of public transport are rare and mostly focus on optimisation of transport networks and specifically bus routes to minimise delays [140, 141, 142]. However, in general, a knowledge gap appears to prevent the combination of simulated data with machine learning algorithms [143], which could be beneficial to improve many areas especially in public transport research. This study demonstrates the use of such hybrid datasets to improve prediction quality. Furthermore, it highlights the lack of framework previously noted by us [144]. A prediction accuracy comparison with the wider literature for this study is not possible as similar research aims to solve different problems. The reason for this is that the research focus regarding short horizon predictions are focused on time frames of >5 min [145, 146] or are defined as a distance rather than a time horizon [147]. Shorter prediction horizons are found in the literature but are aimed at predicting different metrics such as speed [3] or the elimination of bus-bunching [148]. As there are, to the author's knowledge, no examples in the literature predicting the position of urban buses in an ultrashort prediction horizon, a comparison with other studies cannot be drawn. Additionally, this study does not claim predictive superiority but demonstrates that the use of hybrid data can improve the accuracy of prediction. This knowledge will be of value to public transport researchers and can be applied to any prediction problem, as well as to any model architecture, to push the limits of the available data.

4.7 Conclusion

The importance of making public transport as convenient as possible is self-evident and could help increase passenger numbers and reduce urban congestion and pollution. Reliable predictions of the current position and arrival times of vehicles play a crucial role in this effort. However, this is being inhibited by the lack of reliable data, making any such algorithm development difficult.

Therefore, the described method of generating realistic journeys builds a bridge between low-quality recordable data and the real world. As a result, it is a platform to develop algorithms in a simulated and controlled environment, which can later be deployed in a real world scenario. Additionally, this platform allows to simulate user-specified artefacts as demonstrated by the repetition of positions or geofencing based disturbances. This study has highlighted several areas of improvement for urban bus network data to allow the development of reliable predictive solutions. The most striking observation was that any RNN-based predictions in Bournemouth barely outperformed the naïve benchmark. This is due to the varied route shapes and lengths of the same bus line, making generalisation unfeasible. Thus, it can be recommended from a managerial as well as software development point of view that either route shapes should be standardised between the lines or that the lines are subdivided based on their route shapes. This will greatly improve the potential of the collected data and the development of data-based software solutions.

The second observation was that the prediction performance can be improved if the data are as clean as possible. This means that technology providers need to collaborate to ensure the best possible outcome for public transport as a whole. Although geofencing methods to determine the arrival at a stop are useful, the produced artefacts of some systems do have a negative impact on the tested predictive algorithms. In addition, indicating whether a vehicle has started or ended a journey will help with overall prediction accuracy. The differences between the two example cities highlight the need for a national standard if accurate predictions are desired, universally preventing the need to develop a predictive system from the ground up for each city and operational line. This would be a big step forward to an implementation of mobility as a service and would benefit all public transport operators.

The limitations of this study are that the ground truth can only be approximated due to the lack of high-quality data. However, this is also the driving force behind the demonstrated approach to further advance this research and any other research relying on public transport data, the following key points should be considered for future research:

- Develop a standardised framework to transmit and record public transport data.
- Standardise the use of route patterns to ensure they can be used for data driven applications.
- Develop a benchmarking framework specifically for predictive algorithms in urban bus networks.

In the meantime, until such standardisations become reality, our data generation method described here is a good approximation of reality and a useful tool in simulating effects on urban bus networks.

Chapter 5

A Model Architecture for Public Transport Networks using a combination of a Recurrent Neural Network Encoder Library and an Attention Mechanism

Abstract

This study presents a working concept of a model architecture allowing to leverage the state of an entire transport network to make estimated arrival time (ETA) and next-step location predictions. To this end, a combination of an attention mechanism with a dynamically changing recurrent neural network (RNN) based encoder library is used. To achieve this, an attention mechanism was employed that incorporates the states of other vehicles in the network by encoding their positions using gated recurrent units (GRU) of the individual bus line to encode their current state. By muting specific parts of the imputed information, their impact on prediction accuracy can be estimated on a subset of the available data. The results of the experimental investigation show that the full model with access to all the network data performed better in some scenarios. However, a model limited to vehicles of the same line ahead of the target was the best performing model, suggesting that the incorporation of additional data can have a negative impact on the prediction accuracy if it does not add any useful information. This could be caused by poor data quality but also by a lack of interaction between the included lines and the target line. The technical aspects of this study are challenging and resulted in a very inefficient training procedure. We highlight several areas where improvements to our presented method are required to make it a viable alternative to current methods. The findings in this study should be considered as a possible and promising avenue for further research into this novel architecture. As such, it is a stepping stone for future research to improve public transport predictions if network operators provide high-quality datasets.

5.1 Introduction

A crucial part of making cities more sustainable is the transition from private transport methods to public modes of transport. In cities with existing transport networks, this means that operators need to make their services more attractive to potential passengers. For many patrons, convenience is a key area for improvement [94]. Therefore, it is crucial to improve the estimated arrival time (ETA) predictions, which allow passengers to better plan their journeys. Especially in the case of bus networks, passengers rely heavily on Real-Time Passenger Information (RTPI) systems at bus stops, online, and in mobile apps. Such RTPI systems can be unreliable [2], thus making the bus less attractive as a mode of transport. The UK has seen a steady decline in bus patronage since records began in 1985, bus travel has decreased by a total of 0.7 billion journeys [108]. Because local buses in most areas can only be replaced by private vehicles, this suggests that more passengers opt for their private cars, which can be seen in the steady increase in car traffic on British roads [108]. Taking into account the environmental and social impact of congestion, which causes a substantial waste of energy and human time, this is a troubling trend. Data for 2018/19 show that 4.8 billion bus trips were made in the UK, 58% of all public transport trips [108]. In sum, these travels correspond to an estimated 27.4 billion kilometres travelled and saved approximately 96 million tonnes of CO_2 [109]. In a recent study, the social costs of owning a privately owned SUV were estimated to be close to €1 million if the costs associated with pollution, infrastructure maintenance, and climate are taken into account. This study highlights that the ownership of private vehicles is associated with substantial costs to society and should therefore be reduced [149]. This highlights the importance of making bus networks as convenient as possible to attract travellers who are currently using private cars. If this is achieved, not only will it have a positive environmental impact but will also alleviate congestion issues in urban areas. Additionally, the pandemic has had an impact on the usage of public transport, and operators must restore public confidence in the safety of this mode of transport. Alongside these efforts, reliable ETA predictions will make a difference in the perceived passenger convenience of public transport [15].

We previously noted that the latency of data transmission from buses is caused by the delay in the wireless network infrastructure and the fact that the data in our operational area passes through a number of 3rd party systems [110]. Consequently, the RTPI system may suggest that the vehicle is further away from a bus stop than it is in reality.

The literature contains a wide range of approaches to predict bus ETAs. These range from more conventional methods such as historic averages [54, 53], ARIMA [55] or Kalman Filters [35, 77, 58, 61, 62, 65]. In general, such methods have low predictive power, and the introduction of Neural Networks (NN) drastically improved the performance of ETA predictions [61, 62, 65]. In the more recent literature, NN-based approaches have taken centre stage with some impressive results [27], however, further improvements compared to NNs were achieved using hierarchical NNs [51]. A particular focus can be found on RNN structures due to the sequential nature of ETA prediction problems. These methods include Long Short Term Memory (LSTM) networks [76], bidirectional LSTMs [56] or even convolutional LSTMs [81]. However, much more complex methods have become more common and tend to use different types of models for different aspects of the prediction task [79, 57, 60]. As there is no limit to the complexity of an ETA model somewhat more exotic methods, such as the artificial bee colony algorithm are also represented in the recent literature [67].

As a continuation of our previous work, we investigated possible architectural solutions to capture the interconnectedness of public transport networks and its effects on the accuracy of the prediction. All urban transport networks are, as the name suggests, networks consisting of directly or indirectly connected routes. Disruptions in a specific part of the network can have an impact on vehicles in different areas of the system [150]. Therefore, it is expected that any prediction that is made based on either the entire network or a more extensive part of the network could improve ETA and other predictions. Some examples that address a similar approach are studies that include vehicles on the same route in their prediction, allowing any algorithm to have a better view of the state of the network and thus improving prediction accuracies [72, 41, 29]. To the best of the author's knowledge, only one study used true network-based information including some short-term historical data from the entire network in their ETA predictions [30]. Examples from freight networks are more common and have demonstrated that a prediction based on multiple network-based models can improve ETA predictions [151, 152]. Another example demonstrates a similar approach for the prediction of taxi ETAs [153].

In sequential data, such as language translation, the so-called attention mechanisms can significantly improve predictive performance. The underlying idea is that the attention head will learn the importance of the order of words in a sentence and will focus more on the important parts of the query. In practice, this is achieved by using an encoder-decoder model with either an attention mechanism in between or a flavour of the attention head that doubles as a decoder. Various versions of attention mechanisms have been described in the recent literature [154, 155, 156].

In this study, we present a working concept of a model architecture allowing to leverage the state of an entire transport network to make ETA and next-step predictions. To this end a combination of an attention mechanism with a dynamically changing recurrent neural network (RNN) based encoder library. This study presents a pilot investigation into the suitability of this novel model architecture but does not claim superiority. The findings in this study should be considered as a possible and promising avenue for further research into this novel architecture.

5.2 Methods- data processing

To avoid confusion, the term "network" will be used for bus networks and road networks, and all neural networks are hereafter referred to as "models".

5.2.1 Data collection

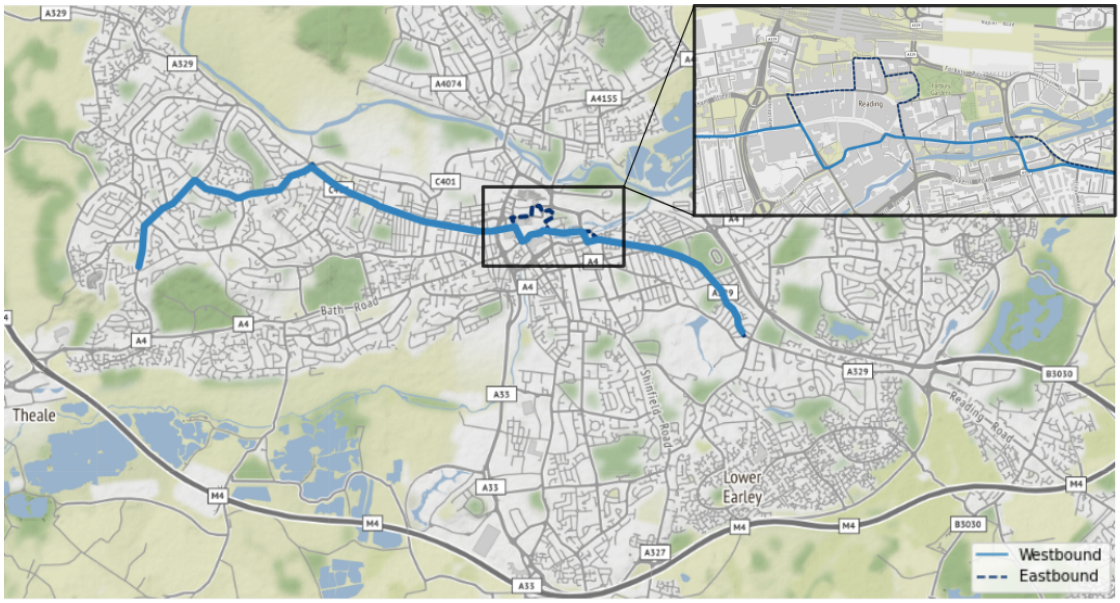


Figure 5.1: Map of Reading showing both the Eastbound as well as Westbound journey patterns. The blow-out area shows the city centre where the line negotiates a one-way system and therefore, runs on two separate routes depending on the direction of travel.

Data were accessed through the infrastructure of our collaborators. For this study, the city Reading (UK) was selected due to the largest amount of available data (Figure 5.1). As a line of interest, bus line 17 was chosen as it runs with the highest frequency and thus generates the most data. For this line, the predictions were made based on all vehicles which interact with this particular line, see Section 5.2.2. Automatic Vehicle Location (AVL) data were collected for all vehicles within the Reading bus network. Each vehicle sends its position approximately every 40 seconds, and the company providing the integrated AVL system passes the data on to several third-party entities before it is recorded. Due to the handling of data by several independent companies, only limited amounts of information are transmitted and retained. The available data are as follows.

- Timestamp
- Position (latitude and longitude)
- Line number
- Direction (eastbound or westbound)

Based on this limited information, it is not possible to match the vehicles with the timetables for the current journey. A journey is a specific trip found in the bus line timetables, such as the 9 AM eastbound service. An additional challenge is matching a vehicle to a specific route pattern. These patterns are slightly different routes that a vehicle on the same line might take. On the basis of the available data, a vehicle cannot be matched to such a pattern. Therefore, a route pattern for each city was arbitrarily selected and used to calculate route trajectories, which is an acceptable approach, as in the selected cities the differences between patterns are negligible.

5.2.2 Data processing

We have previously described a heuristic method to identify individual journeys, which was applied to the collected data [157]. In summary, it involves the identification of an individual journey based on the change in direction of a vehicle. Then a journey is represented as a trajectory, which is the distance travelled along a route. Finally, the repetitions at the start where the vehicle did not move further than 10 m were removed, and the journey is assumed to start once the vehicle has started moving and ends once the vehicle has reached its destination.

The final dataset included for the westbound direction 113,358 training samples and 24,214 holdout samples and for the eastbound 107,831 and 22,953 respectively.

Vehicle interactions

As our hypothesis assumes that additional information can be gained from vehicles which interact with buses on line 17, such interactions should be defined. A road section was selected in the city centre of Reading, which poses a bottleneck, most vehicles have to pass. Vehicles passing through the same section as vehicles on line 17 (east or west) were assumed to constitute an interacting line.

The lines that will be included to test the effect of interactions are identified using an area of interest in the city centre by Reading (Figure 5.2). A randomly selected subset of 100 k data points are then used to identify lines that travel through this area at any point in time in the same direction as the line of interest. This means that when predicting, for example, line 17 eastbound, not all other lines necessarily are assigned the direction "eastbound" as some might have different starting points, meaning the direction does not match the line of interest.

This results in 70 possible lines, of which many only contribute a minuscule fraction. The final selection is made by choosing those lines that contribute more than 3% of the total number of data points in the area of interest Figures 5.3 and 5.4.

Figure 5.2: Map of the Reading city centre showing the route of the eastbound line 17 in blue with a square indicating the area of interest where the selection of additional lines to be included in the model was made.

.

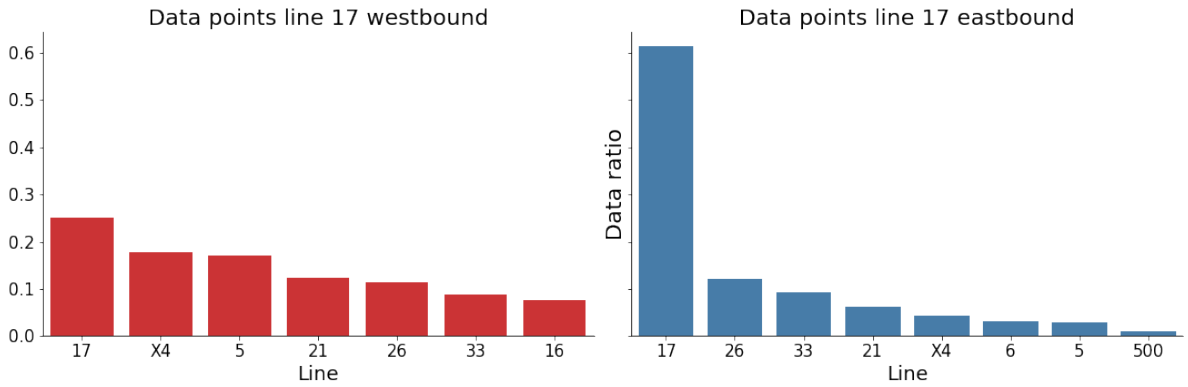


Figure 5.3: Ratio of data contribution for both directions an their interacting lines. Only those lines are shown which contribute more than 3% of data points in the area of interests as shown in figure 5.2. **Left** shows the ratio of data points to the target line with origin from each of the included lines for the westbound direction. **Right** shows the ratio of data points to the target line for the eastbound direction.

Input features

The features included were: coordinates normalised to a bounding box, the bearing reported by the AVL system, the time delta between consecutive recordings, the elapsed time from the start of the journey and time embeddings as described below. The input features were min-max normalised unless stated otherwise.

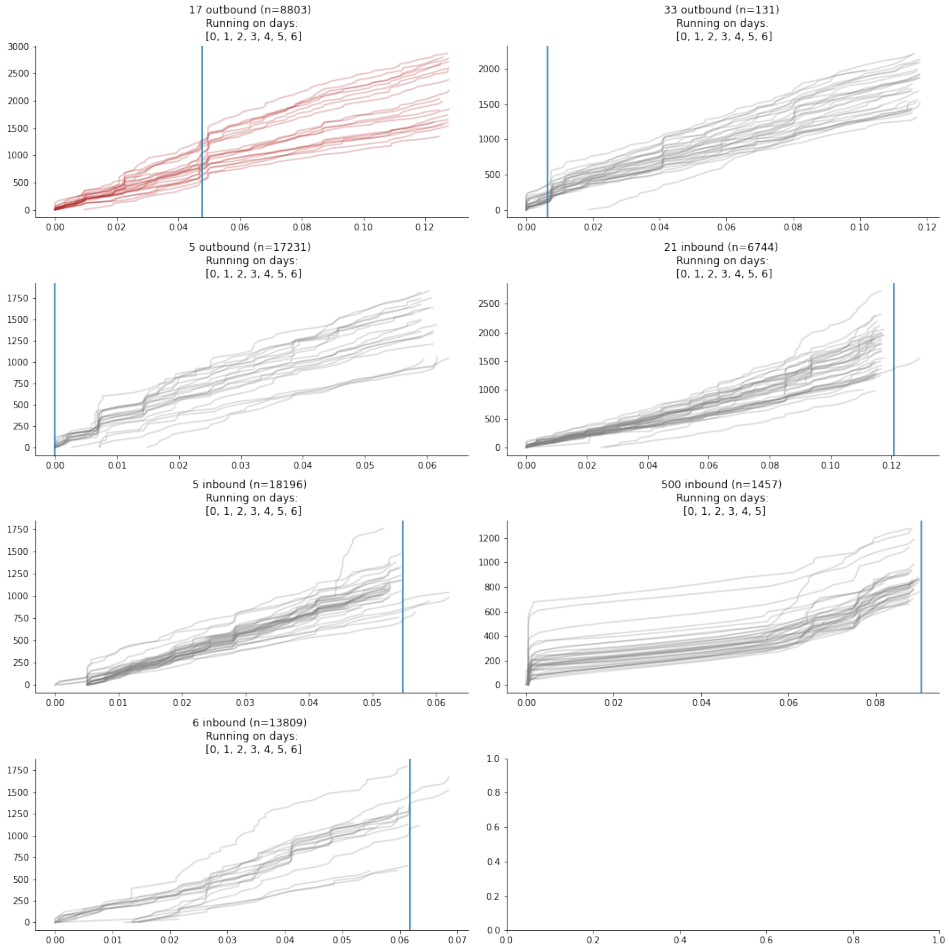


Figure 5.4: Interacting line trajectories blue line is the are of interaction from figure 5.2, red trajectories are the line of interest in this case line 17 eastbound.

Time embeddings

The time information was split into its components to allow algorithms to learn seasonal patterns. To achieve this, the timestamp was translated into minutes of the day, hour of the day, and day of the week. These were embedded in a multidimensional space as detailed in the architecture description.

Target encoding

Two targets were simultaneously predicted with the reasoning that this might give gradients with richer information and thus could benefit convergence (Section 5.3.3 for details of how the loss was calculated). The first target was an ETA to the final known position. It has to be kept in mind that this could also be a point along the route where a short journey ends and does not necessarily have to be the final stop on the trajectory. This is expressed as minutes from the last data transmission. The second target is the next position along the trajectory, which is equivalent to a fraction along the route and can be decoded to give exact GPS coordinates. As noted previously, reducing the prediction space improves the final prediction. Therefore, the target was expressed as the distance along the trajectory from the last known position, which can be simply added to the previous distance to give a location

along the trajectory. This enforces a forward prediction and is more useful, as a vehicle should never change direction in the middle of a journey. The combination of the two targets is an example which is more applicable, as network operators will in most cases be interested in ETA predictions and more accurate vehicle locations.

5.3 Methods- model architecture

Several models and techniques are combined to form a single workflow that accounts for the current state of the entire city network. In the following sections, each individual part is described, followed by a workflow that combines all models into one predictor.

5.3.1 Line based models

Each included line is assigned a model for both the westbound and eastbound directions. These models are simple Gate Recurrent Units (GRU), Figure 5.5. Additionally, a separate model was included for the target vehicle, which means that a specific GRU was used for vehicles of line 17, depending on whether they were the target or an interacting vehicle. The time embeddings were learned by the model in a multidimensional space. The dimension is half the possible value of each embedded variable. These 46-dimensional embeddings were fed into a linear layer and reduced to their original dimensions. The output of the linear layer was concatenated with the remaining input features and fed into the GRU. The dimensionality of the output as well as the number of layers was empirically derived based on the training results of a small subset of data (1000 samples). This was necessary due to the very slow training of the model, which will be discussed in Section 5.3.3.

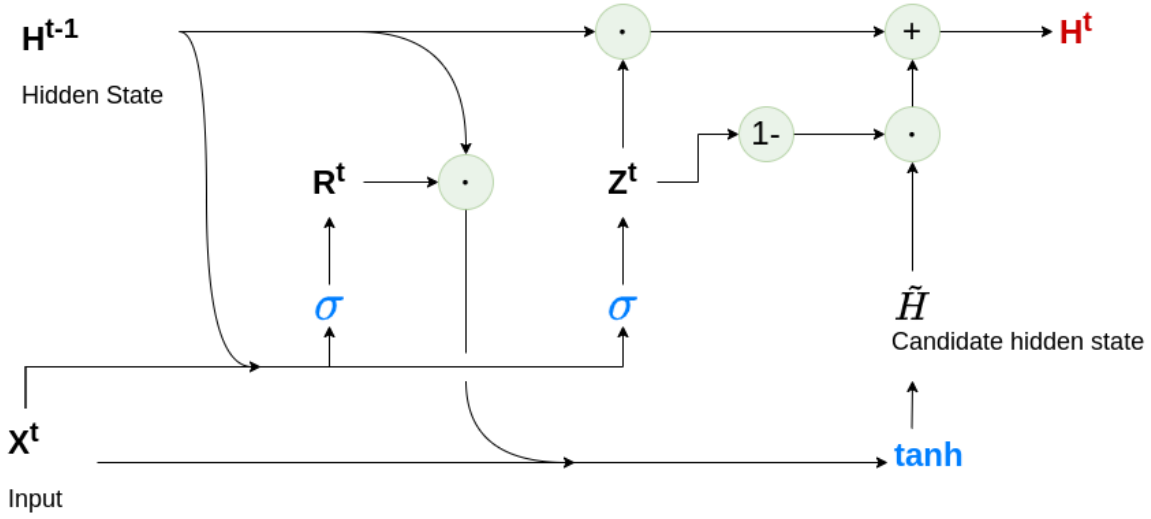


Figure 5.5: The architecture of a single GRU cell where:
H^{t-1} = previous hidden state; R = Reset gate = $\sigma(x^t) \cdot Wx + H^{t-1} \cdot Whr + br$; Z = Update gate = $\sigma(x^t) \cdot Wxz + H^{t-1} \cdot Whz + bz$; H = Hidden state = $tanh(x^t \cdot Wxh) + (R \cdot H^{t-1}) \cdot Wh$; $\hat{y}$ = Output = $H^t + Whq + bq$.

5.3.2 Attention mechanism

The outputs of the encoding line-based models are handed over to the decoder, using a user-defined number *n* of outputs $y_{t\text{-}n}$ from the last *n* historic network states. These historical network states are used as encoding *e* to be used by the attention mechanism Figure 5.6. The first step of the attention mechanism is to derive *Queries (Q)*, *Keys (K)*, and *Values (V)*. To obtain these values, a matrix product is calculated between *e* and the previously randomly initialised corresponding weights as shown below:

$$Q = e \times W_Q \tag{5.1}$$

$$K = e \times W_K \tag{5.2}$$

$$V = e \times W_V \tag{5.3}$$

where:

e = embedding of user-defined *n* network states

W_Q, W_K, W_V = randomly initialised weights for *Q*, *K* and *V* respectively

The decoder employs a scaled dot product attention as described by [154]. The authors used the following scaling method:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5.4}$$

where:

d_k = dimension of queries and keys

Q, K, V = Queries, Keys and Values respectively

The authors of [154] hypothesised that the reasons this scaling is necessary are issues caused by vanishing gradients of the softmax layer if the input data of the encoder had high dimensions. We found in our experiments that scaling did worsen the overall performance of the prediction model, and therefore the scaling was abandoned, and the simple dot product attention was modified by upscaling the attention by a constant of 1.5 which was empirically chosen by testing the performance of small subsets of data.

$$Attention(Q,K,V) = softmax\left(QK^T c\right)V \tag{5.5}$$

where:

Q, K, V, c = Queries, Keys, Values and upscaling constant (1.5) respectively

The output of this attention decoder was fed through a fully connected layer followed by a sigmoid layer to give the final prediction.

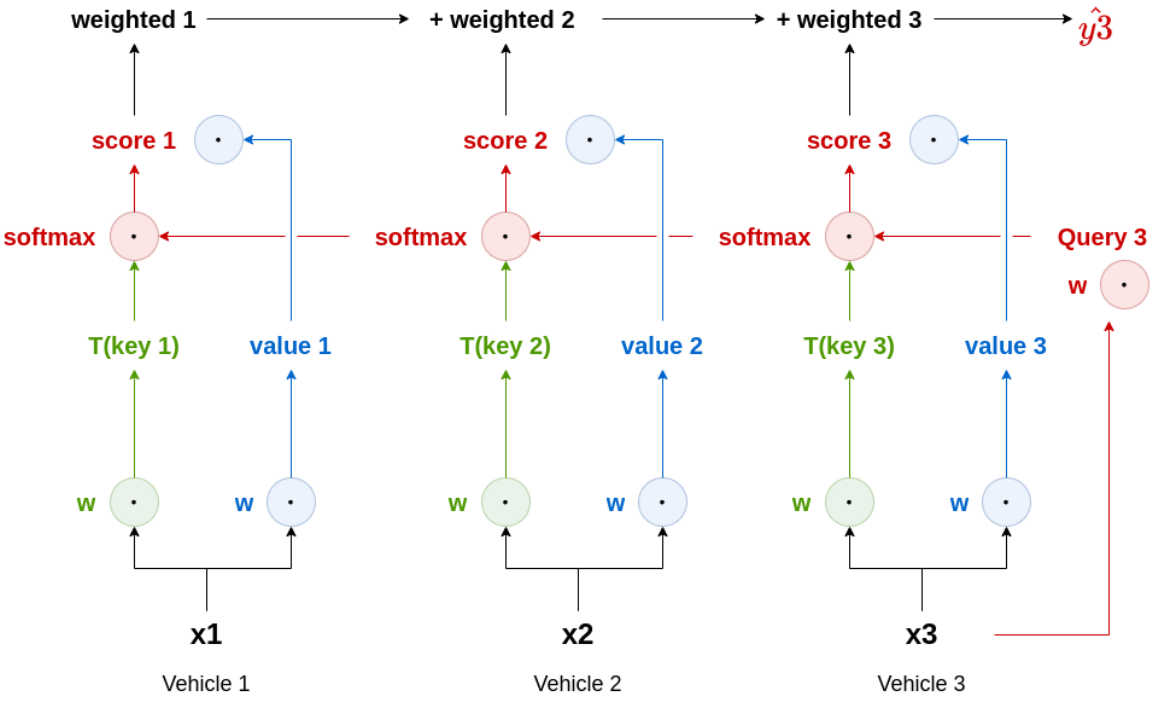


Figure 5.6: Schema of attention without fully connected layer or sigmoid. This illustrates the attention mechanism itself.

5.3.3 The training procedure

The training of this model ensemble is challenging, as it dynamically changes depending on the state of the transport network. At the same time, the weights of a line will be shared between vehicles on the same route and therefore could be accessed several times for each sample. The training procedure is performed in several steps, which are described below in detail. Pytorch [158] was used as the software library of choice.

Initialisation and optimisation

The GRU initialisation uses a random initialisation with an optional randomness seed for reproducibility for all parameters except for biases. The biases are initialised as zero. All parameters of the attention mechanism are also initialized randomly with the option of providing a seed.

The parameter initialisation for both types of models is performed before the model is defined. This means that in the case of the line GRUs n (n = number of interacting bus lines) sets of parameters are defined. These are stored as a dictionary, which are then loaded into each of the corresponding models. The same procedure is repeated for the attention model.

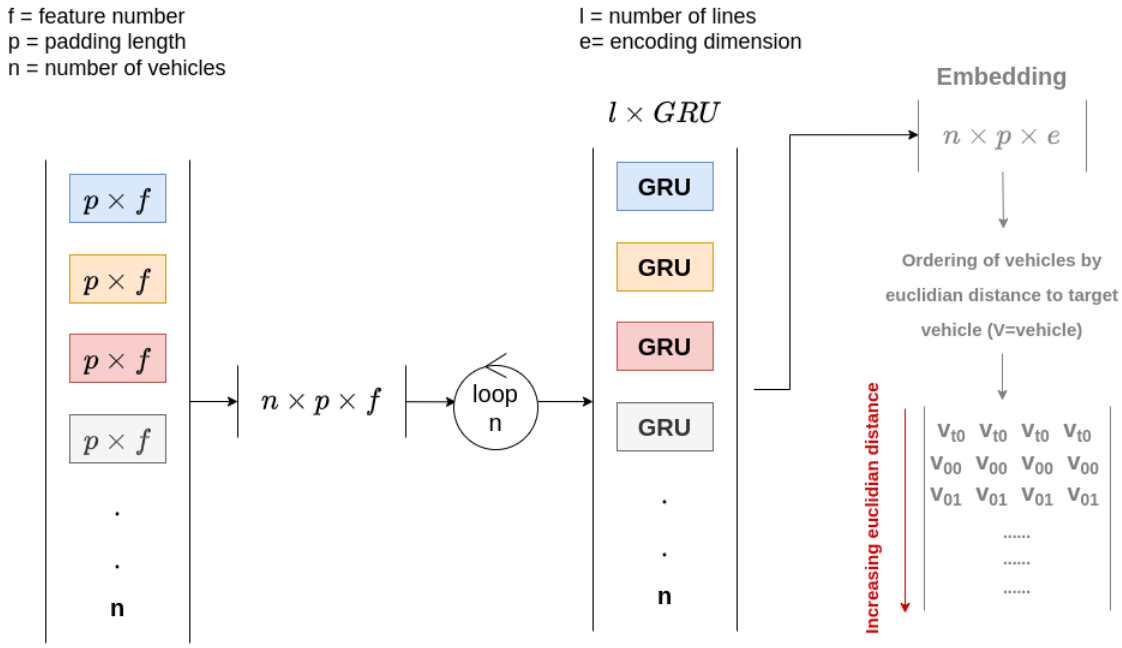
The handling of these weights poses a technical challenge, as sharing weights between several instances can easily prevent a successful backpropagation. If the parameters are explicitly stored, this causes issues by overwriting the gradients through inplace operations. This is avoided by the described procedure. As a result, it is possible to leverage Pytorch's automatic differentiation engine autograd [158]. In practice, this is done by iteratively adding parameters to an optimiser until all parameters of all models are included. This then allows us to backpropagate all models at the same time. Stochastic Gradient Descent [159] was used for this purpose with a momentum of 0.9.

Initial encoder pass

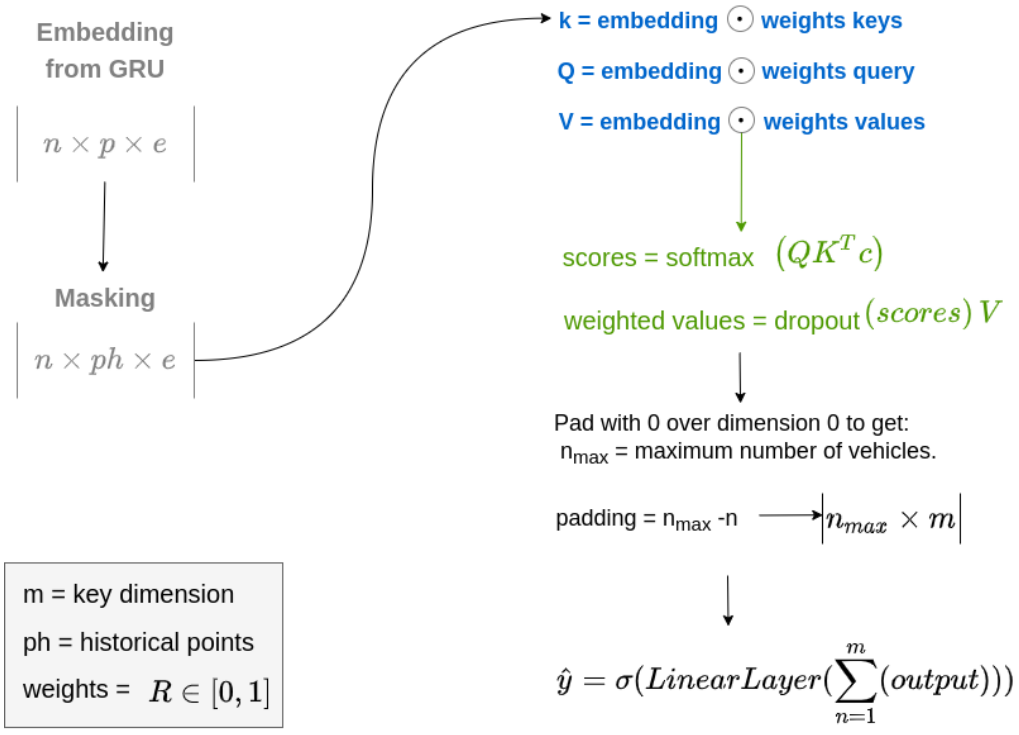
Each sample consists of the last 5 positions of the interacting bus lines. If fewer positions are available, zero padding is applied. Due to the dynamic nature of the data, where a varying number of vehicles with a varying number of lines make up the input data, an iterative approach for training is required. This means that for each vehicle, the corresponding line model is selected from a dictionary acting as a model library (Figure 5.7a). This means that if the vehicle is assigned line 3, the line-model number 3 is selected and the vehicle data are passed to this model. The output is temporarily stored for later use in the attention model described in section 5.3.3. This process is then repeated until all vehicles have been included in the initial training step. The number of vehicles will vary depending on the time of the day and week.

Pass through attention decoder

The temporarily stored encoded outputs of the line models are then passed to the attention decoder. The number of historical outputs from positions further in the past can be adjusted to maximise performance. There is no fast rule, and an iterative approach has to be used. The order used was based on the Euclidian distance of the normalised coordinates of all vehicles, where each individual bus is ranked by the distance to the target vehicle. Although the attention mechanism should be able to account for the order of vehicles, it became apparent through experimental tests that an increase in performance of approximately 30% could be achieved by ordering vehicles. As the interest lies in focusing attention on individual vehicles rather than on the progress of the journey itself, the output is zero-padded to the maximum number of vehicles seen in the dataset (in this example, 24). This is necessary to keep the dimension of the weighted values constant to allow them to be fed into the



(a.) GRU embedding method.



(b.) Attention decoder

Figure 5.7: a. GRU embedding method where each vehicle input is iterative fed into the corresponding line model to then generate the embedding matrix with the embedding dimension e. **b.** Attention decoder for a single embedding shown in figure a. This shows the generation of keys, values and scores as well as the weighting of the and finally the generation of the final output.

final fully connected layer of the attention mechanism, see figure 5.7 a & b . This model will return two decimal predictions corresponding to the trajectory of the line of interest (line 17) describing the progress along the route and the time to the final stop. This prediction is stored to be used for the loss calculation at a later point (Section 5.3.3).

Pseudo-batching

Due to the complexity of the described training procedure, a true batching of the data is not possible, as the number of underlying line models is dynamic and has to be individually adapted to each sample. Therefore, model training must be done iteratively for each sample. As backpropagation after each sample would cause instability of the model, an alternative was chosen where backpropagation was applied after chunks of 500 samples. This compromise is used to avoid having to wait until the end of an epoch before backpropagation can be run with the intuition that this should speed up the convergence of the model with fewer epochs needed for training. This method is of course not as effective as true batching, as it cannot leverage the parallel computing capability of a GPU and thus is a very slow process.

Batching

Although true batching is not possible because the composition of the dataset changes for each sample, a batching method was applied to the attention mechanism. To achieve this, the outputs from the line-based GRU models were collected into a batch, which was then handed over to the attention mechanism. This was hypothesised to increase processing speed and improve performance through a regularising effect [160]. To directly compare this batching method with the pseudo-batching method described in 5.3.3 equal chunk size and batch sizes were used to compare training times.

Loss calculation and backpropagation

After a user-defined number of samples, which are considered a pseudo batch, the loss is calculated based on the stored predictions. Due to the initialisation of the optimiser described in Section 5.3.3 the backpropagation can simply be calculated using a single optimiser and will be applied to all models. With the optimised model, a new pseudo-batch can be fed through the model.

As in this study, two targets are predicted, the next position along the trajectory as well as the time of arrival at the final stop, the Mean Average Error (MAE) is calculated for each metric individually, and both are summed during the training process using a constant to multiply the ETA loss so that both losses have the same magnitude. This allows to also monitor these individually during training to allow a better evaluation of the training progress.

5.3.4 Hyper parameters

Due to the slow training of this model and the large data set, it was not possible to use an automated method such as a gridsearch or a genetic approach to fine-tune the hyperparamaters. Thus, an empirical evaluation was performed using a small subset of the data (3000 randomly selected samples). The convergence speed and final loss for this subset were assessed to make an informed decision on the choice of suitable hyperparameters.

5.3.5 Performance evaluation

The loss performance as a whole but also for each target between models. Additionally, the loss distributions were monitored to assess any skewness within the training losses.
For any machine learning model, both training and testing losses have to be considered to allow an objective comparison of whether a model generalises well.

Human interpretable errors

Furthermore, to make the prediction error more interpretable, the next-step prediction is translated into GPS coordinates based on the shape of the trajectory. This then allows us to calculate the Haversine distance. Note that this will calculate the direct distance and not the distance along the route. Thus, the error could be smaller than the actual distance a vehicle would have to travel along the road.

Generalisation error

The ultimate goal of most machine learning algorithms is to be generalisable to new unseen data. The ability of an algorithm to generalise can be reduced by overfitting, therefore, the generalisation error was included as a performance metric. The generalisation error is calculated simply by subtracting the training error from the testing error [106]. In a perfectly balanced model, the generalisation error should be towards 0. A negative value indicates a tendency to underfit, while a positive value indicates overfitting.

To highlight the training and testing process of the data subsets, all metrics are shown alongside each other.

5.4 Results and discussion

The results shown in the following sections represent training runs on a subset of the data containing 6000 samples. This small subset was chosen due to time constraints caused by the inefficient training procedure, making training several models on the entire dataset prohibitively computationally expensive. Therefore, the models were evaluated on the basis of the training performance of the small dataset. For this reason, both the training loss and the testing loss were used to compare the models and decide on the best performing version for each of the two directions. These models were then trained on the entire dataset and described in Section 5.4.4.

5.4.1 Muting of all vehicles except target

A muted model in which only the target vehicle is considered in the prediction was used as a relative comparison to assess whether information from additional vehicles in the network will improve the prediction accuracy of the model. To test this, vehicles were muted during attention application. After the calculation of the weighted values, all values that were not from the target vehicle were multiplied by 0. This removes information about the network state and, as a hypothesis, should perform with lower accuracy compared to the model, which can leverage network information. The results for the eastbound direction showed that on a small subset of the dataset, the addition of network-based information improved the performance of the model, and a model without network information is inferior to one with information from the entire bus network (Figure 5.9) which confirms the hypothesis. The results are not as clear cut for the westbound direction, where the network-based model outperforms the muted version in ETA prediction, but there is no clear competitive advantage in the trajectory prediction. This can be explained by the fact that the generalisation error remains negative in the training process, suggesting that the model is currently underfitting. This is intuitive if the data contribution of the bus lines shown in figure 5.8 is considered, where the proportion of data contributed by target vehicles is greater than in the eastbound direction, which could indicate that longer training times are needed.

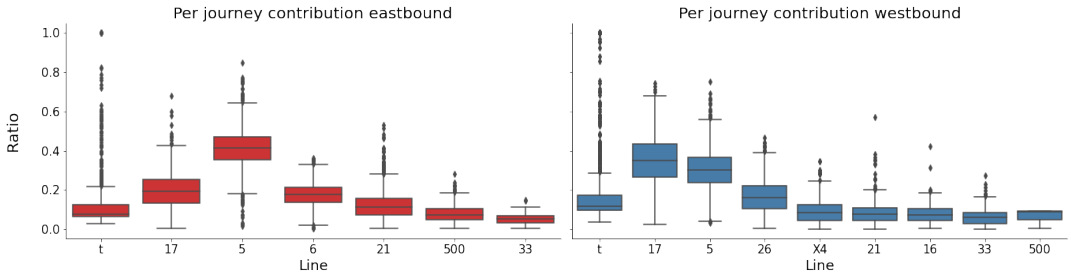


Figure 5.8: Per journey data composition of 1000 randomly selected journeys.

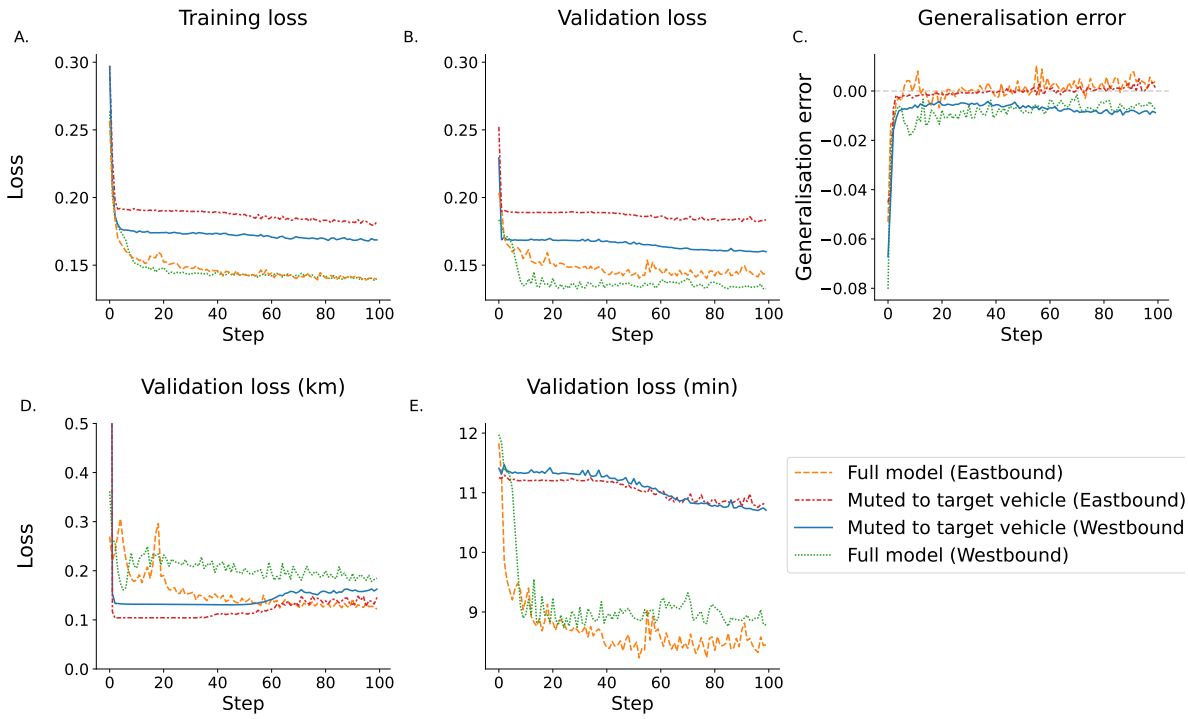


Figure 5.9: These figures show the model muted to the target vehicle itself thus removing all external information. This is compared to the network-based model clearly showing that the performance is reduced if the model does not have access to any external data. **A.** shows the training loss. **B.** shows the validation loss. **C.** shows the generalisation error calculated by subtracting the training error from the testing error. **D.** shows the estimated validation error in km. **D.** shows the estimated validation loss in minutes. (Note that subfigure (**D**) has a truncated *y* axis for illustration purposes).

Muting of all vehicles except target line

As a logical continuation of these findings, it can be assumed that if gradually more information is added, this should incrementally improve the model performance. To test this, all vehicles from other lines were muted to include only those running under the same line name and direction. This includes vehicles which will run on earlier schedules than the target vehicle, but also those that follow the target vehicle on later journeys. The results are shown in Figure 5.10. This modification was compared to the results of the full network-based model and are shown in Figure 5.4.1. For both directions, ETA prediction performance was reduced if the model was limited to a single line. However, both directions showed that the muted-line-based model outperformed the full network model. Interestingly, the muted model reached its best performance very quickly, whereas the full model converged slower. These results will be put into context in Section 5.4.1.

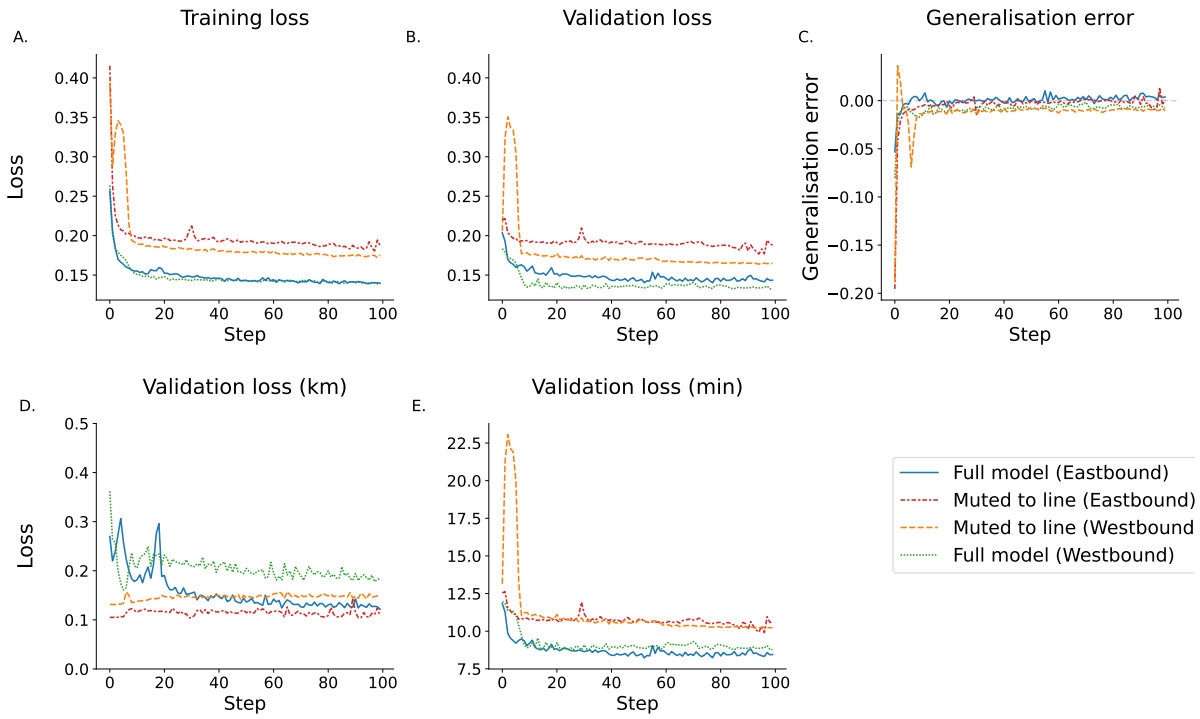


Figure 5.10: These figures show the model muted to vehicles of the same line regardless of their location in relation to the target vehicle. This is compared to the network-based model clearly showing that the performance is reduced if the model does only have access to indiscriminate information of its own line. **A.** shows the training loss. **B.** shows the validation loss. **C.** shows the generalisation error calculated by subtracting the training error from the testing error. **D.** shows the estimated validation error in km. **D.** shows the estimated validation loss in minutes. (Note that subfigure (**D**) has a truncated *y* axis for illustration purposes).

Muting of all vehicles except vehicles ahead of target

Finally, the focus was on vehicles on the same line, but specifically, only those that are ahead of the target vehicle at their last known position. It would be expected that this will pose an easier problem to model as, based on intuition, the vehicles in front of the target bus will have a more important role compared to those which are following the target. The results are shown in Figure 5.11 and are compared to the network-based predictor. For both directions, ETA predictions, were not affected by the muting compared to the full model. The muted trajectory prediction was superior to the full model in both directions. However, the generalisation errors for the eastbound direction are more negative than for the westbound direction, suggesting that the model is underfitting in this scenario, see Figure 5.11.

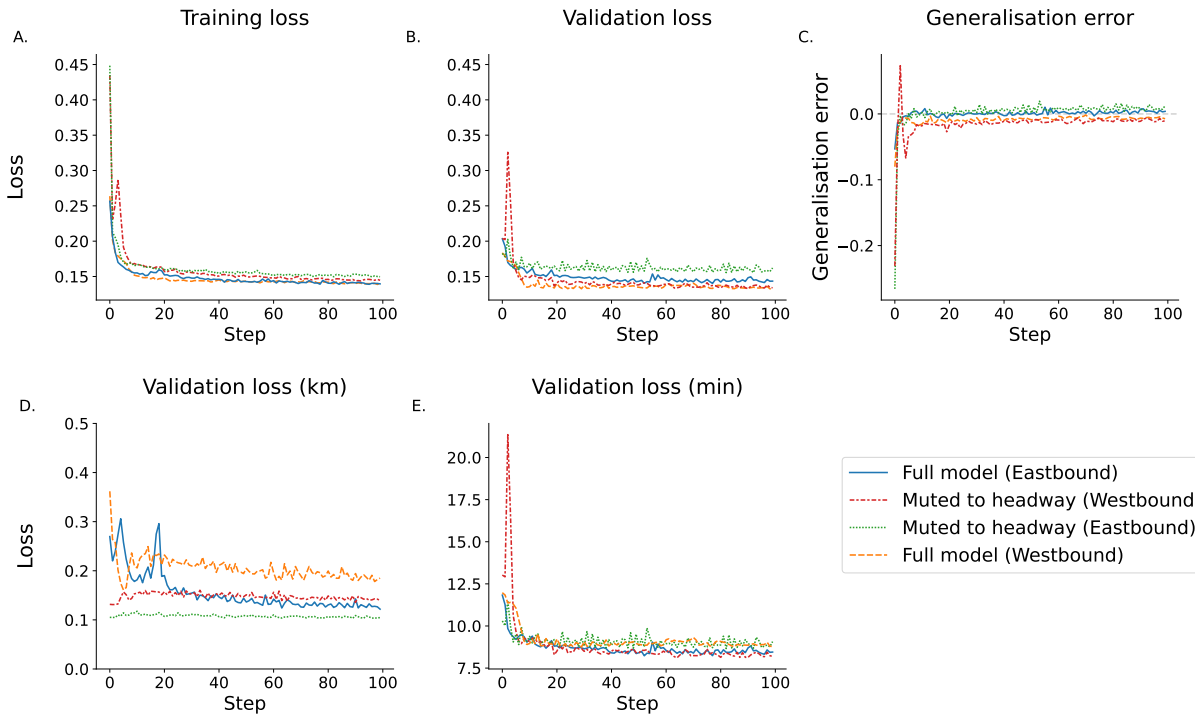


Figure 5.11: These figures show the model muted to vehicles of the same line ahead of the target vehicle. This is compared to the network-based model clearly showing that the performance is reduced if the model does only have access to indiscriminate information of its own line.**A.** shows the training loss. **B.** shows the validation loss. **C.** shows the generalisation error calculated by subtracting the training error from the testing error. **D.** shows the estimated validation error in km. **D.** shows the estimated validation loss in minutes. (Note that subfigure (**D**) has a truncated *y* axis for illustration purposes).

Summary of muting effects

It is expected that the more information about the network is available, the better the performance of the model. This was only partially true. In the experimental investigation, it was shown that the full model with access to all the network data performed better in some scenarios. Figures 5.12 and 5.13 show a summary of all muting methods for the eastbound and westbound directions respectively.

However, when the model is limited to the headway, this outperforms the network-based model in both directions of travel for trajectory predictions. It should be noted that the difference between headway and line-limited models in the westbound direction is negligible. The generalisation errors for all models are similar. This makes intuitive sense, as, for example, a delay in a vehicle ahead of the target would most likely translate into a delay for the target vehicle itself. However, a disruption of a vehicle running on a different line might or might not affect the target vehicle. This could be different for instances where several lines run on the same road for longer periods of time. Additional information from the entire network was expected to allow better predictions, as delays in one part of the system could propagate throughout the network and thus eventually affect the target vehicle. This was not the case but could be explained by data quality issues, as discussed before. If the interacting lines are severely affected by quality issues, this could, instead of improving the available information, introduce confusing noise,

thus making the prediction less accurate. Furthermore, it could be possible that a different choice of interacting lines could produce better predictions. In the presented example, the lines mostly interact at the end of the journey of the target line when considering the area where both travel on the same stretch of road (see Figure 5.4). When using lines that run in parallel for longer times, this might improve the importance of the interacting lines. Finally, the choice of target line 17 was made because it is the line with the most available data in the network, which means that the data from the line itself will always outnumber the data from the other lines (Figure 5.8). This results in the fact that the underlying line models of interacting lines will only be updated very infrequently and thus will take much longer to train. Therefore, it can be imagined that if the network-based model was trained for much longer and with more data, it could eventually outperform the headway-limited model.

When assessing the ETA predictions, the results are different, as in the eastbound direction the full model performed best 5.12, whereas in the westbound direction the headway limited model performed best [**?**]. This is an interesting finding, as the data contribution is different for the two directions. The eastbound direction contains, in addition to the target itself, line number 5 as the most common line (Figure 5.8). In the westbound direction, the most common line is line 17 (non-target vehicles). This seems to be reflected in the ETA prediction, where the headway limited model performs best in the scenario where the most common data come from line 17 (westbound), whereas in the case where the most common data come from a different line (line 5) the full model performs better (Figure 5.8).

Why this is not found in the trajectory prediction can be explained by the fact that in a setting where other lines contribute more to the data, this can be leveraged by the ETA prediction because intuitively the progress through the network of other lines might affect the final arrival time. In contrast, trajectory prediction is a short-term prediction for the next 40 seconds only, where the overall network state might be less important. For this example, vehicles immediately ahead of the target will be the most important to make an accurate prediction. This highlights the importance of designing not only a custom method for each line but also for each prediction target.

5.4.2 Performance of batching vs. pseudo-batching

In an attempt to speed up the model training mechanism, the pseudo-batching method (Section 5.3.3) was compared with the batching method (Section 5.3.3). It should be noted that this batching method only batches the input to the attention mechanism, and due to the complexity of the model selection for each line, it cannot be expected to achieve the same training speed improvements as batching of a conventional model. Furthermore, this experiment was run on an unrestricted prediction space for trajectory prediction, meaning that the model was able to predict any position along the trajectory in contrast to the other experiments discussed where the prediction was limited to points ahead of the last known position. Interestingly, this batching method did not achieve any improvement in training speed, but increased the average processing time required for each epoch by 24 % and 4 % for the westbound and eastbound directions, respectively (see Figure 5.14). Considering that the total number of journeys was the same for both conditions, this is a surprising finding but could be explained by the fact that the westbound data are less complex, meaning fewer data points from other lines are included. The eastbound dataset, on the other hand, has a higher proportion of data from other lines and is thus

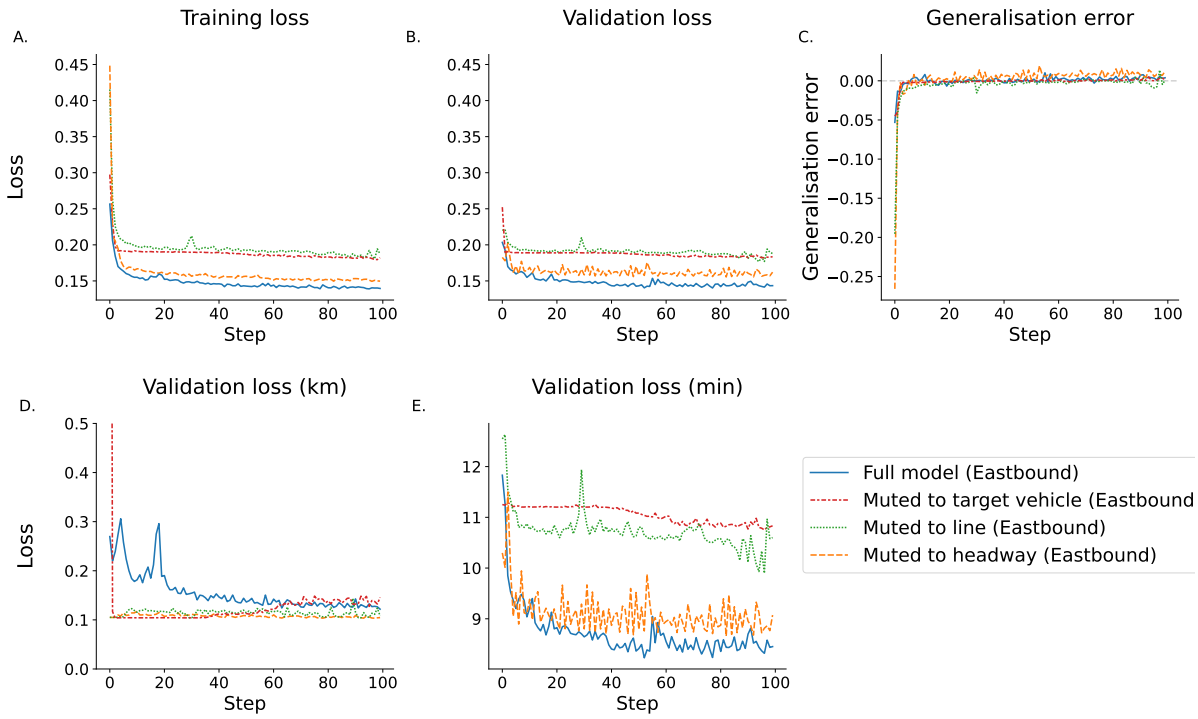


Figure 5.12: Training and validation of the **eastbound direction** for all muted models in comparison to the model with access to the entire network data. In the example of this direction the best validation loss was achieved by the model incorporating the entire network data.**A.** shows the training loss. **B.** shows the validation loss. **C.** shows the generalisation error calculated by subtracting the training error from the testing error. **D.** shows the estimated validation error in km. **D.** shows the estimated validation loss in minutes. (Note that subfigure (**D**) has a truncated *y* axis for illustration purposes).

more complex, which could mean that the performance reduction using the GPU processing is less pronounced. It can be hypothesised that although performance was not improved in this dataset, with increasing complexity, GPU processing might become a more efficient training method. This could be useful in a setting that, for example, includes all vehicles at any time in the network.

Although no training speed improvement was observed using this batching method, a benefit of using batching could be that applying batching to the attention mechanism could have a regularising effect on the model gradients [160]. This is due to the phenomenon that very small batch sizes, for example, a single sample as is the case in the pseudo-batching approach, can make the model unstable due to the high variance of the gradient estimates. It should be kept in mind that the line-based GRU models were not batched, and using a method that could apply batching to these models might further increase the model performance. However, this is to date not possible with the current model architecture.

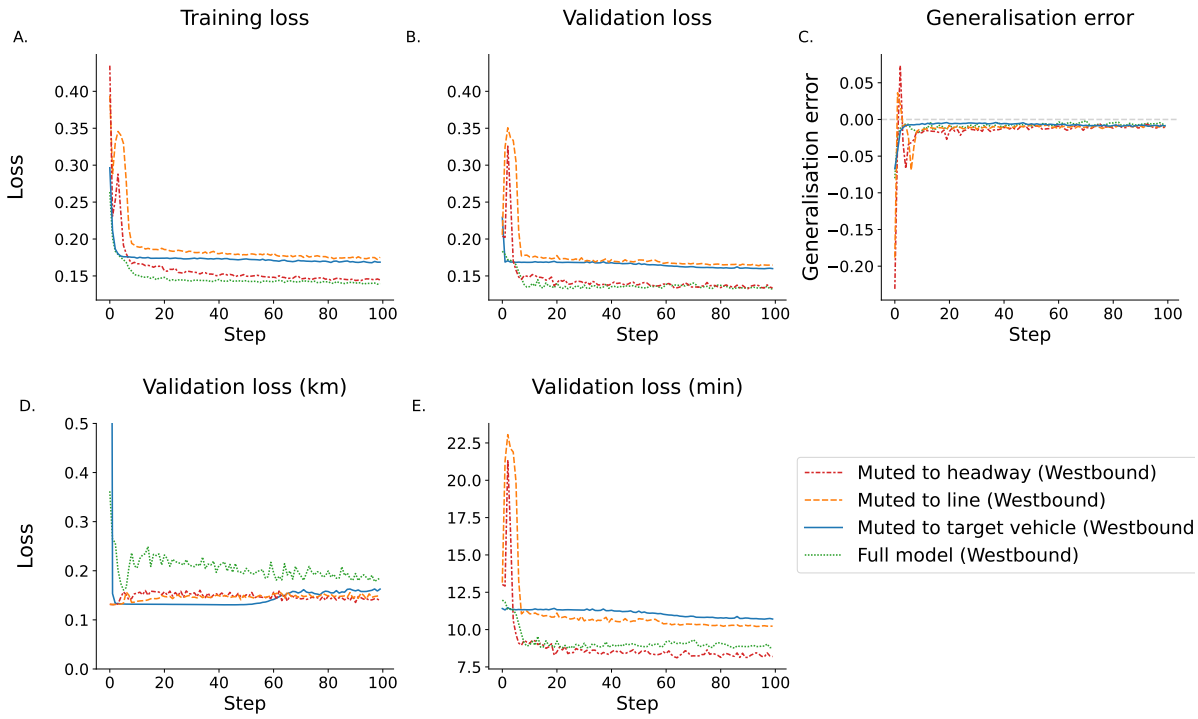


Figure 5.13: Training and validation of the **westbound direction** for all muted models in comparison to the model with access to the entire network data.In the example of this direction the best validation loss was achieved by the model incorporating the entire network data.**A.** shows the training loss. **B.** shows the validation loss. **C.** shows the generalisation error calculated by subtracting the training error from the testing error. **D.** shows the estimated validation error in km. **D.** shows the estimated validation loss in minutes. (Note that subfigure (**D**) has a truncated *y* axis for illustration purposes).

5.4.3 Effect of journey direction

A complication seen in many of our datasets shows that buses on the same line can have a significantly different route patterns. One cause may be, for example, one-way systems in city centres, which is the case in Reading (Figure 5.1) and will cause vehicles to travel on different geographical routes depending on the direction. Furthermore, the same line can operate on partial routes and omit certain stops on specific runs. This means in practice that a vehicle of the same line could sometimes stop or start its journey halfway along the route. This complicates any prediction and cannot be extracted from the available data, as vehicles cannot be matched with specific timetable entries. All investigated bus lines are affected by these conditions, which would be expected to introduce significant and complex noise into the dataset. This issue also makes the interpretation of model performance difficult, as each direction has different timetable patterns, which could affect the difficulty of the prediction task. Furthermore, the westbound direction runs less frequently and therefore produces fewer data points.

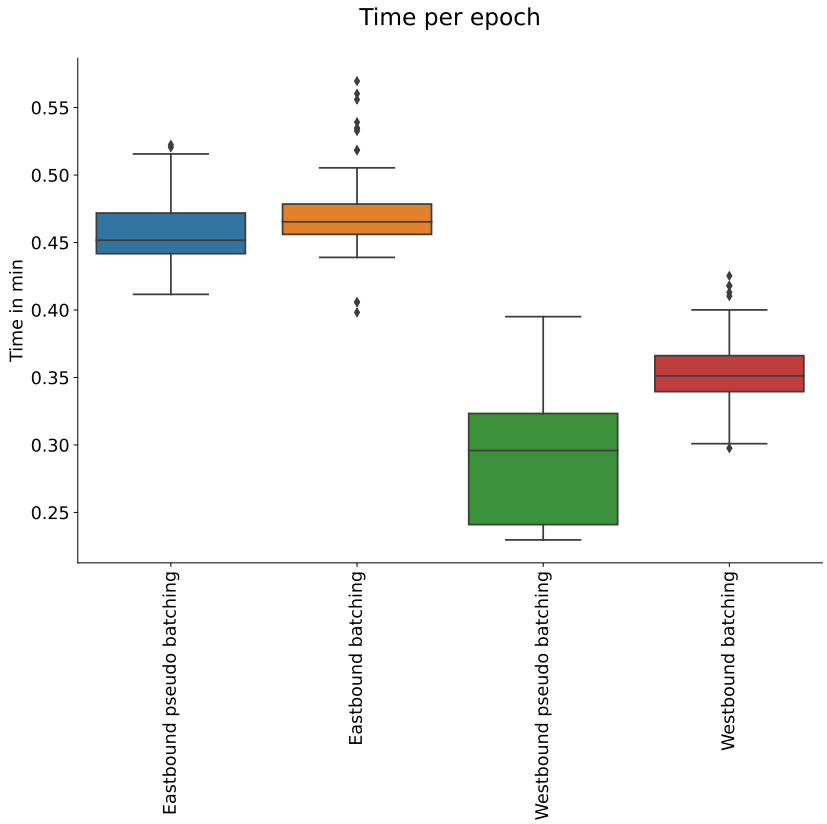


Figure 5.14: Figure shows boxplots of the time per epoch for both directions.

5.4.4 Performance on full data

As the results described above show that the model limited to vehicle headways on the same line performs the best in predicting trajectory positions, this version of the model was chosen and ran on a large dataset containing 56,679 and 12,187 training and testing samples for the westbound direction, respectively. In the eastbound direction, the dataset consisted of 107,831 and 22,953 training and testing samples, respectively. In both directions, a moderate improvement in per-epoch validation performance can be noticed. More importantly, a regularisation effect of the larger dataset and consequently a larger number of batches is evident. This can be seen in the less volatile validation curves. A similar reduction in volatility can be seen in the ETA validation performance, especially in the westbound direction shown in Figure 5.15 and 5.16. This can be explained by the smaller size of the dataset and, thus, the smaller number of batches within this dataset. From these results, it can be concluded that larger datasets have a two-fold benefit: in a reduction in validation volatility and thus generalisation error and, secondly, a moderate improvement of overall prediction performance. This highlights the need for very large datasets if data quality is poor. As a consequence, this results in very long training times. A possibility to further improve the performance of the proposed model structure is to include synthetic data during the training process. This has the caveat that it is unlikely that any network interaction can be incorporated into synthetic models, as these tend to be unknown factors.

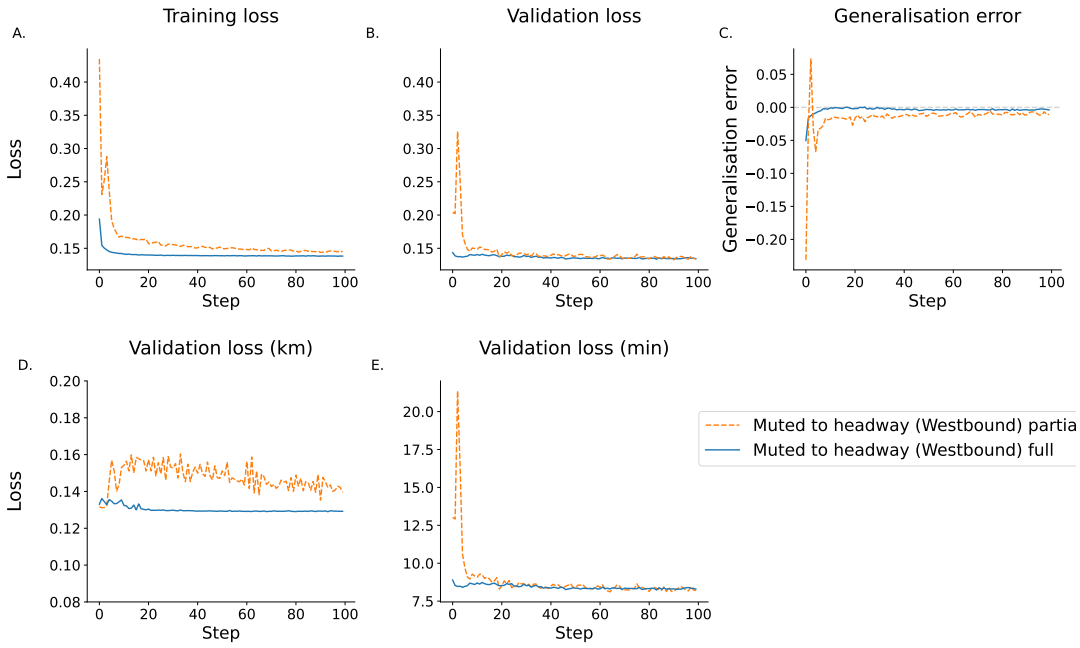


Figure 5.15: Comparison of the westbound headway muted model trained on a partial as well as a full dataset.**A.** shows the training loss. **B.** shows the validation loss. **C.** shows the generalisation error calculated by subtracting the training error from the testing error. **D.** shows the estimated validation error in km. **D.** shows the estimated validation loss in minutes. (Note that subfigure (**D**) has a truncated *y* axis for illustration purposes).

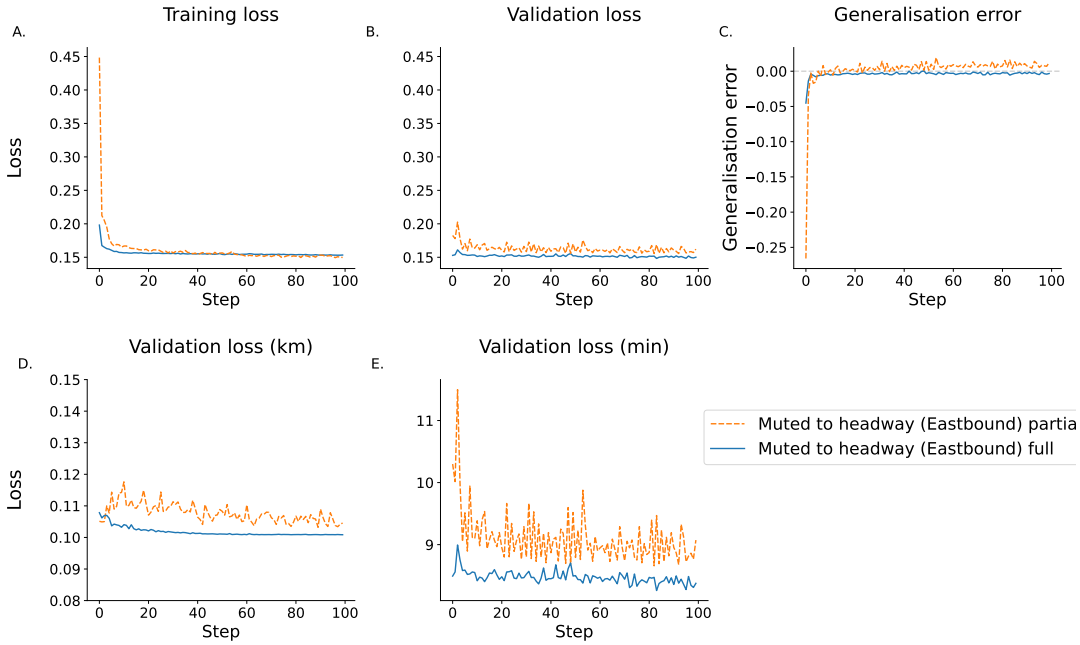


Figure 5.16: Comparison of the eastbound headway muted model trained on a partial as well as a full dataset. **A.** shows the training loss. **B.** shows the validation loss. **C.** shows the generalisation error calculated by subtracting the training error from the testing error. **D.** shows the estimated validation error in km. **D.** shows the estimated validation loss in minutes. (Note that subfigure (**D**) has a truncated *y* axis for illustration purposes).

5.4.5 Findings in context with previous results

Both datasets are of different sizes (westbound 12.478 journeys and eastbound 23.960 journeys). As an approximation, we assume a dataset of 12.500 journeys, while our previous paper used 17.115 journeys with an average error of approx. 120 m [110] and showed that this can be improved by incorporating synthetic data [157]. However, because the interacting lines only represent $\approx 5\%$, of the total data this is equivalent to ≈ 625 journeys for each of the interacting lines. Therefore, to make the dataset equivalent to our previous one for all interacting vehicles, this means the dataset has to be increased by a factor of 27 $(17.000/625 = 27)$.

Due to the processing time currently needed, this would become prohibitive. A dataset of ≈ 12.500 requires ≈ 30 min per epoch. If we assume a linear scaling of processing time (this is not fully true as the eastbound dataset does not scale linearly but the run time increases marginally slower than if it was scaling linearly), this then indicates a required epoch time for $30\ min * 27 = 13.5\ hrs$.

Superconvergence as used in our previous paper by employing the one-cycle policy speeds up the process by approx. 5 fold. This means that the 50 epochs used previously would be equivalent to approximately 250 epochs without a one-cycle policy [106]. As a result, a total processing time of 140 d can be expected, making such an experiment prohibitive.

It should be noted that these values are based on the number of data points in the area of interest as defined in Figure 5.2 and therefore represent only a small part of the traffic system. Another way to look at the problem would be to assess the percentages of data points in the final dataset. These show a different picture as each vehicle can be found in several samples for different target vehicles. Therefore, data points related to a target vehicle can be outnumbered by data points from other lines. This means concretely that more data for interacting lines can be present, thus it can be assumed that if more data from the target line was present, this could further improve the model accuracy.

5.5 Future work

This study highlighted that, in some cases, a network-wide attention-based prediction method can improve ETA, as well as next-step location prediction. It also showed some drawbacks of this method, such as the complexity of implementation due to the fact that a dynamically changing numbers of models have to be trained in parallel. This made the process computationally very inefficient and, thus, in the current state of research, not a viable alternative to conventional models. The complexity of the model training prevented an efficient way of batching training samples. The single most important continuation of this work would be the implementation of a batching method, which not only speeds up training, making the development of a prediction algorithm less computationally costly, but might also improve the generalisation due to a regularising effect. If this is implemented efficiently, this should also allow the use of GPUs to further improve processing times. Another limitation of this study is the definition of what constitutes an interaction between lines. Ideally, all vehicles within the network should be included to prevent suboptimal bias in the choice of lines to be included in the model. Due

to the highlighted computational cost, this is currently not feasible. Once the aforementioned training inefficiencies are addressed, this will become possible and could boost the prediction performance. Another issue related to the choice of interacting lines is the fact that most of these interact at the end of the journey of line number 17 as shown in Figure 5.4. This means that the interaction between the lines is short-lived and might not have a large effect on the prediction accuracy. This could be alleviated by either choosing a different target line other than line 17 or modifying the selection criteria and definition of interacting lines. Furthermore, hyperparameter tuning was conducted empirically due to processing cost and only on a small subset of the data. Ideally, a hyperparameter search should be conducted on the entire dataset if possible, using a gridsearch, Bayesian tuning or potentially a genetic approach. However, due to the extremely long processing time, this is currently prohibitive. Finally, as we have highlighted before, our dataset has serious data quality issues [110]. As these issues are caused by infrastructure problems, these could be easily addressed if all involved companies prioritised their data collection and processing methods and made these publicly available.

5.6 Conclusion

To summarise, we have demonstrated the potential benefit of a novel model architecture using network-wide data to make predictions in public transport networks. The method developed is based on a library of GRU models for each individual line within a network to embed the temporal information of the individual vehicles. The combination of this embedding method with a transformer model allows the presented method to expand its information range to other vehicles within a bus network. The results are especially promising if vehicles ahead of the target vehicle are included. As this is a pilot study and was designed to test whether the described architecture is a promising area of research, more work is required to compare this architecture to more conventional methods. We have also highlighted several areas where improvements to our presented method are required to make it a viable alternative to current methods. As such, it is a stepping stone for future research to improve public transport predictions if network operators provide high-quality datasets. Furthermore, the developed method could also be used to simulate specific traffic scenarios, such as delays of specific vehicles and the effect on the rest of the bus network. Therefore, the presented method could not only result in better predictions, but could be a valuable simulation tool for network operators.

Chapter 6

Discussion and conclusion of the thesis

6.1 Summary of contributions

This thesis investigated (i) in chapter 3, the impact of the data quality of the public transport dataset used in this thesis. Some serious data problems (ii) that had a significant impact on any prediction task were identified through an ablation study, in which the artefacts were removed sequentially and the resulting predictions were compared. It was found that the removal of artefacts can improve predictive performance on next step predictions. Furthermore (iii), it was demonstrated that rephrasing the prediction target into a smaller predictive space can have significant positive impact on next-step predictions.

It was found that simplifying the prediction task at hand improved performance, with the constrained trajectory being the best performing prediction space. Furthermore, it was shown that the naive average speed based predictor was performing very well, but if the volatility of the predictions was considered, the naive baseline was outperformed by the RNN based approaches. Thus, even in the presence of serious data quality issues, an improvement was made when employing timeseries-aware RNN prediction methods.

Due to the observed lack of data quality, in chapter 4 it was necessary to develop a method to synthesize data (iv) in an effort to simulate data and also allow the controlled introduction of specific artefacts. This allowed to test whether models trained on either purely synthetic data and hybrid data consisting of real-world data and synthetic data behaved different when used to make predictions on noisy real-world data. It was successfully demonstrated that the mixing of synthetic data and real-world data improved the prediction accuracy compared to models just trained on synthetic data. Thus, it was shown that in the absence of high quality real-world data, hybrid data can be used as a workaround to improve predictive models.

Finally in chapter 5 a novel method was developed (v) that used a model library to embed information gained from several vehicles operating within the bus network. To this end, models for each line were trained where all vehicles operating on this line shared model weights. This dynamic approach allowed to combine these embedded vehicle-based data by using a transformer mechanism, which expands the view of the predictor to a recent snapshot of the model. It was shown through experiments that this is a promising approach and the model did improve, particularly if data from vehicles ahead of the target vehicle were included.

6.2 Limitations

Because the data available was of low quality this does naturally have impact on the shown results. If high-quality data had been available the initial next-step predictions might have performed with better accuracy. As the developed data generator did rely on these data any data issues will be repeated by the data-generator. If the data had been reliable an over sampling method could also have been used, however as the data reliability was not good enough such an approach would have potentially increased the issue. The demonstrated data generator circumvents this issue by approximating the ground truth in the absence of reliable data.

Although the network based predictor was successfully demonstrated this suffered from scalability issues due to its computation cost. Therefore, it was not possible to train a model with the maximum number of journeys available. Some initially attempts to employ pseudo batching did not result in an increase in computation efficiency.

6.3 Future work

This computation cost of the network-based predictor is the most important area of future research from this chapter, the improvement of computation speed of network-based dynamic multi-model prediction approaches. This could be potentially achieved by improving the batching process, which could as a result allow the use of GPU parallel processing capabilities and result in a drastic reduction of computation cost. However this could result in a reduction of the resolution of the attention mechanism. The chosen attention-based method is an adaptation of the original transformer architecture. This aims to make the algorithm computationally less costly and allows for better parallelisation [154]. As mentioned, this paralelisation was not available in the presented approach thus presenting an area for future research.

The original dot product attention [161] and most similar implementations [162] use a form of RNN as their encoder. These are limited by the RNN in terms of paralelisation as the RNN structure explicitly requires the sequential training due to their sequence based structure. The further development of this technique into the transformer resulted in a method which does not rely on the squential structure of the data, allowing better parallisation compared to RNNs [154]. However, due to it's architecture, the complexity of transformers increases quadratically with sequence length thus making it a computationally heavy model to train. Some authors have addressed these issues in efforts to make transformers less computationally expensive [163, 164]. Others even suggest that learning attention weights from token-token (query-key) might not be that important to get similar results [165]. An approach dispensing with the need of RNNs could be developed, as in the proposed model the attention is applied to the vehicles, which would require a different treatment of the sequence data from the individual vehicles which currently use a RNN. It is feasible to use line-based transformers that are then fed into a top layer transformer but this would be an extremely memory heavy model to train. However, as the progress of a vehicle along a route is inherently sequential, a sequential model such as an RNN should be

ideally suited. On the other hand, an event early on in the route could have knock-on effects along the journey, thus making the transformer another sensible approach to be tested. As this will result in a very complex model, this would be a worthwhile continuation of this work once better data become available.

If data quality issues are addressed the demonstrated network-based ETA prediction method is promising for future research and could be a first step to make urban bus ETA prediction methods use a holistic network-based approach and thus improve overall ETA predictions.

Concluding it can be said that the overarching aim of the thesis to establish whether it is possible to improve on naive short-horizon prediction of vehicle locations has been achieved. This could in the future be used to bring customer vehicle location interfaces up to real time and compensate for any transmission delays. Furthermore, the thesis successfully demonstrates a method to generate synthetic data which can be used in the future by researchers and engineers who are faces with low quality data in the bus public transport sector. It also allows the introduction of specific types of artefacts. Additionally it was demonstrated that the synthetic data can be used to improve the generalisability of public transport algorithms. This is a crucial insight that could be used in future research to not only allow the development of predictive algorithms in cases where no high quality data is available but also to potentially boost the performance of algorithms trained on high quality data. Finally, a novel method of using the network state of an entire city was proposed. This builds a platform to conduct further work on which could ultimately lead to novel network aware predictive algorithms for the public transport network. To make this a reality high quality data needs to be made available. Furthermore, a focus on the training efficiency could lead to further developments in the technical aspects of training highly complex collaborative models.

Bibliography

[1] Grotenhuis, J. W., Wiegmans, B. W. & Rietveld, P. The desired quality of integrated multimodal travel information in public transport Customer needs for time and effort savings. *Transport Policy* **14**, 27–38 (2007).

[2] Salvador, M. M., Budka, M. & Quay, T. Automatic Transport Network Matching Using Deep Learning. *Transportation Research Procedia* **31**, 67–73 (2018). URL `https://linkinghub.elsevier.com/retrieve/pii/S2352146518301273`.

[3] Ye, Q., Szeto, W. Y. & Wong, S. C. Short-term traffic speed forecasting based on data recorded at irregular intervals. *IEEE Transactions on Intelligent Transportation Systems* **13**, 1727–1737 (2012).

[4] Department for Transport. TransXChange (2014). URL `https://www.gov.uk/government/collections/transxchange`.

[5] Department for Transport. National Public Transport Access Nodes (NaPTAN) (2015). URL `https://www.gov.uk/government/publications/national-public-transport-access-node-schema`.

[6] Department of Transport. Technical guidance: SIRI-VM (2020). URL `https://www.gov.uk/government/publications/technical-guidance-publishing-location-data-using-the-bus-open-data-service-siri-vm/technical-guidance-siri-vm`.

[7] Boisot, M. & Canals, A. Data, information and knowledge: Have we got it right? *Journal of Evolutionary Economics* **14**, 43–67 (2004).

[8] Kilkenny, M. F. & Robinson, K. M. Data quality: "Garbage in – garbage out". *Health Information Management Journal* **47**, 103–105 (2018).

[9] Sanders, H. & Saxe, J. Garbage in, garbage out (How purportedly great ML models can be screwed up by bad data). *Proceedings of Blackhat 2017* 6 (2017). URL `https://www.blackhat.com/docs/us-17/wednesday/us-17-Sanders-Garbage-In-Garbage-Out-How-Purportedly-Great-ML-Models-Can-Be-Screwed-Up-By-Bad-Data-wp.pdf`.

[10] Department of Transport. Road Use Statistics Great Britain 2016. *Statistical Release* **3**, 452–6 (2016). URL `https://www.licencebureau.co.uk/wp-content/uploads/road-use-statistics.pdf`.

[11] Cookson, G. & Pishue, B. INRIX Global Traffic Scorecard (2017). URL `https://media.bizj.us/view/img/10360454/inrix2016trafficscorecarden.pdf`.

[12] Xia, T. *et al.* Traffic-related air pollution and health co-benefits of alternative transport in Adelaide, South Australia. *Environment International* **74**, 281–290 (2015).

[13] Wang, P., Hunter, T., Bayen, A. M., Schechtner, K. & González, M. C. Understanding road usage patterns in urban areas. *Scientific Reports* **2** (2012).

[14] Transport Focus. Bus Passenger Survey Autum 2017 Report. Tech. Rep., Transport Focus (2017). URL `http://www.transportfocus.org.uk/research-publications/publications/bus-passenger-survey-full-report-autumn-2014/`.

[15] Mishalani, R. G., Mccord, M. M. & Wirtz, J. Passenger Wait Time Perceptions at Bus Stops : Empirical Results and Impact on Evaluating Real- Time Bus Arrival Information. *Journal of Public Tr* **9**, 89–106 (2006).

[16] Watkins, K. E., Ferris, B., Borning, A., Rutherford, G. S. & Layton, D. Where Is My Bus? Impact of mobile real-time information on the perceived and actual wait time of transit riders. *Transportation Research Part A: Policy and Practice* **45**, 839–848 (2011). URL http://dx.doi.org/10.1016/j.tra.2011.06.010.

[17] Xinghao, S., Jing, T., Guojun, C. & Qichong, S. Predicting Bus Real-time Travel Time Basing on both GPS and RFID Data. *Procedia - Social and Behavioral Sciences* **96**, 2287–2299 (2013). URL http://linkinghub.elsevier.com/retrieve/pii/S1877042813023847.

[18] Choudhary, R., Khamparia, A. & Gahier, A. K. Real time prediction of bus arrival time: A review. In *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, 25–29 (IEEE, 2016). URL http://ieeexplore.ieee.org/document/7877384/.

[19] Pekel, E. & Kara, S. S. A Comprehensive Review for Artificial Neural Network Application to Public Transportation. *Sigma Journal of Engineering and Natural Sciences* **35**, 157–179 (2017).

[20] Altinkaya, M. & Zontul, M. Urban Bus Arrival Time Prediction: A Review of Computational Models. *International Journal of Recent Technology and Engineering* **2**, 2277–3878 (2013).

[21] Bie, Y., Wang, D. & Qi, H. Prediction Model of Bus Arrival Time at Signalized Intersection Using GPS Data. *Journal of Transportation Engineering* **138**, 12–20 (2012). URL http://ascelibrary.org/doi/abs/10.1061/(ASCE)TE.1943-5436.0000318http://ascelibrary.org/doi/10.1061/%28ASCE%29TE.1943-5436.0000310.

[22] Raut, R. D. & Goyal, V. K. Public transport Bus Arrival Time Prediction with Seasonal and Special Emphasis on Weather Compensation changes using RNN. *IJARCCE* **1**, 378–382 (2012).

[23] Sun, Y., Yan, Q., Jiang, Y. & Zhu, X. F. Reliability prediction model of further bus service based on random forest. *Journal of Algorithms & Computational Technology* **11**, 327–335 (2017). URL http://journals.sagepub.com/doi/10.1177/1748301817725306.

[24] Čelan, M., Klemenčič, M., Mrgole, A. L. & Lep, M. Bus-stop Based Real Time Passenger Information System - Case Study Maribor. *IOP Conference Series: Materials Science and Engineering* **245** (2017).

[25] Panovski, D. & Zaharia, T. Long and Short-Term Bus Arrival Time Prediction with Traffic Density Matrix. *IEEE Access* **8**, 226267–226284 (2020).

[26] Panovski, D. & Zaharia, T. Real-time public transportation prediction with machine learning algorithms. *Digest of Technical Papers - IEEE International Conference on Consumer Electronics* **2020-Janua** (2020).

[27] Treethidtaphat, W., Pattara-Atikom, W. & Khaimook, S. Bus arrival time prediction at any distance of bus route using deep neural network model. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 988–992 (IEEE, 2017). URL http://ieeexplore.ieee.org/document/8317891/.

[28] Amita, J., Singh, J. S. & Pradeep Kumar, G. Prediction of Bus Travel Time Using Artificial Neural Network. *International Journal for Traffic and Transport Engineering* **5**, 410–424 (2015). URL http://dx.doi.org/10.7708/ijtte.2015.5.

[29] Bai, C., Peng, Z. R., Lu, Q. C. & Sun, J. Dynamic bus travel time prediction models on road with multiple bus routes. *Computational Intelligence and Neuroscience* **2015** (2015).

[30] Čelan, M. & Lep, M. Bus-arrival time prediction using bus network data model and time periods. *Future Generation Computer Systems* **110**, 364–371 (2020). URL https://doi.org/10.1016/j.future.2018.04.077.

[31] Chen, M., Yaw, J., Chien, S. I. & Liu, X. Using automatic passenger counter data in bus arrival time prediction. *Journal of Advanced Transportation* **41**, 267–283 (2007).

[32] Chen, M., Liu, X., Xia, J. & Chien, S. I. A dynamic bus-arrival time prediction model based on APC data. *Computer-Aided Civil and Infrastructure Engineering* **19**, 364–376 (2004).

[33] Chen, C.-h. An Arrival Time Prediction Method for Bus System. *IEEE Internet of Things Journal* **PP**, 1–1 (2018). URL https://ieeexplore.ieee.org/document/8425651/.

[34] Chien, S. I. J., Ding, Y. & Wei., C. Dynamic Bus Arrival Time Prediction with Artificial Neural Networks. *Journal of Transportation Engineering* **128**, 29–438 (2002).

[35] Dailey, D., Maclean, S., Cathey, F. & Wall, Z. Transit Vehicle Arrival Prediction: Algorithm and Large-Scale Implementation. *Transportation Research Record: Journal of the Transportation Research Board* **1771**, 46–51 (2001). URL http://trrjournalonline.trb.org/doi/10.3141/1771-06.

[36] Deng, L., He, Z. & Zhong, R. The Bus Travel Time Prediction Based on Bayesian Networks. *2013 International Conference on Information Technology and Applications* 282–285 (2013). URL http://ieeexplore.ieee.org/document/6709989/.

[37] Dong, J., Zou, L. & Zhang, Y. Mixed Model For Prediction OfBus Arrival Times. *2013 IEEE Congress on Evolutionary Computation* 2918–2923 (2013).

[38] Gal, A., Mandelbaum, A., Schnitzler, F., Senderovich, A. & Weidlich, M. Traveling time prediction in scheduled transportation with journey segments. *Information Systems* **64**, 266–280 (2017). URL http://dx.doi.org/10.1016/j.is.2015.12.001.

[39] He, P., Jiang, G., Lam, S. K. & Tang, D. Travel-Time Prediction of Bus Journey with Multiple Bus Trips. *IEEE Transactions on Intelligent Transportation Systems* **20**, 4192–4205 (2019).

[40] Heghedus, C. PhD Forum: Forecasting Public Transit Using Neural Network Models. *2017 IEEE International Conference on Smart Computing, SMARTCOMP 2017* (2017).

[41] Hua, X., Wang, W., Wang, Y. & Ren, M. Bus arrival time prediction using mixed multi-route arrival time data at previous stop. *Transport* **33**, 1–12 (2017). URL https://www.tandfonline.com/doi/full/10.3846/16484142.2017.1298055.

[42] Jalaney, J. & Ganesh, R. S. Accurate Bus Arrival Time from Linear and Non-Linear Route Parameters Using Hybrid Predictors. *Proceedings - 2nd International Conference on Smart Electronics and Communication, ICOSEC 2021* 633–638 (2021).

[43] Jeong, R. & Rilett, R. Bus arrival time prediction using artificial neural network model. *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749)* 988–993 (2004). URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1399041.

[44] Julio, N., Giesen, R. & Lizana, P. Real-time prediction of bus travel speeds using traffic shockwaves and machine learning algorithms. *Research in Transportation Economics* **59**, 250–257 (2016). URL http://dx.doi.org/10.1016/j.retrec.2016.07.019.

[45] Junyou, Z., Fanyu, W. & Shufeng, W. Application of Support Vector Machine in Bus Travel Time Prediction. *International Journal of Systems Engineering* **2**, 21–25 (2018). URL https://www.tandfonline.com/doi/full/10.1080/21680566.2017.1353449.

[46] Kee, C. Y., Wong, L. P., Khader, A. T. & Hassan, F. H. Multi-label classification of estimated time of arrival with ensemble neural networks in bus transportation network. *2017 2nd IEEE International Conference on Intelligent Transportation Engineering, ICITE 2017* 150–154 (2017).

[47] Khosravi, A., Mazloumi, E., Nahavandi, S., Creighton, D. & Van Lint, J. W. A genetic algorithm-based method for improving quality of travel time prediction intervals. *Transportation Research Part C: Emerging Technologies* **19**, 1364–1376 (2011). URL http://dx.doi.org/10.1016/j.trc.2011.04.002.

[48] Kumar, A., Kumar, V., Vanajakshi, L. & Subramanian, S. Performance Comparrison of Data Driven and Less Data Demanding Techniques for Bus Travvel Time on Prediction. *European Transport* **9** (2017).

[49] Li, Y., Huang, C. & Jiang, J. Research of bus arrival prediction model based on GPS and SVM. *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018* 575–579 (2018).

[50] Lin, W.-H. & Zeng, J. An Experimental Study on Real Time Bus Arrival Time Prediction With Gps Data. *Transportation Research Record: Journal of the Transportation Research Board* **1666**, 101–109 (1999). URL http://docshare01.docshare.tips/files/12862/128628627.pdf.

[51] Lin, Y., Yang, X., Zou, N. & Jia, L. Real-Time Bus Arrival Time Prediction: Case Study for Jinan, China. *Journal of Transportation Engineering* **139**, 1133–1140 (2013). URL http://ascelibrary.org/doi/10.1061/%28ASCE%29TE.1943-5436.0000589.

[52] Liu, Z. Z., Wang, Y. & Huang, P. Q. AnD: A many-objective evolutionary algorithm with angle-based selection and shift-based density estimation. *Information Sciences* **509**, 400–419 (2020). URL https://doi.org/10.1016/j.ins.2018.06.063.

[53] Maiti, S., Pal, A., Pal, A., Chattopadhyay, T. & Mukherjee, A. Historical Data based Real Time Prediction of Vehicle Arrival Time. *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)* 1837–1842 (2014). URL http://ieeexplore.ieee.org/document/6957960/.

[54] Meng, L., Li, P., Wang, J. & Zhou, Z. Research on the Prediction Algorithm of the Arrival Time of Campus Bus. In *Advances in Intelligent Systems Research (AISR),*, vol. 142, 31–33 (2017).

[55] Napiah, M. & Kamaruddin, I. Arima Models for Bus Travel Time Prediction. *Journal-The Institution of Engineers* **71**, 49 (2009). URL https://pdfs.semanticscholar.org/b2c1/6e5eb4efbafbd7f8ca3f93eb1a13f31f7677.pdf.

[56] Nadeeshan, S. & Perera, A. S. Multi-step bidirectional LSTM for low frequent bus travel time prediction. *MERCon 2021 - 7th International Multidisciplinary Moratuwa Engineering Research Conference, Proceedings* 462–467 (2021).

[57] Nimpanomprasert, T., Xie, L. & Kliewer, N. Comparing two hybrid neural network models to predict real-world bus travel time. *Transportation Research Procedia* **62**, 393–400 (2022). URL https://doi.org/10.1016/j.trpro.2022.02.049.

[58] Padmanaban, R. P. S., Vanajakshi, L. & Subramanian, S. C. Estimation of bus travel time incorporating dwell time for APTS applications. *IEEE Intelligent Vehicles Symposium, Proceedings* 955–959 (2009).

[59] Pan, J., Dai, X., Xu, X. & Li, Y. A Self-learning algorithm for predicting bus arrival time based on historical data model. In *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, 1112–1116 (IEEE, 2012). URL `http://ieeexplore.ieee.org/document/6664555/`.

[60] Petersen, N. C., Rodrigues, F. & Pereira, F. C. Multi-output Deep Learning for Bus Arrival Time Predictions. *Transportation Research Procedia* **41**, 138–145 (2019). URL `https://doi.org/10.1016/j.trpro.2019.09.025`.

[61] Shalaby, A. & Farhan, A. Bus Travel Time Prediction Model for Dynamic Operations Control and Passenger Information Systems. *Proceedings of the CD-ROM. Transportation Research Board 82nd Annual Meeting, National Research Council* (2003).

[62] Shalaby, A. & Farhan, A. Prediction model of bus arrival and departure times using AVL and APC data. *Journal of Public Transportation* **7**, 41–61 (2004). URL `http://www.nctr.usf.edu/wp-content/uploads/2010/03/JPT-7-1.pdf#page=46`.

[63] Sinn, M., Yoon, J. W., Calabrese, F. & Bouillet, E. Predicting arrival times of buses using real-time GPS measurements. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* **2**, 1227–1232 (2012). URL `http://ieeexplore.ieee.org/document/6338767/`.

[64] Taparia, A. & Brady, M. Bus journey and arrival time prediction based on archived AVL/GPS data using machine learning. *2021 7th International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2021* (2021).

[65] Vanajakshi, L., Subramanian, S. & Sivanandan, R. Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses. *IET Intelligent Transport Systems* **3**, 1 (2009). URL `http://digital-library.theiet.org/content/journals/10.1049/iet-its_20080013`.

[66] Wang, L., Zuo, Z. & Fu, J. Bus Arrival Time Prediction Using RBF Neural Networks Adjusted by Online Data. *Procedia - Social and Behavioral Sciences* **138**, 67–75 (2014). URL `http://linkinghub.elsevier.com/retrieve/pii/S1877042814041019`.

[67] Wu, X. *et al.* Bus Arrival Time Estimation Based on GPS Data by the Artificial Bee Colony Optimization BP Neural Network. *Proceedings - 2020 5th International Conference on Smart Grid and Electrical Automation, ICSGEA 2020* 264–267 (2020).

[68] Xu, H. & Ying, J. Bus arrival time prediction with real-time and historic data. *Cluster Computing* **20**, 3099–3106 (2017).

[69] Ye, L., Thiengburanathum, P. & Thiengburanathum, P. A Real-Time Bus Arrival Time Prediction System Based on Spark Framework and Machine Learning Approaches: A case study in Chiang Mai. *2021 Joint 6th International Conference on Digital Arts, Media and Technology with 4th ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering, ECTI DAMT and NCON 2021* 243–248 (2021).

[70] Yin, T., Zhong, G., Zhang, J., He, S. & Ran, B. A prediction model of bus arrival time at stops with multi-routes. *Transportation Research Procedia* **25**, 4627–4640 (2017). URL `http://dx.doi.org/10.1016/j.trpro.2017.05.381`.

[71] Yu, B., Yang, Z.-Z., Chen, K. & Yu, B. Hybrid model for prediction of bus arrival times at next station. *Journal of Advanced Transportation* **44**, 193–204 (2010). URL http://doi.wiley.com/10.1002/atr.136.

[72] Yu, B., Lam, W. H. K. & Tam, M. L. Bus arrival time prediction at bus stop with multiple routes. *Transportation Research Part C: Emerging Technologies* **19**, 1157–1170 (2011). URL http://dx.doi.org/10.1016/j.trc.2011.01.003.

[73] Yu, B., Wang, H., Shan, W. & Yao, B. Prediction of Bus Travel Time Using Random Forests Based on Near Neighbors. *Computer-Aided Civil and Infrastructure Engineering* **33**, 333–350 (2017). URL http://doi.wiley.com/10.1111/mice.12315.

[74] Zaki, M., Ashour, I., Zorkany, M. & Hesham, B. Online Bus Arrival Time Prediction Using Hybrid Neural Network and Kalman filter Techniques. *International Journal of Modern Engineering Research (IJMER)* **3**, 1–9 (2013). URL http://ijmer.com/papers/Vol3_Issue4/BC3420352041.pdfhttp://www.hindawi.com/journals/cin/2015/432389/.

[75] Zhang, Q., Zhang, Y. & Li, J. EasyComeEasyGo: Predicting Bus Arrival Time with Smart Phone. In *Proceedings - 2015 9th International Conference on Frontier of Computer Science and Technology, FCST 2015*, vol. 44, 268–273 (IEEE, 2015). URL http://doi.wiley.com/10.1002/atr.136http://ieeexplore.ieee.org/document/7314689/.

[76] Zeng, L. *et al.* A LSTM based bus arrival time prediction method. *Proceedings - 2019 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation, SmartWorld/UIC/ATC/SCALCOM/IOP/SCI 2019* 544–549 (2019).

[77] Cathey, F. W. & Dailey, D. J. A prescription for transit arrival/departure prediction using automatic vehicle location data. *Transportation Research Part C: Emerging Technologies* **11**, 241–264 (2003).

[78] Lipton, Z. C., Berkowitz, J. & Elkan, C. A Critical Review of Recurrent Neural Networks for Sequence Learning. *IEEE transactions on neural networks* **5**, 157–66 (2015). URL http://arxiv.org/abs/1506.00019http://www.ncbi.nlm.nih.gov/pubmed/18267787.

[79] Liu, H. & Schneider, M. Similarity measurement of moving object trajectories. *Proceedings of the Third ACM SIGSPATIAL International Workshop on GeoStreaming - IWGS '12* 19–22 (2012). URL http://dl.acm.org/citation.cfm?doid=2442968.2442971.

[80] Kumar, B. A., Jairam, R., Arkatkar, S. S. & Vanajakshi, L. Real time bus travel time prediction using k-NN classifier. *Transportation Letters* **7867**, 1–11 (2017). URL http://dx.doi.org/10.1080/19427867.2017.1366120.

[81] Xie, Z. Y., He, Y. R., Chen, C. C., Li, Q. Q. & Wu, C. C. Multistep Prediction of Bus Arrival Time with the Recurrent Neural Network. *Mathematical Problems in Engineering* **2021** (2021).

[82] Jinglin Li, Jie Gao, Yu Yang & Heran Wei. Bus arrival time prediction based on mixed model. *China Communications* **14**, 38–47 (2017). URL http://ieeexplore.ieee.org/document/7942193/.

[83] The Royal Society. *Machine learning: the power and promise of computers that learn by example*, vol. 66 (The Royal Society, 2017). URL https://royalsociety.org/~/media/policy/projects/machine-learning/publications/machine-learning-report.pdf.

[84] Luo, W. *et al.* Guidelines for Developing and Reporting Machine Learning Predictive Models in Biomedical Research: A Multidisciplinary View. *Journal of medical Internet research* **18**, e323 (2016). URL http://www.ncbi.nlm.nih.gov/pubmed/27986644http://www.ncbi.nlm.nih.gov/pubmed/27986644.

[85] Baker, M. Over half of psychology studies fail reproducibility test. *Nature* (2015). URL http://www.nature.com/doifinder/10.1038/nature.2015.18248http://www.nature.com/articles/nature.2015.18248.

[86] Yarkoni, T. & Westfall, J. Choosing Prediction Over Explanation in Psychology: Lessons From Machine Learning. *Perspectives on Psychological Science* **12**, 1100–1122 (2017).

[87] Baker, M. 1,500 scientists lift the lid on reproducibility. *Nature* **533**, 452–454 (2016). URL http://www.nature.com/doifinder/10.1038/533452a.

[88] Hutson, M. Artificial intelligence faces reproducibility crisis Unpublished code and sensitivity to training conditions make many claims hard to verify (2018). URL http://www.ncbi.nlm.nih.gov/pubmed/29449469.

[89] Schooler, J. W. Metascience could rescue the 'replication crisis'. *Nature* **515**, 9–9 (2014). URL http://www.nature.com/doifinder/10.1038/515009a.

[90] Li Deng. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine* **29**, 141–142 (2012).

[91] Xiao, H., Rasul, K. & Vollgraf, R. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv* 1–6 (2017). URL http://arxiv.org/abs/1708.07747.

[92] Cui, Z., Ke, R. & Wang, Y. Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction. *IEEE* 1–12 (2018). URL http://arxiv.org/abs/1801.02143.

[93] Liu, H. *et al.* Bus Arrival Time Prediction Based on LSTM and Spatial-Temporal Feature Vector. *IEEE Access* **8**, 11917–11929 (2020).

[94] Peek, G.-J. & van Hagen, M. Creating Synergy In and Around Stations: Three Strategies for Adding Value. *Transportation Research Record: Journal of the Transportation Research Board* **1793**, 1–6 (2002). URL http://trrjournalonline.trb.org/doi/10.3141/1793-01.

[95] Department of Transport & Drivers and Vehicle Licensing Agency. All vehicles (VEH01) (2018). URL https://www.gov.uk/government/statistical-data-sets/all-vehicles-veh01table-veh0101.

[96] Hickman, M. Bus Automatic Vehicle Location (AVL) Systems. In *Assessing the Benefits and Costs of ITS*, 59–88 (Kluwer Academic Publishers, Boston, 2006). URL http://link.springer.com/10.1007/1-4020-7874-9_5.

[97] Al Ghifari, N. T., Setijadi Prihatmanto, A., Wijaya, R. & Yusuf, R. Data Quality Measures and Data Cleaning for Pattern Analysis Angkot Transportation in Bandung City. *Proceeding - ICoSTA 2020: 2020 International Conference on Smart Technology and Applications: Empowering Industrial IoT by Implementing Green Technology for Sustainable Development* (2020).

[98] Li, Y. & Voege, T. Mobility as a Service (MaaS): Challenges of Implementation and Policy Required. *Journal of Transportation Technologies* **07**, 95–106 (2017). URL http://www.scirp.org/journal/doi.aspx?DOI=10.4236/jtts.2017.72007.

[99] Yin, H., Wong, S. C., Xu, J. & Wong, C. K. Urban traffic flow prediction using a fuzzy-neural approach. *Transportation Research Part C: Emerging Technologies* **10**, 85–98 (2002).

[100] Agard, B., Morency, C. & Trépanier, M. MINING PUBLIC TRANSPORT USER BEHAVIOUR FROM SMART CARD DATA. *IFAC Proceedings Volumes* **39**, 399–404 (2006). URL https://linkinghub.elsevier.com/retrieve/pii/S1474667015359310.

[101] Misra, A., Gooze, A., Watkins, K., Asad, M. & Le Dantec, C. Crowdsourcing and its application to transportation data collection and management. *Transportation Research Record* **2414**, 1–8 (2014).

[102] Wepulanon, P., Sumalee, A. & Lam, W. H. K. A real-time bus arrival time information system using crowdsourced smartphone data: a novel framework and simulation experiments. *Transportmetrica B* **6**, 34–53 (2018). URL https://doi.org/10.1080/21680566.2017.1353449.

[103] Van Der Maaten, L. J. P., Postma, E. O. & Van Den Herik, H. J. Dimensionality Reduction: A Comparative Review. *Journal of Machine Learning Research* **10**, 1–41 (2009).

[104] Bengio, Y., Courville, A. & Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**, 1798–1828 (2013).

[105] Zech, J. R. *et al.* Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLoS Medicine* **15**, 1–17 (2018).

[106] Smith, L. N. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. *arXiv* 1–21 (2018). URL http://arxiv.org/abs/1803.09820.

[107] Sharpe, W. F. The Sharpe Ratio. *The Journal of Portfolio Management* **21**, 49–58 (1994).

[108] Department for Transport. Transport Statistics Great Britain 2019 Moving Britain Ahead. Tech. Rep. (2019).

[109] Department for Business Energy & Industrial Strategy. Greenhouse gas reporting: conversion factors 2019 (2019). URL https://www.gov.uk/government/publications/greenhouse-gas-reporting-conversion-factors-2019.

[110] Reich, T., Budka, M. & Hulbert, D. Impact of Data Quality and Target Representation on Predictions for Urban Bus Networks. In *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020*, 2843–2852 (IEEE, 2020). URL https://ieeexplore.ieee.org/document/9308166/.

[111] Rasdorf, W. *et al.* Transportation Distance Measurement Data Quality. *Journal of Computing in Civil Engineering* **17** (2003).

[112] Robinson, S., Narayanan, B., Toh, N. & Pereira, F. Methods for pre-processing smartcard data to improve data quality. *Transportation Research Part C: Emerging Technologies* **49** (2014).

[113] Arbex, R. & Cunha, C. B. Estimating the influence of crowding and travel time variability on accessibility to jobs in a large public transport network using smart card big data. *Journal of Transport Geography* **85** (2020).

[114] Sutskever, I., Vinyals, O. & Le, Q. V. Sequence to Sequence Learning with Neural Networks 1–9 (2014). URL http://arxiv.org/abs/1409.3215.

[115] Yamashita, R., Nishio, M., Do, R. K. G. & Togashi, K. Convolutional neural networks: an overview and application in radiology (2018).

[116] Theophilus, O. *et al.* Truck scheduling optimization at a cold-chain cross-docking terminal with product perishability considerations. *Computers and Industrial Engineering* **156** (2021).

[117] Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M. & Tavakkoli-Moghaddam, R. Red deer algorithm (RDA): a new nature-inspired meta-heuristic. *Soft Computing* **24**, 14637–14665 (2020). URL https://doi.org/10.1007/s00500-020-04812-z.

[118] Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M. & Tavakkoli-Moghaddam, R. The Social Engineering Optimizer (SEO). *Engineering Applications of Artificial Intelligence* **72**, 267–293 (2018). URL https://doi.org/10.1016/j.engappai.2018.04.009.

[119] Islam, M. R., Ali, S. M., Fathollahi-Fard, A. M. & Kabir, G. A novel particle swarm optimization-based grey model for the prediction of warehouse performance. *Journal of Computational Design and Engineering* **8**, 705–727 (2021).

[120] Moosavi, J., Naeni, L. M., Fathollahi-Fard, A. M. & Fiore, U. Blockchain in supply chain management: a review, bibliometric, and network analysis. *Environmental Science and Pollution Research* (2021).

[121] Ghadami, N. *et al.* Implementation of solar energy in smart cities using an integration of artificial neural network, photovoltaic system and classical Delphi methods. *Sustainable Cities and Society* **74**, 103149 (2021). URL https://www.sciencedirect.com/science/article/abs/pii/S2210670721004315https://linkinghub.elsevier.com/retrieve/pii/S2210670721004315.

[122] Mohammadi, M. *et al.* A hybrid computational intelligence approach for bioremediation of amoxicillin based on fungus activities from soil resources and aflatoxin B1 controls. *Journal of Environmental Management* **299**, 113594 (2021). URL https://linkinghub.elsevier.com/retrieve/pii/S030147972101656X.

[123] Huttunen, J. M., Kärkkäinen, L. & Lindholm, H. Pulse transit time estimation of aortic pulse wave velocity and blood pressure using machine learning and simulated training data. *PLoS Computational Biology* **15**, e1007259 (2019). URL https://doi.org/10.1371/journal.pcbi.1007259.

[124] Withers, K. B., Moschetti, M. P. & Thompson, E. M. A Machine Learning Approach to Developing Ground Motion Models From Simulated Ground Motions. *Geophysical Research Letters* **47** (2020). URL https://agupubs.onlinelibrary.wiley.com/doi/epdf/10.1029/2019GL086690https://onlinelibrary.wiley.com/doi/abs/10.1029/2019GL086690.

[125] Sethuraman, G. *et al.* Effects of Bus Platooning in an Urban Environment. In *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, 974–980 (Institute of Electrical and Electronics Engineers Inc., 2019).

[126] Ding, Y., Chien, S. I.-J. & Zayas, N. A. Simulating bus operations with enhanced corridor simulator: Case study of New Jersey transit bus route 39. *Transportation Research Record* 104–111 (2000).

[127] Ristoski, P., De Vries, G. K. D. & Paulheim, H. A collection of benchmark datasets for systematic evaluations of machine learning on the semantic web. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9982 LNCS, 186–194 (Springer Verlag, 2016). URL https://link.springer.com/chapter/10.1007/978-3-319-46547-0_20.

[128] TomTom. Bournemouth traffic report | TomTom Traffic Index (2020). URL https://www.tomtom.com/en_gb/traffic-index/bournemouth-traffic/.

[129] Azzalini, A. & Capitanio, A. Statistical applications of the multivariate skew normal distribution. *Journal of the Royal Statistical Society. Series B: Statistical Methodology* **61**, 579–602 (1999).

[130] Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning Internal Representations by Error Propagation. *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence* 399–421 (2013).

[131] Cho, K. *et al.* Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, vol. 4, 1724–1734 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2014). URL http://arxiv.org/abs/1409.3215http://aclweb.org/anthology/D14-1179.

[132] Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **9**, 1735–1780 (1997).

[133] Hittmeir, M., Ekelhart, A. & Mayer, R. On the utility of synthetic data: An empirical evaluation on machine learning tasks. *PervasiveHealth: Pervasive Computing Technologies for Healthcare* (2019).

[134] Rankin, D. *et al.* Reliability of supervised machine learning using synthetic data in health care: Model to preserve privacy for data sharing. *JMIR Medical Informatics* **8** (2020).

[135] Bolón-Canedo, V., Sánchez-Maroño, N. & Alonso-Betanzos, A. A review of feature selection methods on synthetic data. *Knowledge and Information Systems* **34**, 483–519 (2013).

[136] Kuchin, Y. I., Mukhamediev, R. I. & Yakunin, K. O. One method of generating synthetic data to assess the upper limit of machine learning algorithms performance. *Cogent Engineering* **7** (2020). URL https://doi.org/10.1080/23311916.2020.1718821.

[137] Dai, D., Sakaridis, C., Hecker, S. & Van Gool, L. Curriculum Model Adaptation with Synthetic and Real Data for Semantic Foggy Scene Understanding. *International Journal of Computer Vision* **128**, 1182–1204 (2020).

[138] Kaur, H., Pannu, H. S. & Malhi, A. K. A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys* **52** (2019).

[139] Das, H. P. *et al.* Conditional Synthetic Data Generation for Robust Machine Learning Applications with Limited Pandemic Data **19** (2021). URL http://arxiv.org/abs/2109.06486.

[140] Kalle, R. K., Kumar, P., Mohan, S. & Sakata, M. Simulation-driven optimization of urban bus transport. *WIT Transactions on the Built Environment* **186**, 97–108 (2019).

[141] Moosavi, S. M. H., Ismail, A. & Yuen, C. W. Using simulation model as a tool for analyzing bus service reliability and implementing improvement strategies. *PLoS ONE* **15**, 1–26 (2020). URL http://dx.doi.org/10.1371/journal.pone.0232799.

[142] Pells, S. R. An approach to the simulation of bus passenger journey times for the journey to work. *Transportation Planning and Technology* **14**, 19–35 (1989). URL https://www.witpress.com/Secure/elibrary/papers/UT19/UT19009FU1.pdfhttp://www.tandfonline.com/doi/abs/10.1080/03081068908717411.

[143] von Rueden, L., Mayer, S., Sifa, R., Bauckhage, C. & Garcke, J. *Combining Machine Learning and Simulation to a Hybrid Modelling Approach: Current and Future Directions*, vol. 12080 LNCS (Springer International Publishing, 2020). URL http://dx.doi.org/10.1007/978-3-030-44584-3_43.

[144] Reich, T., Budka, M., Robbins, D. & Hulbert, D. Survey of ETA prediction methods in public transport networks. *arXiv* (2019).

[145] Zhou, T. *et al.* Evaluation of Urban Bus Service Reliability on Variable Time Horizons Using a Hybrid Deep Learning Method. *Reliability Engineering & System Safety* **217**, 108090 (2021). URL https://doi.org/10.1016/j.ress.2021.108090.

[146] Hans, E., Chiabaut, N., Leclercq, L. & Bertini, R. L. Real-time bus route state forecasting using particle filter and mesoscopic modeling. *Transportation Research Part C: Emerging Technologies* **61**, 121–140 (2015). URL http://dx.doi.org/10.1016/j.trc.2015.10.017.

[147] Coffey, C., Pozdnoukhov, A. & Calabrese, F. Time of arrival predictability horizons for public bus routes. *4th ACM SIGSPATIAL International Workshop on Computational Transportation Science 2011, CTS'11, in Conjunction with ACM SIGSPATIAL GIS 2011* 1–5 (2011).

[148] Varga, B., Tettamanti, T. & Kulcsár, B. Energy-aware predictive control for electrified bus networks. *Applied Energy* **252**, 113477 (2019). URL https://doi.org/10.1016/j.apenergy.2019.113477.

[149] Gössling, S., Kees, J. & Litman, T. The lifetime cost of driving a car. *Ecological Economics* **194**, 107335 (2022).

[150] Cats, O. & Jenelius, E. Beyond a complete failure: the impact of partial capacity degradation on public transport network vulnerability. *Transportmetrica B* **6**, 77–96 (2018). URL https://doi.org/10.1080/21680566.2016.1267596.

[151] Balster, A., Hansen, O., Friedrich, H. & Ludwig, A. An ETA Prediction Model for Intermodal Transport Networks Based on Machine Learning. *Business and Information Systems Engineering* **62**, 403–416 (2020).

[152] Poschmann, P. *et al.* Realization of ETA Predictions for Intermodal Logistics Networks Using Artificial Intelligence. *Lecture Notes in Logistics* **2**, 155–176 (2019).

[153] Paliwal, C. & Biyani, P. To each route its own ETA: A generative modeling framework for ETA prediction. *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019* 3076–3081 (2019).

[154] Vaswani, A. *et al.* Attention Is All You Need. *Visual Neuroscience* **31**, 153–163 (2017). URL http://arxiv.org/abs/1706.03762https://www.cambridge.org/core/product/identifier/S0952523813000308/type/journal_article.

[155] Tang, G., Müller, M., Rios, A. & Sennrich, R. Why self-attention? A targeted evaluation of neural machine translation architectures. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018* 4263–4272 (2020).

[156] Elbayad, M., Besacier, L. & Verbeek, J. Pervasive Attention: 2D Convolutional Neural Networks for Sequence-to-Sequence Prediction (2018). URL http://arxiv.org/abs/1808.03867.

[157] Reich, T., Budka, M. & Hulbert, D. Bus journey simulation to develop public transport predictive algorithms. *Soft Computing Letters* **3**, 100029 (2021). URL `https://doi.org/10.1016/j.socl.2021.100029`.

[158] Paszke, A. *et al.* PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems* **32** (2019).

[159] Kiefer, J. & Wolfowitz, J. Stochastic Estimation of the Maximum of a Regression Function. *The Annals of Mathematical Statistics* **23**, 462–466 (1952).

[160] Goodfellow, I., Bengio, Y. & Courvill, A. *Deep Learning (Adaptive Computation and Machine Learning series)* (MIT Press, 2016). URL `http://www.deeplearningbook.org`.

[161] Luong, M. T., Pham, H. & Manning, C. D. Effective approaches to attention-based neural machine translation. *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing* 1412–1421 (2015).

[162] Kim, Y., Denton, C., Hoang, L. & Rush, A. M. Structured attention networks. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings* 1–21 (2017).

[163] Gupta, A., Dar, G., Goodman, S., Ciprut, D. & Berant, J. Memory-efficient Transformers via Top-k Attention. *arXiv* 39–52 (2022).

[164] Kitaev, N., Kaiser, & Levskaya, A. Reformer: The Efficient Transformer. *arXiv* 1–12 (2020). URL `http://arxiv.org/abs/2001.04451`.

[165] Tay, Y. *et al.* Synthesizer: Rethinking Self-Attention in Transformer Models. *arXiv* (2020). URL `http://arxiv.org/abs/2005.00743`.

Appendix A

Further explanations

As further clarification to equations 4.4 and 4.5 here the same equations are shown in a different modification.

$$P = P_{i-1} + (t + [(v \times m) - d_{i-1} \pm \eta]) \tag{A.1}$$

Where: η = noise to be added, v =volatility, P=position, t =expected time at next position

Equation A.1 shows the case if a vehicle is deemed to be delayed at the next position. The delay ***volatility*** (v) is defined as the ratio of the reference curve standard deviation to the average reference curve itself multiplied by m. Where ***m*** is a modification factor applied if a vehicle is delayed which is sampled from a tailed random distribution. Thus *m* is the likelihood a vehicle makes up some time at the next bus stop. As this is multiplied by *v* it represents the data derived variation at the current location along the trajectory. Naturally the delay known from the previous location needs to be taken into account which is done by subtracting the known delay d_{i-1} form the possible time. Finally, random noise is generated to simulate GPS inaccuracies and applied using η. This derived progress along the route is added to the previous known location P_{i-1} thus simulating a realistic progress along the trajectory.

$$P = [P_{i-1} + t] - [p \times t] \tag{A.2}$$

Equation A.2 shows the scenario if a vehicle is most likely on time if no delay is added while allowing a time gain to either adjust the delay closer to the expected time or in some rare cases to arrive earlier than expected. If the bus is most likely on time or in other words no delay or time gain is expected, the probability p of it being on time is used to generate an adjustment towards the reference curve. This means a vehicle will not gain more delay or might make up some time if it has been previously delayed. The probability p doubles as a modification factor through its multiplication with the expected time at the next position t. By subtracting the estimated time gain from the expected time based on the last position P_{i-1} and the expected run-time t a time gain based on the data derived likelihood a vehicle is on time can be simulated.

Where: P=position, p = probability a vehicle is on time t =expected time at next position