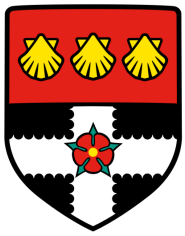


University of Reading

Department of Computer Science

Reading Bus Time Prediction - A Data Science Approach

Jade Ford

Supervisor: Dr. Varun Ojha

A report submitted in partial fulfilment of the requirements of
the University of Reading for the degree of
Bachelor of Science in *Computer Science*

May 11, 2020

Declaration

I, Jade Ford, of the Department of Computer Science, University of Reading, confirm that all the sentences, figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

Jade Ford
May 11, 2020

Abstract

Arrival time of public transport is a crucial point of information systems for passengers. This report investigates the analysis of Reading bus times and how factors of the day can influence the predictions of arrival and departure of a bus. The intention is to draw on the data from the Reading bus history. The report will be in the form of information related to the bus, such as the service number, type of bus, the stops it takes and names, the detailed time it arrives and the expected scheduled time for arrival. The arrival time and delay of the bus between designated stops is a crucial point of information for passengers when on public transport. The information helped to understand how the bus prediction would be studied by comparing the scheduled times to the actual arrival times that occurred on a given day. The data gathered is used to analyse further into each bus and bus stop to understand how the time of the day could affect any possible changes to the efficiency and operation of a bus. The purpose of this is the ability to provide an accurate prediction of a given bus using this information as a model and base from recurring history of the timetable. The result of the study should indicate and advocate that utilising data science we can predict bus times. This can be beneficial and lead to a more accurate and efficient method of showing the arrival times and possibly any delays of the bus before it even occurs, this then allows the information of this to go towards better services.

Keywords: Data Science, Cloud Computing, Data Processing, Data Visualization

Report's total word count: 19982

Acknowledgments

This project would not have been possible without the support of many people. Firstly, I would like to express my gratitude to Dr Varun Ojha for his unique and valuable guidance that has helped the completion of the project possible. Knowledge, recommendations and directions from him have made it possible for me to complete areas of data science tools and techniques that were new to me. I am also grateful to Martin Millmore from Oracle, who has helped me tremendously with his kind approach to instructing me how to use Oracle, encouragement and insight within machine learning. Finally, I would like to thank members in my household for the support, brightening up my day when I was struggling and keeping me motivated throughout the year.

Contents

List of Figures

List of Tables

Listings

List of Abbreviations

API	Application Programming Interface
ANN	Artificial Neural Network
APC	Automatic Passenger Counter
AVL	Automatic Vehicle Location
ARIMA	Autoregressive Integrated Moving Average
CORSIM	Corridor Simulation
GPS	Global Position System
LSTM	Long Short Term Memory
MSE	Mean Squared Error
PID	Project Initiation Document
RODS	Reading Buses Open Data Service
RNN	Recurrent Neural Network
SVM	Support Vector Machine

Chapter 1

Introduction

1.1 Background

The project entails the analysis and collecting of the Reading buses data to further use this information for predictions of the arrival times of the buses. It includes ingestion of live and historical data of the transport services to further the prediction of delays or arrivals. The improvement of predictions entailed the use of a data model required from using the language python with the intent to use machine learning techniques that help to develop this. The use of jupyter notebook is long-established when it comes to data science and machine learning as it includes many features that help to develop models of data sets. The model approach for Reading buses suggests unreliability; this can be perceived when looking at historical data. Using the bus as a mode of transport daily gives knowledge of knowing that the objectives of being on schedule with the timetable are not being fulfilled in some instances. Models need to be developed and improved when it comes to predictions of data that is being used every day and so frequently to have an efficient outcome in predictions and information that can not only be accurate but useful to the masses of users. The set of inputs and outputs of the data will be utilised to train the application to perform the identification of patterns in the data set for the bus times and to create training sets. This method in data science will need to be used to help create algorithms using Python. Another method that would be beneficial to the project is supervised and unsupervised learning. A feature that the program will use to display the given bus for the predicted time so a user can view it. The data used will need to be displayed as a timetable so that it can be easy to read and understand. Other than this, the application should be able to use the data to update the predictive times effectively when encountering changes in data, such as the weather or traffic.

Using data science is a field in the technology industry that is increasing in numbers where developers are beginning to learn the insights of this topic further and is expanding. Therefore, the interest is becoming fonder and allowing the choice of more sub-topics to branch out for a career in the future.

1.2 Aims and Objectives

1.2.1 Aims

There are expected aims that come with a project. Therefore it will need to have goals to achieve usability and be developed in the highest expectation and visualisation. The aims include:

- To apply data science tools and techniques to predict arrival and delays of Reading buses.
- To allow users to search a specific bus history and predictability of arrival and delays.

1.2.2 Objectives

The purpose of the study is to develop an improved method in predicting Reading bus times when it comes to departures and arrivals of various buses. The objectives of the project are:

- To explore and implement machine learning and data science tools over Reading Buses historical data combined with weather data.
- To be able to demonstrate the use of cloud computing while using it to its full capacity and utilising it so that it can receive sufficient data.
- To explore data visualisation tools and techniques to report busses arrivals and delays.
- For the output of the project to use the data to build a model of bus transport in Reading which will include an interface that displays the predicted timetables for the current day for any bus.

1.3 Problem Statement

As stated, the overall objective of the study was to analyse the Reading buses data and to develop a model to predict buses arrival times and delays. Therefore, the purpose of this study is to establish if we can predict the arrival times of the buses using historical data.

Mathematically, the historical data $[X, Y]$ is a set of the n pairs of inputs $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ and outputs $Y = [y_1, y_2, \ldots, y_n]$. For these training data, we want to minimise mean squared error (MSE) $\mathcal{L}(X, f(X, \mathbf{w}))$ induced on a model $f(X, \mathbf{w})$ as per:

$$\mathcal{L}(X, f(X, \mathbf{w})) = \sum_{i=0}^{n} (f(x_i, \mathbf{w}) - y_i)^2. \tag{1.1}$$

1.4 Solution Approach

The solution approach towards this project was best done broken down and identified from the research constructed and discussed in the Literature Review. Each approach and decision will be justified.

1.4.1 Literature Review

Doing research is a vital part of the study so that it can be possible to grasp what is expected. It is imperative to know other studies to investigate the researcher's practices of machine learning models for various projects. This can then help with implementing different methods in the project. The outcome of models and their performances with a different type of data is intriguing to learn about as the information could be vital in comprehending what is expected from the results and how it can collate to their studies.

1.4.2 Data Collection

Part of the solution approach consists of obtaining data for a data set. The data can be obtained from websites that allow public data to be utilised for machine learning—obtaining data consisted of gathering data on day by day basis to create a bus data set which include various buses that will be analysed and used for a prediction.

1.4.3 Data Analyse and Visualisation

Visualising the data is vital as this presents data to be more understandable to the human eye in a visual format. Without visualising data in the data set, it would be challenging to know what is required to be performed and understand what the data represents without looking through a heap of rows and columns. It would also make it challenging to know any insights into the data when it comes to correlations of variables, and patterns and trends that may persist. Data is, therefore, more valuable when it is visualised.

1.5 Summary of contributions and achievements

What has been accomplished within this project is the ability to develop a prediction of a given bus in Reading. The project has been completed by gathering data from an open data service, cleaning the data, visualising various bus services, and developing a prediction using a bus service that relates to the University of Reading. The results consisted of predicting the arrival times and delay of the number 21 bus via Chancellor's Way bus stop. The analysed outcomes in this report did result in a prediction that has much leeway for improvement when it comes to accuracy and being serviceable in the future.

1.6 Organisation of the Report

The organisation of the report entailed completing the literature review first. The reason for this was because the literature review is helpful in the writing process. After the literature review was completed, it was time to prepare for the information that will be included in each section. The sections include the methodology that consists of the process required to accomplish a prediction such as methods of collecting and cleaning data, visualisation techniques; the results that persisted from each bus; a discussion and analysis of the results of how the buses have performed; conclusions of the observations and findings; future work that could be implemented and the reflection on the project. Each section of the report was organised to be completed within two weeks, as this would give each section the right amount of attention to complete substantial material. It was challenging to follow because of other obligations within the course; this contained individual and group course work.

Chapter 2

Literature Review

Data Science involves various tools, algorithms and machine learning principles that are utilised to discover hidden patterns from data either in the past or the present data. It performs an analysis to discover the insights from raw data and uses the different machine learning algorithms to identify patterns or occurrences of certain events. Therefore, the use of data science is to create decisions and predict anticipated outcomes with the use of predictive analytic and machine learning. In data science, there are different approaches to making decisions and predictions; this includes, predictive casual analytic, prescriptive analytic and machine learning.

The amount of data produced and used is increasing substantially. The quality of bus travel arrival time is augmented by the accurate prediction of the travel time and delays that occur. Accurate arrival time information is essential in this world to attract more commuters and increase satisfaction and attractiveness. The data collection process is obtained from Reading city buses source that is open to use. From this, the parameters of the constraints are then optimised and then illustrated and observed in a format that humans could fully understand and comprehend. The data is partitioned to be trained and then tested. The analysis investigates the use of the variables hypothesised, in this case, the arrival time and delay of the bus to influence the accurate travel time prediction. A concept model would then be presented for the real-time prediction with it being competitive to the archival time.

Bus travel time prediction has been explored using GPS, ANN and AVL. New approaches for bus time travel prediction has mostly relied on average models and linear regression, but recent studies show that other models outperform these types of model.

2.1 Artificial Neural Network Models

Creating models for predicting bus arrival times that includes delays are executed by using an Artificial Neural Network (ANN). This method is popular when developing machine learning models and in this case, when predicting bus arrival times, as the ANN models can solve complex situations and the features and facilities to utilise with real-time data. Researchers like (Chien et al., 2002, Jeong and Rilett, 2004, Gurmu and Fan, 2014, Johar et al., 2016, Kumar et al., 2014), have demonstrated the use and development of this model in their studies.

From these studies, it establishes that Artificial Neural Network Models are effective in bus time prediction as it can solve complicated nonlinear and linear relationships and process complex relationships between predictors that occur in the data. ANNs is influenced by imitating and intending to simulate a similar level of behaviour and intelligence than humans when processing data. This model is composed of neurons that contain the functions nonlinear and linear.

In 2002 a model was developed using ANN; this was done by Chien et al. (2002). It consisted of evaluating and comparing the predicted and simulated arrival time at each bus stop while also using CORSIM, a traffic simulation software, to simulate the data, including the passenger demand and volume. The model provided the passengers in New Jersey with bus arrival information by using two ANNs that were trained from link-based and stop-based data. This allowed the information on the predicted arrival times. Chien et al. (2002) concluded that integrating the adaptive algorithm that adapts to the prediction error in real-time with the neural model enhanced the results and outperformed compared to ones without the adaptive algorithm integration. It was also discovered by Jeong and Rilett (2004) that ANN models surpassed not only the regression models but also the historical data models when it came to developing more accurate results from the prediction.

2.2 Regression Models

Regression models are a type of predictive modelling technique that consists of examining the linear relationship between a dependent variable and set of independent variables, where the dependent is the target, and the independent is the predictor. This technique is conventional for time series modelling, forecasting, and investigating relationships between variables. Unlike historical data-based model, the regression model can handle variations and unstable traffic conditions. The regression model measures the effects of the various factors, which is how the independent variable affects the dependent variable.

Jeong and Rilett (2004) proposed using multiple linear regression models, ANN model and historical model in their studies using various sets of inputs. Their regression model developed used various sets of linear regression that estimated the travel times using the current bus stop data to the target bus stop data, which was used as the input. The independent variables that were considered included the schedule adherence, distance, traffic congestion and arrival time at the specific bus stop. Patnaik et al. (2004) also established using regression models in their studies to calculate arrival time using different independent variables and Automatic Passenger Counters (APC) to collect the data that consisted of the number of stops, the number of passengers boarding and alighting, distance, dwell times and weather. The model developed could additionally be used to calculate the arrival time of the buses at various conditions and estimate the arrival time under different circumstances with various independent variables. Other than this, Patnaik et al. (2004) discovered that the independent variable, weather, was an unimportant element to include for input in their model.

2.3 Historical Data Driven Models

Historical models consist of the historical data which enables the ability to predict the future bus arrival times from the previous journey, whether this may be for a specific bus stop or a bus route. The model assumes the prediction from the data patterns not having any changes, because of this, it can have the negative effect of being inaccurate when it comes to real situations. Therefore, this method is only reliable when factors are unwavering, such as the traffic congestion being minimal; this would typically occur in a rural area. Not only did Jeong and Rilett (2004) use regression models and ANN in their studies but also included historical models in the development of using AVL to collect the data. The use of AVL data is to apply the model for real-time applications from the historical data. The inputted data used included the schedule, dwell times of the vehicles and traffic congestion. The historical model was based on the prediction of travel time between two bus stops by having the calculation of the average travel time between the two stop minus the average dwell time at the bus stops.

2.4 Kalman Filtering-Based Models

Kalman filtering algorithm is based on linear and modelled on Markov. Referencing the article by Shalaby and Farhan (2004), it states the model used APC and AVL to collect the data and develop the prediction model that was installed on buses to predict the arrival times using the Kalman filter algorithm. The researcher analysed data gathered from the Toronto public transport system. The performance of the model was tested on real-world data and data from a simulation model to the use of error indices. The use of the Kalman algorithm allowed the performance of this model to be more efficient when it comes to prediction accuracy than more simple models, as it showed better performance when it came to errors on a test set compared to neural, historical and regression models. (Chien et al., 2002, Shalaby and Farhan, 2004) use this model to provide estimates of the current state so that it can serve as the foundation for future predictions of values or improving estimates of the variables from past times. Kumar et al. (2014) model developed focused on using ANN and Kalman filtering. The prediction algorithm performed well when it came to the prediction accuracy compared to the prediction method using discretization, which is the process of transferring continuous models and variables. When it comes to the database not being large Kalman filtering-based model can be advantageous.

2.5 Long Short-Term Memory Models

Long Short-Term Memory models, also referred to as LSTM models are part of a recurrent neural network (RNN). This method of machine learning models is usually developed and pertinent for processing and modelling predictions that are based on time-series data. The model can identify any unknown data between events in the series. The model was developed to have the capability to manage vanishing and gradient problems that occurred when training an RNN model. This can hinder the learning of data sequences creating it to be insignificant, which then causes for there not to be any performance of real learning. This model was introduced by Hochreiter and Schmidhuber (1997) and was popularised by many people in the same area of work. (Agafonov and Yumaganov, 2019) has demonstrated the use and

development of the LSTM model in their studies. Agafonov and Yumaganov (2019) used data from Samara, Russia, to predict the public transport arrival time by considering the varied content information about the cities situation in transport that has affected the travel time negatively and positively. The researchers concluded that this proposed model was able to outperform most typical prediction algorithms by producing high-quality results. They demonstrated that it could be utilised for real-time predictions for a transport bus arrival time subject to being on a transportation network that of a large scale.

2.6 Critique of the review

There were conclusions of ANN and AVL being the most popular techniques that researchers have used to develop a prediction system. Using these two techniques for the model created prosperous models and enabled the achievement of the intentions that researchers thought out and intended. From Jeong and Rilett (2004) study illustrates that by using ANN and AVL, it creates and develops the most accurate method to track and predict bus times. In addition to this study, using regression models with different independent variables can conclude to an excellent prediction performance as regression models can disclose the value of independent variables. The historical section of Jeong and Rilett (2004) study suggests that the prediction compared to other methods is not as accurate and is limited. There were previous studies of using Kalman filtering in time travel prediction. The use of this method in studies demonstrated that it could have a substantial outcome of results in prediction as the models are feasible. However, this model can lack in performance compared to other models such as ANN. Other than these models, the use of regression was additionally researched. From this, the findings concluded that the other models outperformed regression. Even though this may be the case, this model can disclose the importance of each input when predicting arrival times, for example, Patnaik et al. (2004) observed that the weather variable was not vital as an input in the model to conclude to an accurate prediction. This applies to the study by using stated machine learning models to the same extent that has been developed by the researchers.

2.7 Summary

In this section, I have examined literature that aided the investigation, and that is related to the project with the findings of what methodology was beneficial for producing a prediction. It has been found that specific models outperformed others when it came to the accuracy of predictions; this aided in the choice of the model towards the project.

Chapter 3

Methodology

3.1 Problem Study and Understanding

Public transport by bus has always been a popular method of transportation with millions using this way to travel for getting from a to b, whether this may be for people to get to school or work. When individuals are travelling by public transport, they need to know the information that relates to the bus arriving at their stop. The reliability of the transport of a bus is vital to upkeep the traffic of passengers and to attract new passengers so that the bus company is consistent in making revenue and bettering their reputation. Experiencing the services of buses in Reading has come to the observation that some inconsistency persists to the reliability of the service. An example of this is the bus service not occurring on time based on the scheduled arrival stated on the timetable. As of this transit, service reliability is a vital aspect of public transport in the case of being on schedule.

Researchers have defined transit service reliability in various ways. According to literature from Turnquist and Blume (1980), Transit service reliability is defined as "the ability of the transit system to adhere to a schedule or maintain regular head-ways and consistent travel time." While Strathman et al. (1999) relates reliability to being dependent on delays and schedule adherence and Abkowitz et al. (1978) defines it as "the invariability of transit service attributes that affects the decisions of users and operators". From these definitions, it can give the interpretation that the problem of the reliability of the Reading buses can be affected by various factors within a route of a bus that contributes to the outcome of the bus time arrival. As of this, travel time consists of the waiting time and in-vehicle time. It concludes the relation of the waiting time, possibly being dependent on the travel system and the behaviour and actions of the passenger. With the behaviour of the passengers, passengers who look up the scheduled arrival time in advance may experience a lower expected waiting time than passengers who do not look up the scheduled arrival time at the bus stop. Other than this factor, the reliability is mostly down to delays. Delays can be unpredictable and predictable. Unpredictable delays harm the reliability of a bus. This could be precipitated by the inexperience of a driver, traffic congestion or technical problems of the bus. On the other hand, predictable delays still have a negative impact, but they are anticipated, this could consist of the time of day, such as peak hours.

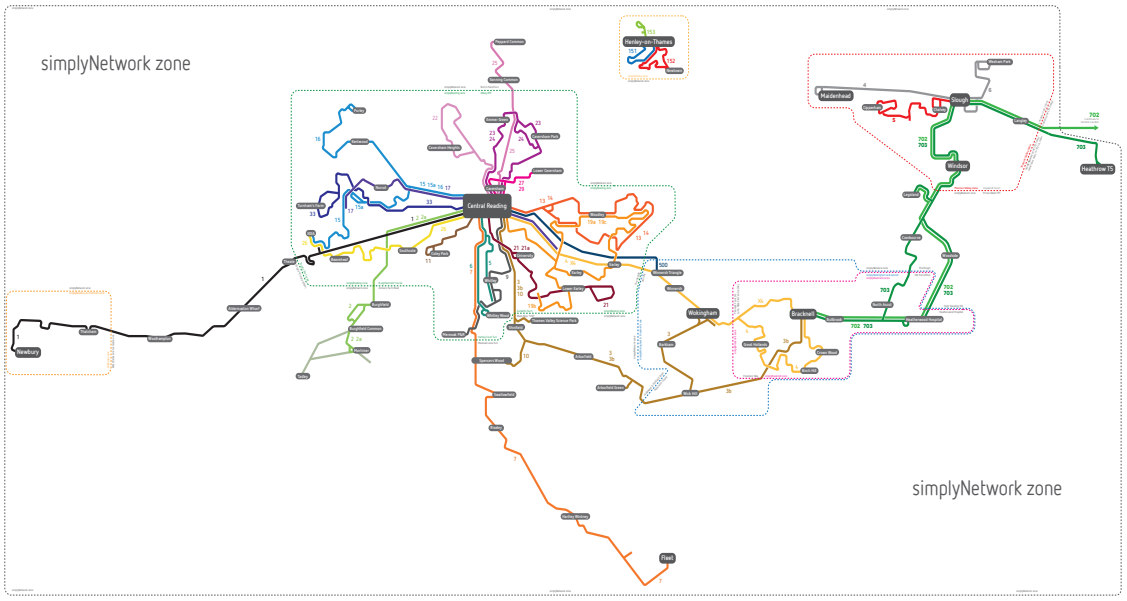


Figure 3.1: Reading Bus Network (ReadingBuses, 2020)

From the Fig. 3.1, it presents the bus routes ran by Reading Transport Limited and operates across Reading and servicing surrounding areas in the county Berkshire with the operation of 92 routes and a fleet that consists of 192 buses. The town routes provide a high frequency of buses every half an hour or hour depending on the time of day and route, with various routes now operating 24 hours a day, seven days a week. This reflects on the high-level demand for these routes around twilight times. These buses are the Claret 21 and 21a, Purple 17, Emerald 5, 6 and 6a, Orange 13, Yellow 26. On the other hand, the routes that do not operate in the centre run at a lower frequency, with a lower amount of buses an hour. The reason for the low frequency is because it operates as far as the surrounding towns, such as Bracknell, Newbury, Fleet and Wokingham. Reading buses are also in demand for Park and Ride. These operate to the south and east of the town.

Reading buses are fitted with trackers that are connected to the ticket machines. This would be by using Automatic Vehicle Location (AVL) so that there is information on the whereabouts of the bus and where it should be at a given time. From Council (2011) it states that the technology they deliver is an "innovative WiMAX communications network, enabling us to collect and disseminate information more effectively and demonstrate a range of transport using wireless technologies". The information is displayed to passengers via the electronic displays at the bus stops, the app and website. Using the AVL also provides the prediction of the duration it will take to travel onward the route based on the time scheduled on the journey. Reading buses operate on a sort of planner. As of this, it adds slack times when making a schedule. The buses are a tracking and schedule-based system that requires to set the departure schedules. Slack time relates to this because it is the difference between the scheduled time and the actual expected travel time. From this, if the slack time is inadequate, it will cause the buses to be impotent in catching up with the schedule. It will eventually fall behind, consequently declining the service quality. The problem with the service of Reading buses is that the bus is running late. Therefore, the prediction produced will not compeer with the scheduled arrival time, as it is stated on the Reading buses website, that the bus system does try to anticipate traffic but is unable to overcome this efficiently because of limitations.

3.2 Data Collection

Data can be adopted to improve an understanding by a human by converting it to information. Regarding this, the collection of real data from Reading buses system is used to fully understand how this data can be beneficial in predicting future arrivals and delays of the services provided, which could consequently improve travelling on transport for individuals.

Intending to understand the problem with Reading buses and to achieve the aims and objectives, it required the action of collecting data. This transpired by using the Reading Open Data Service (RODS) and application programming interface (API) to collect and gather the historical tracking data of a particular bus route. Receiving various information about the route included the route number, scheduled arrival and departure time, actual arrival and departure time, number of stops, location of where it operates and other information related to that route. The collection of this data was done between October 2019 and December 2019 to gather five months of data for six different routes that consisted of, routes 3, 5, 6, 10, 21, and 26.

```
1  import pandas as pd
2  import numpy as np
3  import requests
4  import json
5
6  url = "https://rtl2.ods-live.co.uk/api/trackingHistory?key=0o7wRdajVq&
       service=26&date=2018-05-31&vehicle=&location="
7  r = requests.get(url).json() # get data from url variable
8
9  data = json.dumps(r, indent=4, sort_keys=False)
10 print(data[:400]) # display first 400
11
12 df = pd.read_json(data)
13 df = df[['Site','Operator','LineRef','DepotCode','LocationCode','
       LocationName','ScheduledStartTime','LiveJourneyId','Sequence','
       RunningBoard','Duty','Direction','JourneyCode','VehicleCode','
       DriverCode','TimingPoint','JourneyPattern','ArrivalStatus','
       DepartureStatus','ScheduledArrivalTime','ScheduledDepartureTime','
       ArrivalTime','DepartureTime','ScheduledHeadway','ActualHeadway','
       JourneyId','ServiceGroup','NumberStops','StartPoint','EndPoint','
       Latitude','Longitude','District','JourneyType']]
14 columnsTitles = ['Site','Operator','LineRef','DepotCode','LocationCode','
       LocationName','ScheduledStartTime','LiveJourneyId','Sequence','
       RunningBoard','Duty','Direction','JourneyCode','VehicleCode','
       DriverCode','TimingPoint','JourneyPattern','ArrivalStatus','
       DepartureStatus','ScheduledArrivalTime','ScheduledDepartureTime','
       ArrivalTime','DepartureTime','ScheduledHeadway','ActualHeadway','
       JourneyId','ServiceGroup','NumberStops','StartPoint','EndPoint','
       Latitude','Longitude','District','JourneyType']
15
16 df = df.reindex(columns=columnsTitles)
17
18 df.to_csv('APIData.csv', mode='a', header=False, index=False) # add at
       next line of file
```

Listing 3.1: Code to extract data from Reading Buses Open Data Service using provided URL

To accomplish collecting the data, the method of using an API key was used. API's allows

the communication between websites and users, as it provides users with the ability to obtain data by requesting it from a server. Reading buses provided access to their data through this public API.

Using the API key from the RODS website generated the URL for the tracking history based on the service and date of the journey. This was then used in Python to get the information and store the request-response in an object to pass to the URL. This is done by using the GET request method, which requests data from the server. The content of the data requested is raw and is extracted in the data structure of JSON so that it is more readable structure and can be read to a data frame when inserted into the correct column headings. Each day collected was then inserted into the full data set of the buses line by line into a comma-separated value file. This is demonstrated in Listing 3.1.

3.3 Data Cleaning and Pre-processing Techniques

Data cleaning is a technique that is performed to convert raw data that has been collected to clean and convert it to feasible data; this process is a fundamental step. If this step were not performed, it would hinder the outcome of having a clean data set which could lead to more problems of the data not being feasible for analysis and visualisation. Therefore, it is vital to consider the methodology of this process.

From the collection of the bus data, it is utilised to its full extent. The data requires to be cleaned and prepared for processing. Various techniques can be approached to come to the outcome of a substantial data set. This is achievable by involving the process of identifying any missing data such as null and empty values and removing the values so that the data set does not include any inconsistency when being developed for the model. The result of this makes the data compatible with the objectives and aims that are in consideration. If a column has many rows that have missing values, then that itself can be removed. Dealing with missing values can also be determined by inputting said values based on other observations from the rest of the data, but this method can be sub-optimal because it can lead to loss of data. After all, the data inputted could not be substantial. The removal of duplication is performed. Duplicates are data that is repeated more than once in the data set; therefore, it would merely need to be removed to have a serviceable data set. Removing any unwanted observations consists of data that is irrelevant to the purpose, and that will not have any relation that fits under the content of the results therefore unimportant, as it also enables the ability to understand the related data clearer.

An example of this is keeping the column 'Sequence', which would not relate or bring any contribution to analysing the bus arrivals and develop a prediction. The process of this benefits the results as there would not be any errors when it comes to the prediction and will facilitate the results to be more accurate. Another technique includes fixing any structural errors. This includes typos, words that have inconsistent capitalisation and data types. For strings it is best to ensure that all the values are to be the same standard format, whether this may be upper, lower case or sentence case as this makes it simplistic and more manageable to read and shape. It is vital to ensure that the columns have the correct data type. For example, a column that stores numerical values require the data type of an integer or column. With the purpose to display dates, it requires the values to be stored as a date object or timestamp.

These techniques contribute to understanding the data and the capability to represent data to the standard of being visualised for the project.

Dataset Description

As stated, before the data used in this project came from the RODS website. The data is from January 1st to May 31st, 2018, containing 2,895,859 pieces of data, where it was presented in a tubular form.

Field	Description	Format	Unit
LineRef	Bus service	String	
LocationCode	code of location	Integer	
LocationName	Bus stop name	String	
ScheduledArrivalTime	Time of arrival	DateTime	'Y'-'M'-'D' 'h':'m':'s'
ScheduledDepatureTime	Time of departure	DateTime	'Y'-'M'-'D' 'h':'m':'s'
ArrivalTime	Actual arrival time	DateTime	'Y'-'M'-'D' 'h':'m':'s'
DepartureTime	Actual departure time	DateTime	'Y'-'M'-'D' 'h':'m':'s'
ServiceGroup	Group the bus service belongs to	String	
NumberStops	Amount of stops on route	Integer	
StartPoint	Start of route	String	
EndPoint	End of route to	String	
Latitude	Latitude	Float	
Longitude	Longitude	Float	

Table 3.1: Data description

The meaning of each field within the data set is shown in Table. 3.1. A given sample of the data for bus service 3 is demonstrated in Table. 3.2

Field	Data
LineRef	3
LocationCode	39025920003
LocationName	Kings Road
ScheduledArrivalTime	2018-01-02 05:53:00
ScheduledDepatureTime	2018-01-02 05:53:00
ArrivalTime	2018-01-02 05:52:49
DepartureTime	2018-01-02 05:53:17
ServiceGroup	Leopard
NumberStops	56
StartPoint	Station Road Stop SA
...	...

Table 3.2: Data sample of bus service 3

3.4 Modelling and Prediction Algorithms

Various modelling and prediction algorithms can be applied within data science and machine learning for different cases. Deciding what predictive modelling to apply to the problem is the key to getting the best possible results. The predictive models applied to the project include linear regression and LSTM.

3.4.1 Prediction based on Linear Regression

A liner regression performs a predictive analysis in which an independent variable (also called predictors or inputs) predict a dependent variable [also called output/target] (Seber and Lee, 2012, Ojha et al., 2012). The purpose of linear regression is to construct models that describe and demonstrate the relationships that exist between variables, where this may be independent or dependent. This can be done in a simple case, for example, two variable such as population and time. It can consist of a scatter graph and attempt to meet a curve that is smooth to the point where the curve is as close as possible to the points. It is expected that the curve would not exactly meet all the points as some variables would be subject to wear and be displayed as an anomaly. When there is the use of a straight-line using regression, it is indicated that the model is already fitted.

Linear regression states that there are relationships between variables, knowing this means that the relationship can be summarised. This can be represented in a formula: $y = a + \beta x$. To represent a more complex method for finding correlations, linear regression can also be represented as displayed in Eq. 3.1. The equation illustrates y_i as the predicted value, β_0 is the bias term, $\beta_1, ..., \beta_n$ are represented as the parameters of the model and $x_i, x_i, ..., \epsilon_i$ represented as the feature values. Linear regression aims to identify any relationships between y_i and the variables $x_i, x_i, ..., \epsilon_i$. This contributes to being utilised in the development of a prediction about y_i. The results would be based on the observations of the variables. Within this we need to train the model, this means discovering the best fit to the data and to then assess how well the model fits and to determine if it is truly the best fit or if it needs to be an improvement. A part of this process for this prediction consists of deciding which variables to include in the model so that there is the ability to establish a good predictor. To determine a good fit for the data linear regression calculates the distance between the fitted line and the data points. The model fits the data well from the difference between the observed values and the predicted values when the values are unbiased and small.

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i \tag{3.1}$$

3.4.2 Prediction based on Long Short-Term Memory

(Agafonov and Yumaganov, 2019) proposed long-short-term memory network, which has been addressed in the literature review. LSTM is a type of RNN, which is networks that allow information to persist within loops. LTSMs are designed to avoid long term dependency problems by remembering information or data for long periods (Olah, 2015). Within RNN it has a chain of repeating modules of a neural network and can have the structure of a single tanh layer. In contrast, the LSTM has a similar chain structure but is different from

the repeating modules being structured differently. LSTM is structured by having four neural network layers instead of one.

The four neural network layers are shown in Fig. 3.2. From the figure, arrows are used to illustrate the vector transfer. The vector transfer is from an output of a node to an input of another. The pink circles are to represent operations, such as vector addition, while the yellow boxes represent the learnt neural network layers. When the lines are merging, it indicates a sequence, while if the line is likewise to a fork, it indicates that the content is duplicated with transfers to different locations.

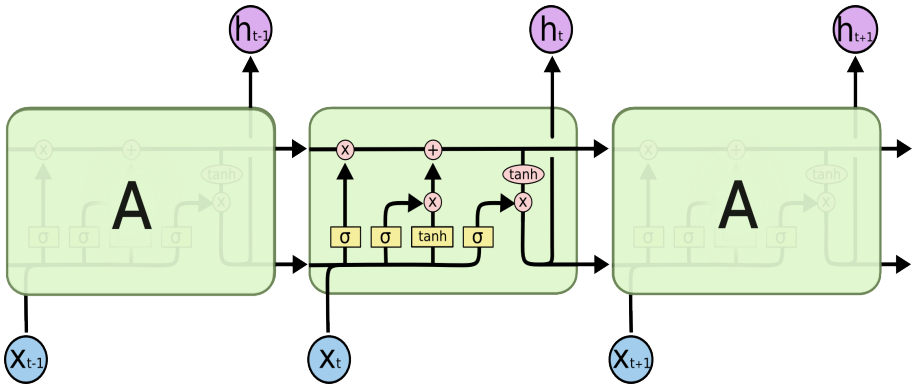
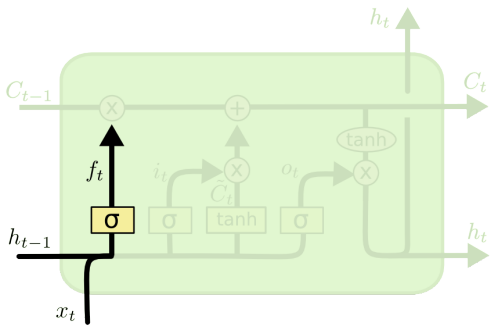


Figure 3.2: Long-short-term memory network example (Olah, 2015)

Referring to Fig. 3.2, the horizontal line that runs through the top of the Fig. 3.2 is the cell state. The cell state runs down the chain with information that the model allows to either be removed or added; this is managed by structures called gates. Gates have the purpose of deciding whether the information is authorised to go through. The sigmoid layer creates this. The sigmoid layer outputs binary values, 1 and 0, to describe the component amount that should pass through the gate. To let everything go through is represented by the value of 1 and to let noting go though is represented by the value of 0. This is done by looking at h_{t-1} and x_t in Fig. 3.3. For the model to predict the arrival of a bus, the cell state would include the timestamp of the present subject. Then when it sees a new subject, it would be best for it to forget the timestamp of the old subject.

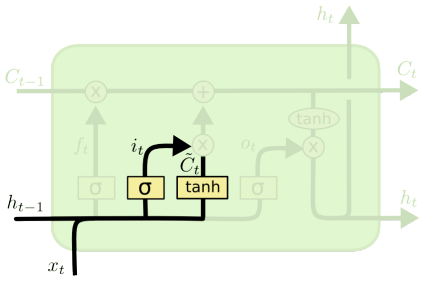


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Figure 3.3: Long-short-term memory network example (Olah, 2015)

LSTM models have three gates to control the cell state. Looking at Fig. 3.3, the first decision made by the sigmoid layer is whether to keep or remove the information for the first cell state C_{t-1} and then to decide which values will be updated. Referring back to the tanh layer that is displayed in Fig. 3.4, it would create a vector value, $\widetilde{C}_t$, that can be added to the cell state.

These are then combined to create an update to the state. The timestamp would be added as a new subject to the cell state so that it replaces the old one and forgets it.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 3.4: Long-short-term memory network example (Olah, 2015)

The next process of the model is to update the old cell state into the new cell state. This can be represented as the old cell C_{t-1} and $\widetilde{C}_t$ as the new cell state. From the equation, it uses f_t to multiply the old cell state to then add $i_t * \widetilde{C}_t$. The result of this is used as the new value and distinct by how much is updated to each state value. This can be represented by the equation displayed in Eq. 3.2.

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{3.2}$$

The end of this process consists of deciding what the output would be. This is based on the cell state. Referring to the first figure, Fig. 3.2, the sigmoid layer decides what part of the cell state to output. The cell state put through tanh so that the values get pushed between -1 and 1 and then multiplied by the output of the gate. This allows the ability to output the parts that are decided. In this instance of prediction, the output information would be a new timestamp of the bus on the next consecutive days.

3.5 Visualisation Techniques

In the field of Data Science and Machine Learning, visualisation is vital. The visualisation allows the representation of data that provides humans with a more innate understanding of what it exhibits and discovers, such as patterns that might arise. This facilitates the further process of creating a well-round analysis and insight from grasping a lot more of the data. This process is vital before developing and proceeding on to the next stage, as it is essential to understand the data in a qualitative manner and have knowledge of where the data falls short. The overview of the data and its attributes can include the distribution, presence of outliers, clustering of data and if it is linearly separable. Using the attributes as an analysis can assist the ability to conclude if a model is suitable for the data. Various languages can be utilised to perform visualisation of data. The ones that are universally used in Data Science is the languages Python, and R. Visualisation in Python is very much demonstrated in this report by using the utilities of the packages that are occupied with the language, this includes Matplotlib, Scikit and Seaborn.

Data can be envisioned with various plots with the combined use of Matplotlib for plots and Seaborn for making the plots appealing. Types of plots consist of line, histogram, box, pair, distribution, scatter, and bar chart. The architecture of Matplotlib consists of the back-end

layer, artist layer and scripting layer. The back-end layer is the bottom layer, which contains the functions required for plotting. The artist layer is the middle layer and has the purpose of plotting the functions on the figure. The scripting layer, which is the top layer, is where most of the code is done. The Fig. 3.5 purpose is to illustrate the artist layer hierarchy and how a plot can be developed. Fig. 3.5 contains axes that are added to the plots. The hierarchy allows more functionality and appearance when it comes to plotting the data.

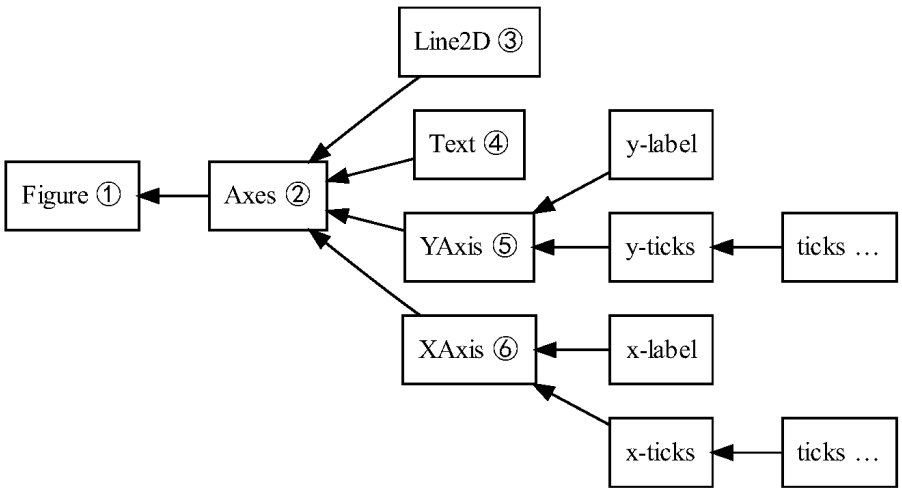


Figure 3.5: The hierarchy of Matplotib (Brown and Wilson, 2012)

Matplotlib and Seaborn libraries need to be imported before they can be used in the Python language. An example of this is presented in the below listing. Various features come with Matplotlib that allows representing data in a more detailed way or making the plot more comfortable to read. A plot could contain the attributes of a grid, axes, legends, title, x and y-axis label or can be displayed within a subplot.

```
1 import matplotlib.pyplot as plt
2 import seaborn as sns
```

Listing 3.2: Code to extract data from Reading Buses Open Data Service using provided URL

3.6 Implementation Details and Experiment Setup

For experiments, it is best to know what the contribution will be, what could go wrong and how to face it if the possibility occurs, what will be the evaluation and is it the desired output. Setting up an experiment can include the steps of planning, designing, and analysing. The step of planning includes a reasonable hypothesis and knowing the considerations that would be addressed, for example, "What are the independent and dependent variables?" and "What is the problem that needs to be solved?". Design is knowing what kind of plots would be the most suitable for representing the data. It would be best to know what the various plots is most suitable for, possibly from making observations from different research. Analysing is discovering any variability, which can be considered by replication.

The implementation utilises the machine learning methodologies that are detailed in the sub-

sections 3.4.1 and 3.4.2. The code that was used for programming the models are presented in Appx. A and Appx. B. Appx. A presents the code that was utilised to build the linear regression model and Appx. B presents the code that was utilised to build the LSTM model.

3.7 Summary

The methodology that is practised in the field of Data Science can be a long process when wanting to achieve outstanding results for the problem being addressed. The process of collecting substantial data from a dependable source could take some time if the data requires to be collected manually for a future data set and then cleaned. Within the cleaning process, there are various techniques to use to assure that the removal of corrupt data is performed and that the data can be used to its full potential for analysis and prediction. Other than this, the algorithms that can be adopted for Machine Learning varies. However, the widely used algorithm is linear regression as it is simplistic to implement with data and can produce exquisite results. The visualisation aspect of this process can be addressed using different techniques to represent data, whether this maybe by using a line plot or a bar chart. Visualisation is vital when it comes to data science as it drastically helps when required to understand and display data for humans to grasp.

Chapter 4

Results

The results help to allow the understanding of patterns that were identified and foreseen through analysing the data when it was presented visually. The Reading buses that were focused on and was popular and widely used by university students that studied at the University of Reading, this contained:

- Bus number 3
- Bus number 5
- Bus number 6
- Bus number 10
- Bus number 21
- Bus number 26

All the graphs displayed are for five months in 2018, (January, February, March, April, and May) and includes the insight of one bus stop related to the bus service.

4.1 Data Exploration and Visualisation

4.1.1 Bus Number 3 Arrival Times

The following displays the graphs created for the bus service number 3. The bus stop that is visualised to look more into detail is the Cressingham Road Church stop.

Figure 4.1: The average delay of bus 3 by months and overall months from January to May

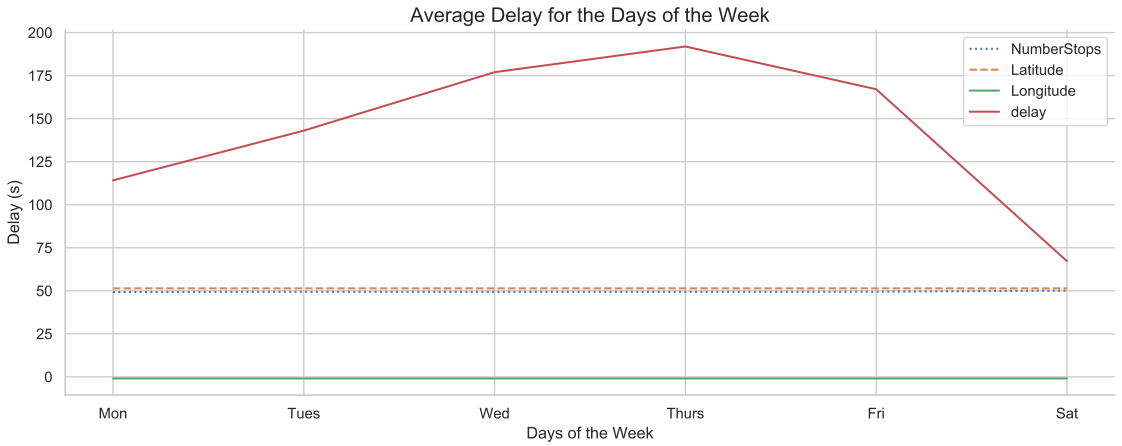


Figure 4.2: Average delay of bus 3 by the days of the week over the course of 5 months

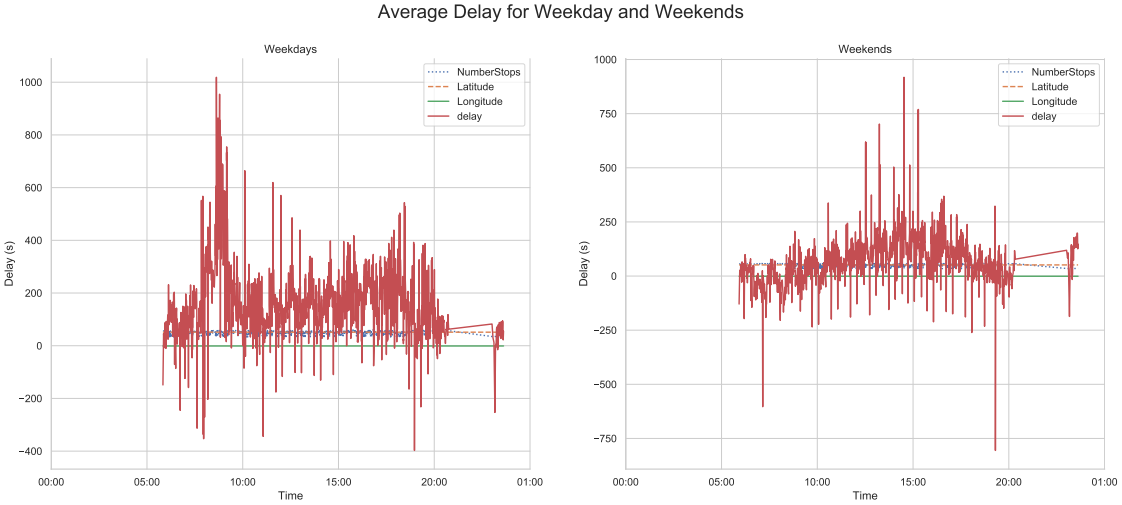


Figure 4.3: Average delay of bus 3 comparing the weekdays to the weekend

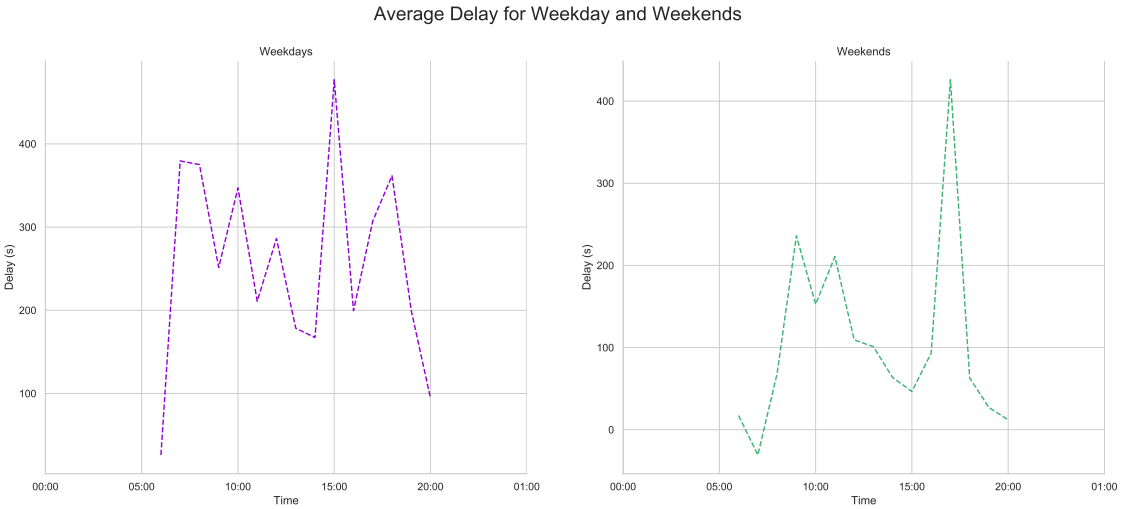


Figure 4.4: Average delay of bus 3 comparing the weekday to the weekend of the bus stop, Cressingham Road Church

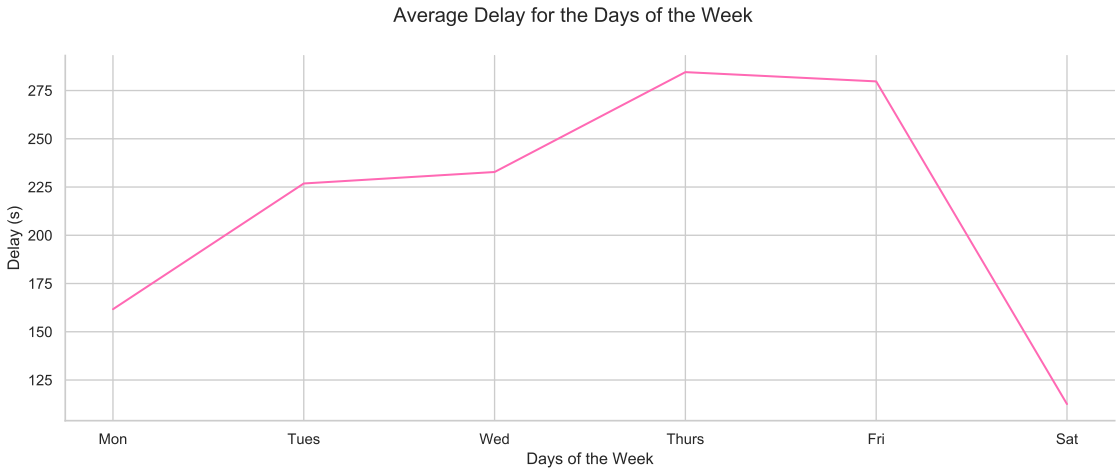


Figure 4.5: Average delay of bus 3 by the days of the week of the bus stop, Cressingham Road Church

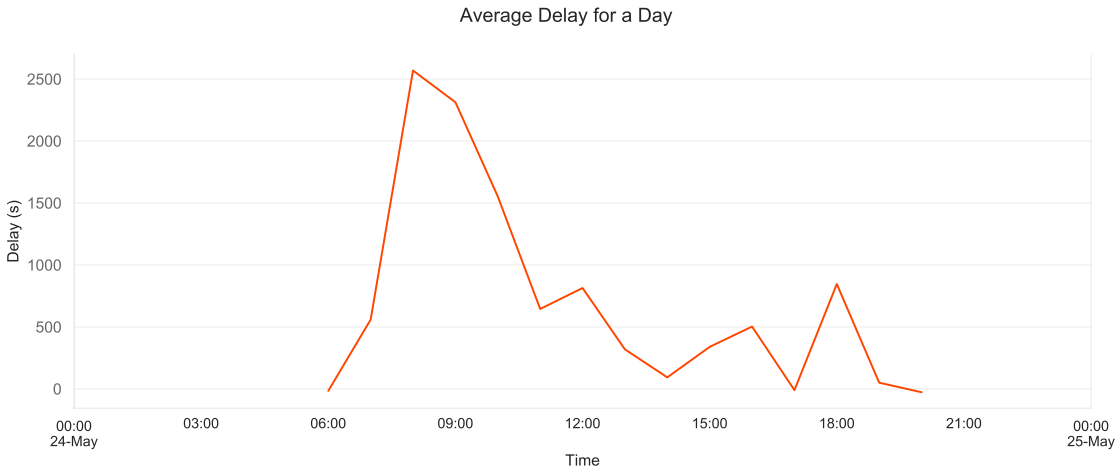


Figure 4.6: Average delay of bus 3 by a day of the week shown in 24 hours of the bus stop, Cressingham Road Church

4.1.2 Bus Number 5 Arrival Times

The following displays the graphs created for the bus service number 5. The bus stop that is visualised to look more into detail is the Whitley Street stop.

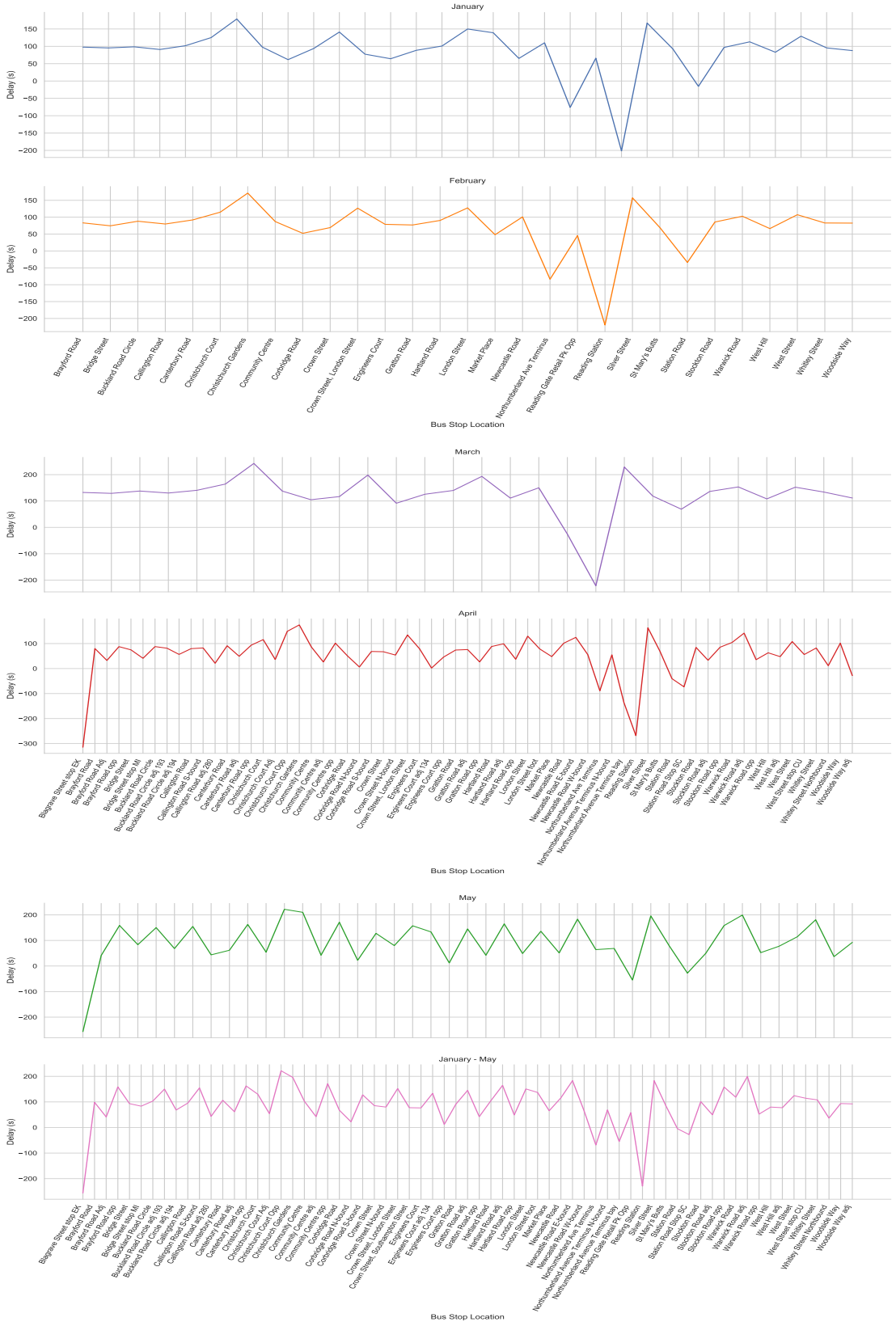


Figure 4.7: The average delay of bus 5 by months and overall months from January to May

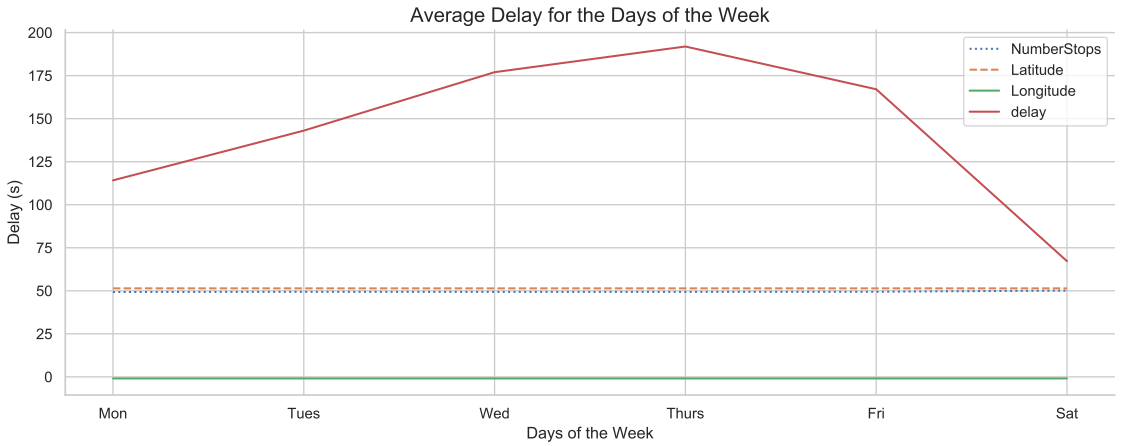


Figure 4.8: Average delay of bus 5 by the days of the week over the course of 5 months

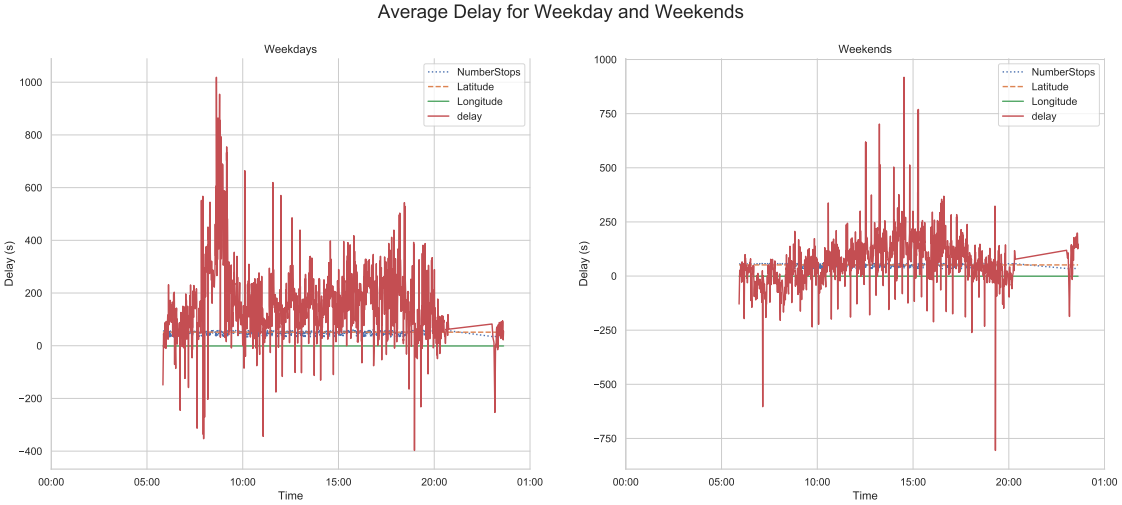


Figure 4.9: Average delay of bus 5 comparing the weekdays to the weekend

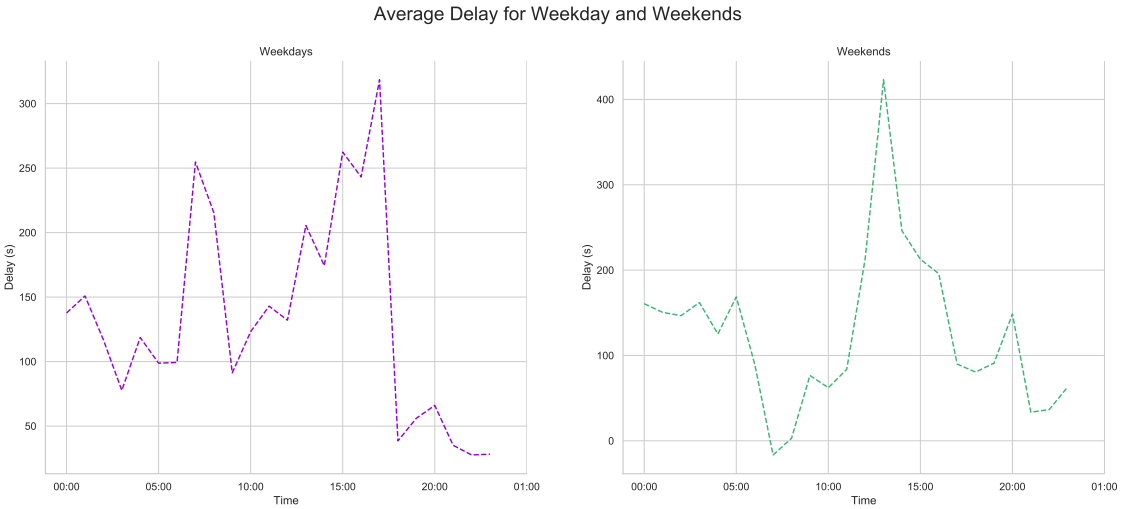


Figure 4.10: Average delay of bus 5 comparing the weekday to the weekend of the bus stop, Whitley Street

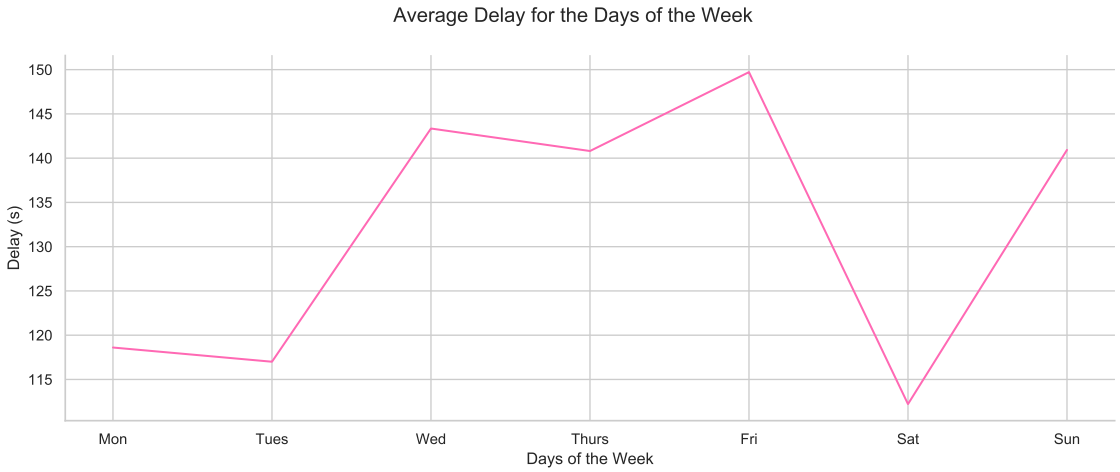


Figure 4.11: Average delay of bus 5 by the days of the week of the bus stop, Whitley Street

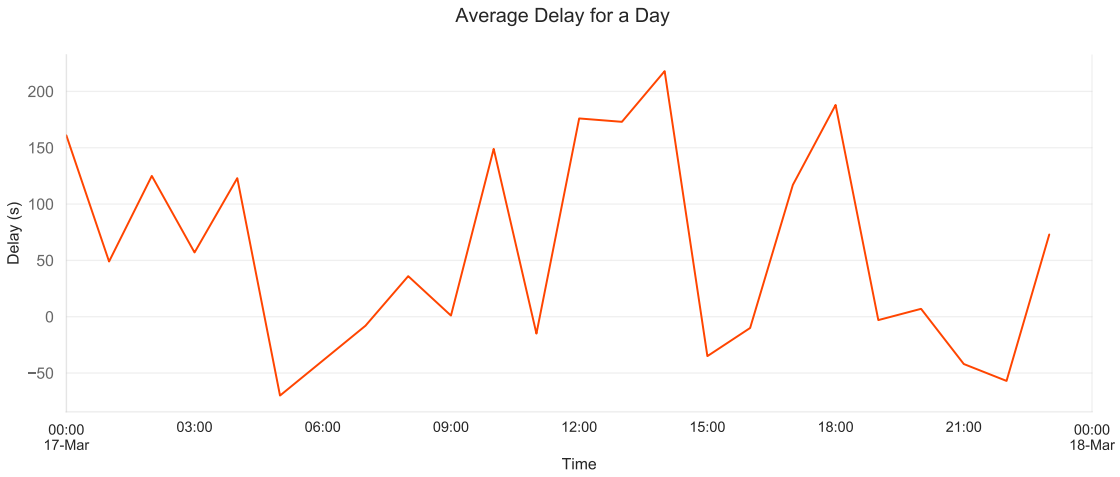


Figure 4.12: Average delay of bus 5 by a day of the week shown in 24 hours of the bus stop, Whitley Street

4.1.3 Bus Number 6 Arrival Times

The following displays the graphs created for the bus service number 6. The bus stop that is visualised to look more into detail is the Reading Station stop.

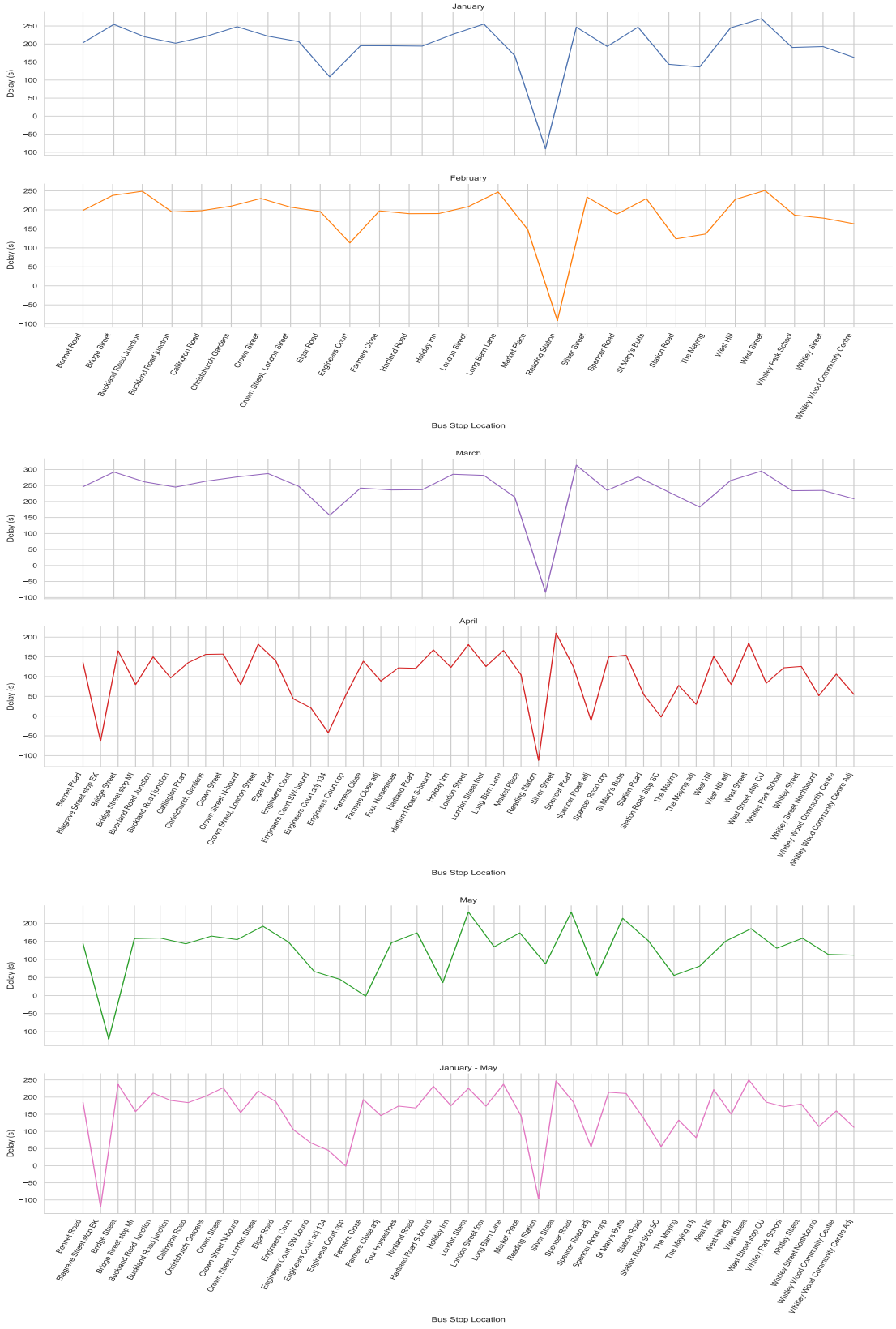


Figure 4.13: The average delay of bus 6 by months and overall months from January to May

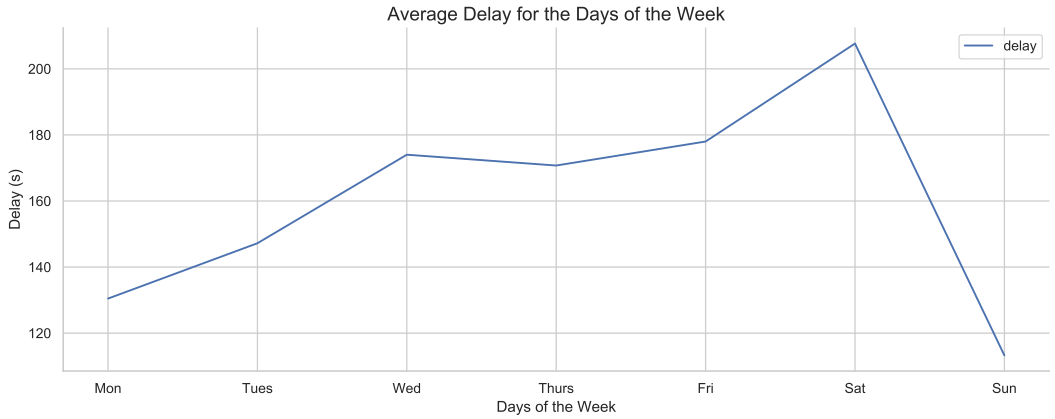


Figure 4.14: Average delay of bus 6 by the days of the week over the course of 5 months

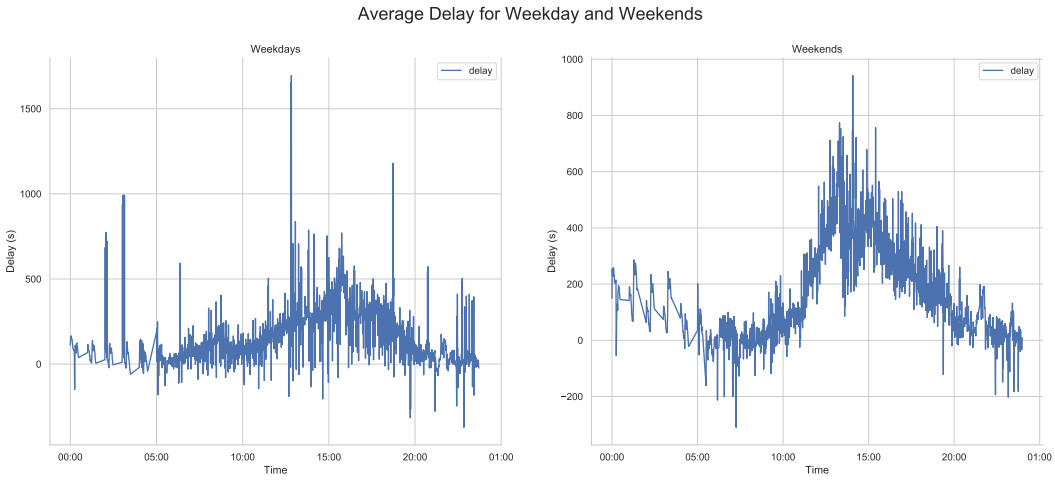


Figure 4.15: Average delay of bus 6 comparing the weekdays to the weekend

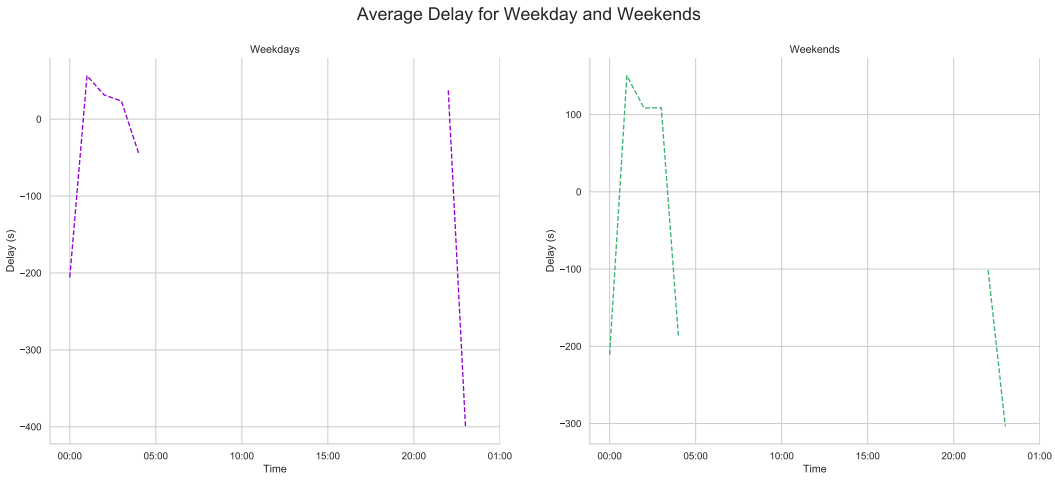


Figure 4.16: Average delay of bus 6 comparing the weekday to the weekend of the bus stop, Reading Station

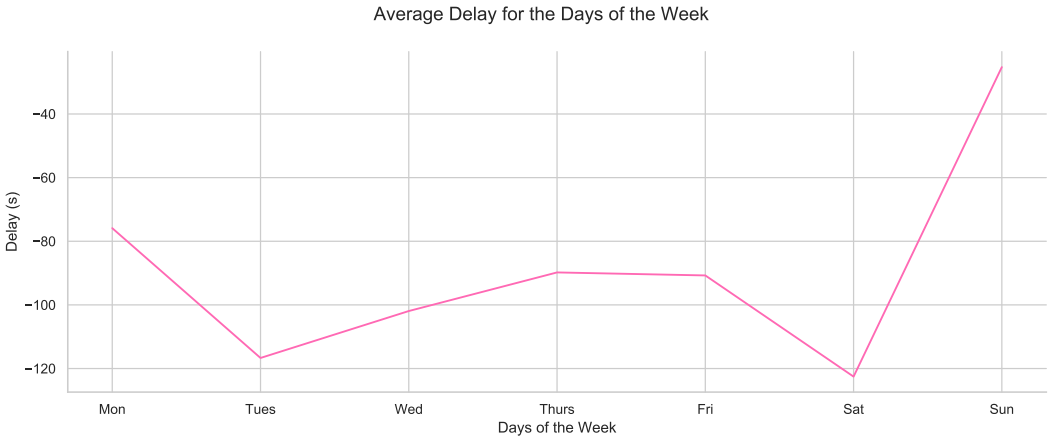


Figure 4.17: Average delay of bus 6 by the days of the week of the bus stop, Reading Station

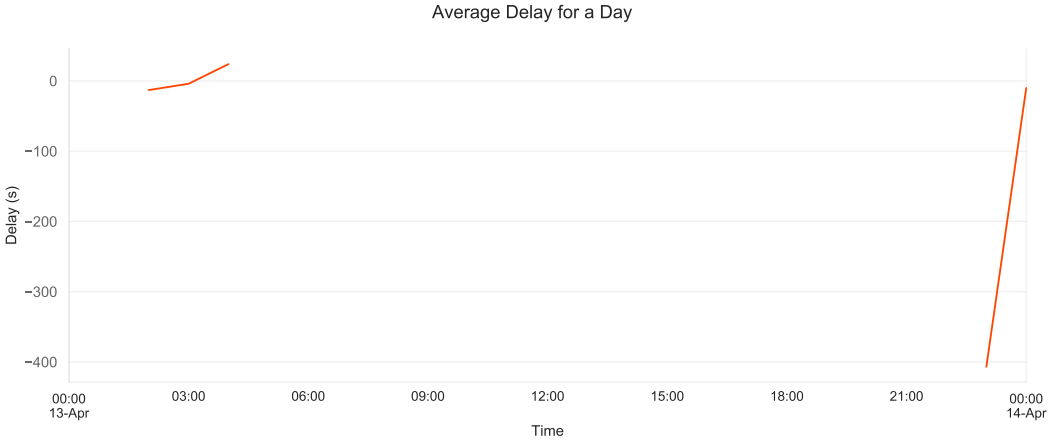


Figure 4.18: Average delay of bus 6 by a day of the week shown in 24 hours of the bus stop, Reading Station

4.1.4 Bus Number 10 Arrival Times

The following displays the graphs created for the bus service number 10. The bus stop that is visualised to look more into detail is the St Mary's Butts stop.

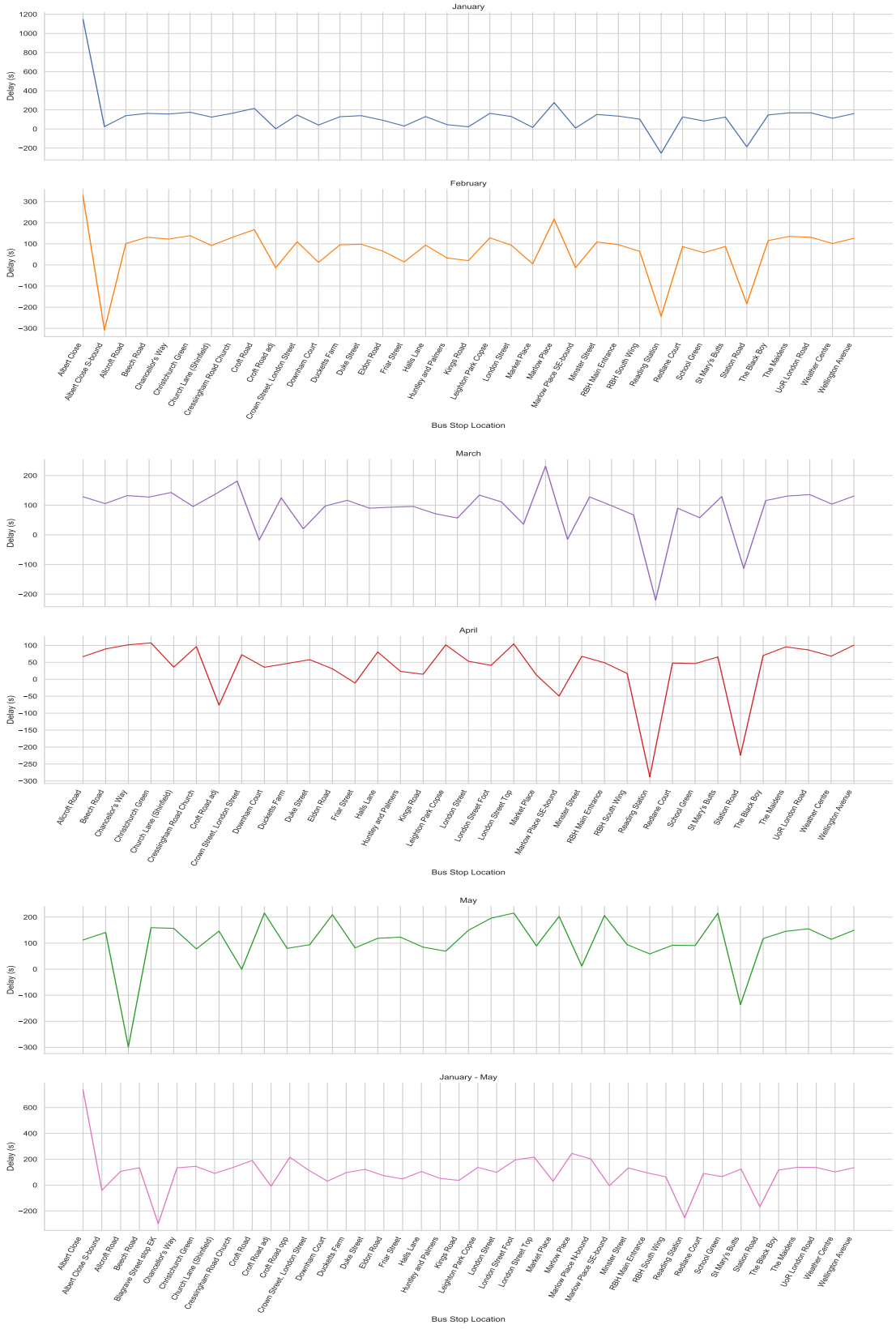


Figure 4.19: The average delay of bus 10 by months and overall months from January to May

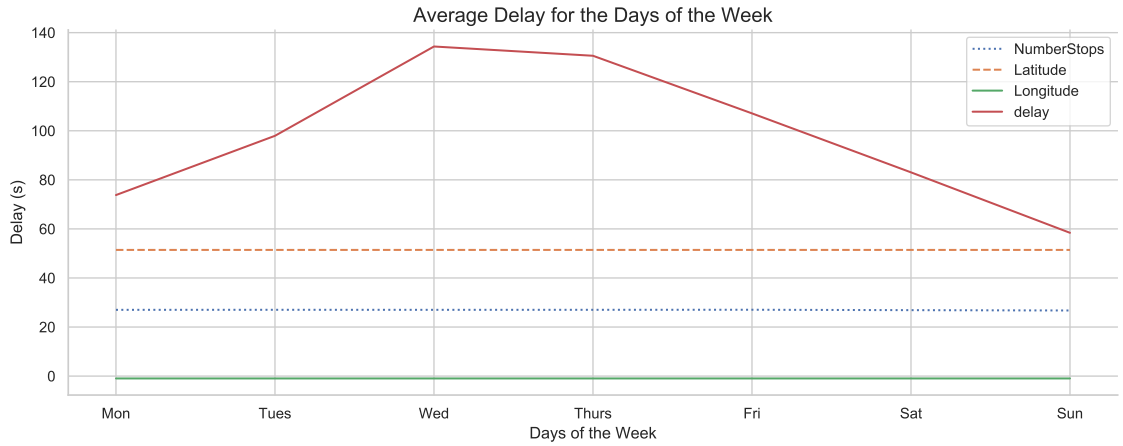


Figure 4.20: Average delay of bus 10 by the days of the week over the course of 5 months

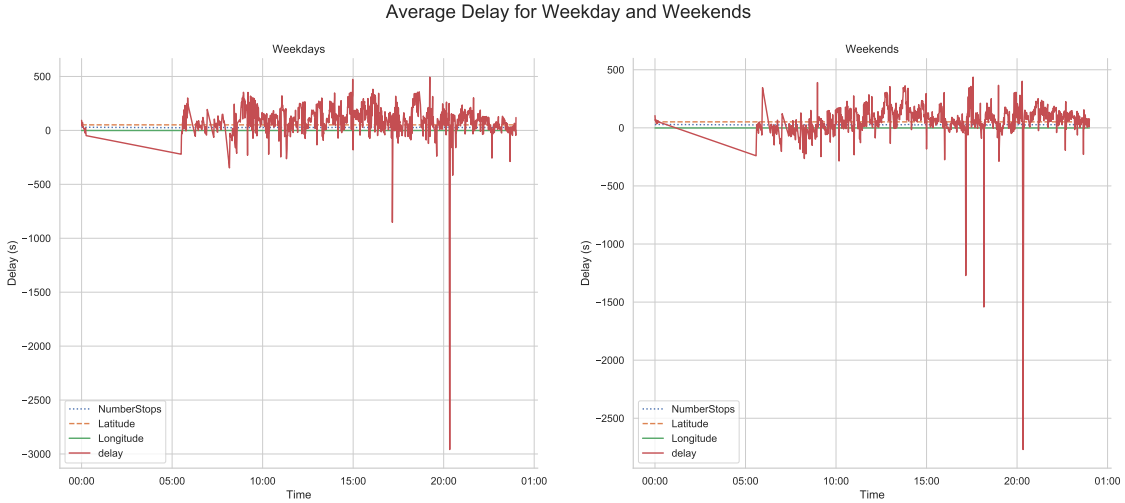


Figure 4.21: Average delay of bus 10 comparing the weekdays to the weekend

Figure 4.22: Average delay of bus 10 comparing the weekday to the weekend of the bus stop, St Mary's Butts

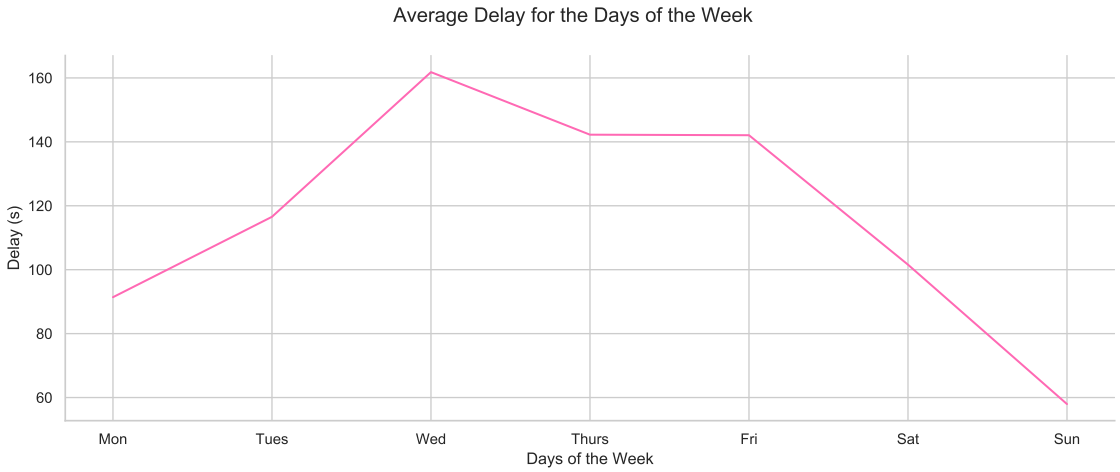


Figure 4.23: Average delay of bus 10 by the days of the week of the bus stop, St Mary's Butts

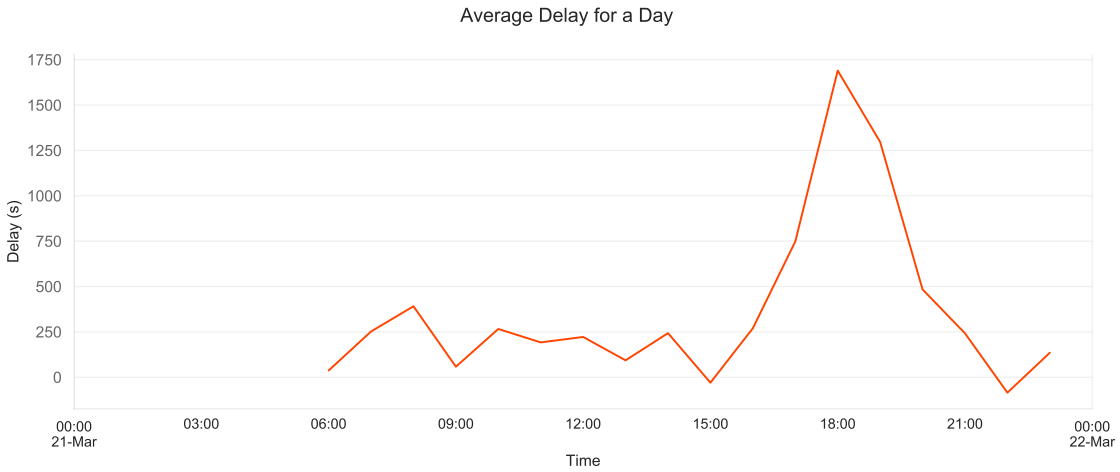


Figure 4.24: Average delay of bus 10 by the days of the week of the bus stop, St Mary's Butts

4.1.5 Bus Number 21 Arrival Times

The following displays the graphs created for the bus service number 21. The bus stop that is visualised to look more in to detail is the Chancellor Way.

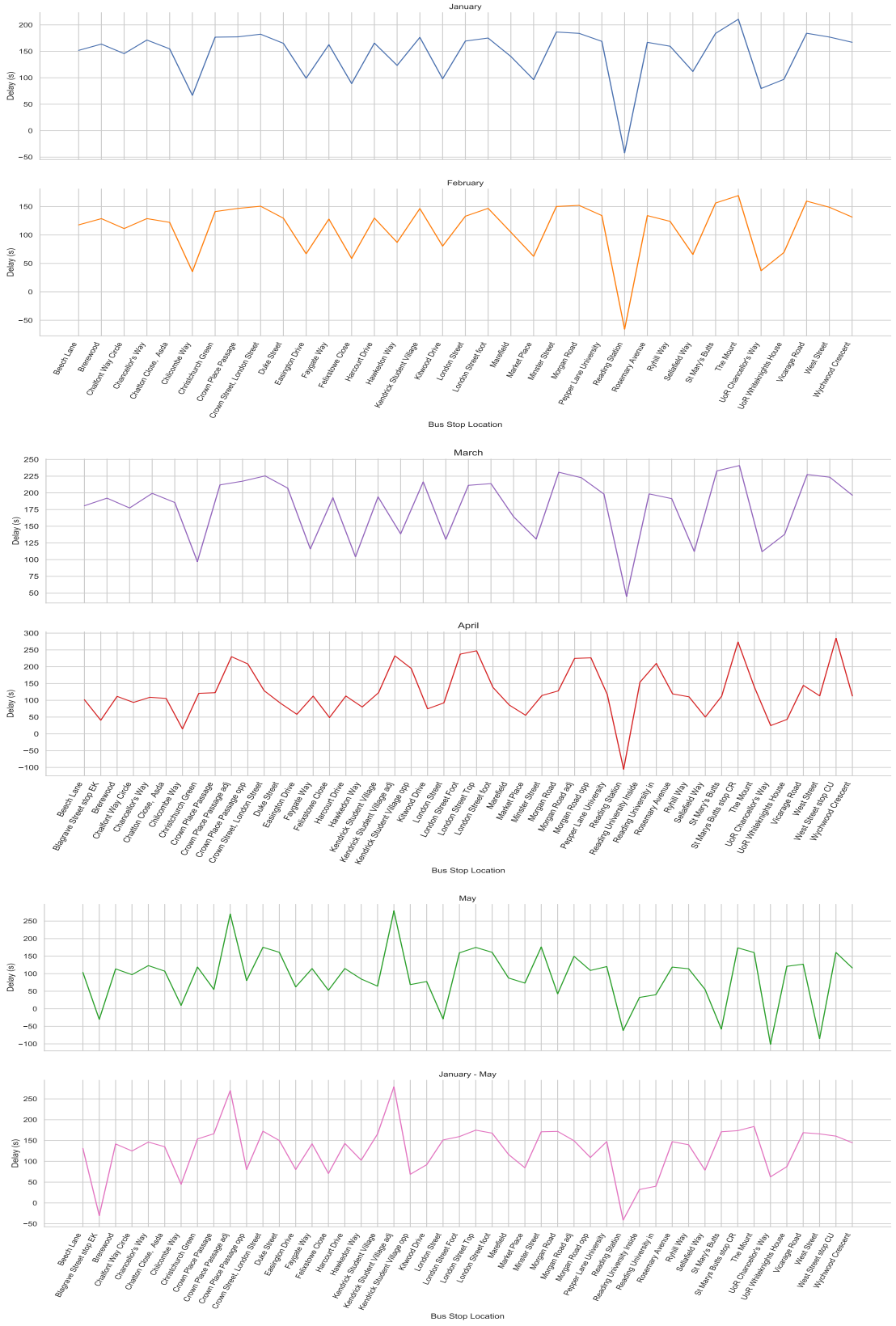


Figure 4.25: The average delay of bus 21 by months and overall months from January to May

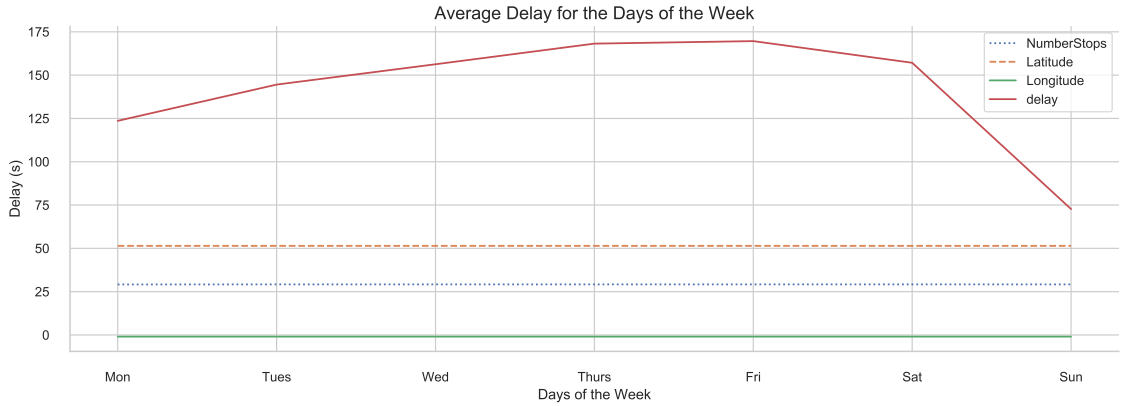


Figure 4.26: Average delay of bus 21 by the days of the week over the course of 5 months

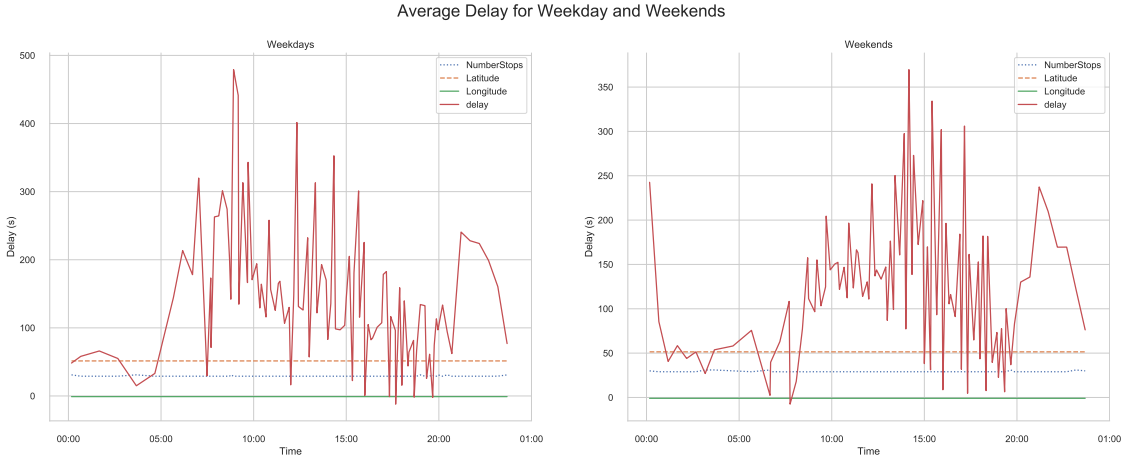


Figure 4.27: Average delay of bus 21 comparing the weekdays to the weekend

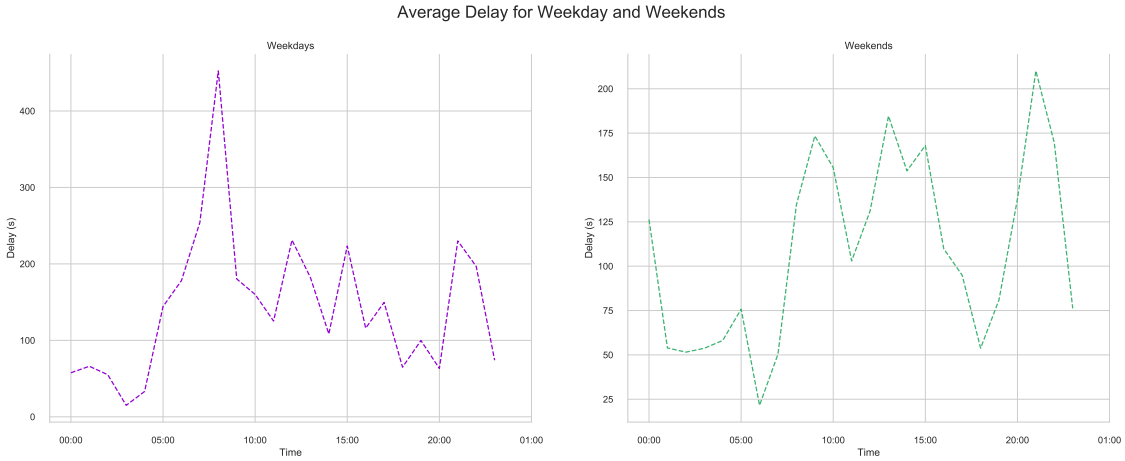


Figure 4.28: Average delay of bus 21 comparing the weekday to the weekend of the bus stop, Chancellor's Way

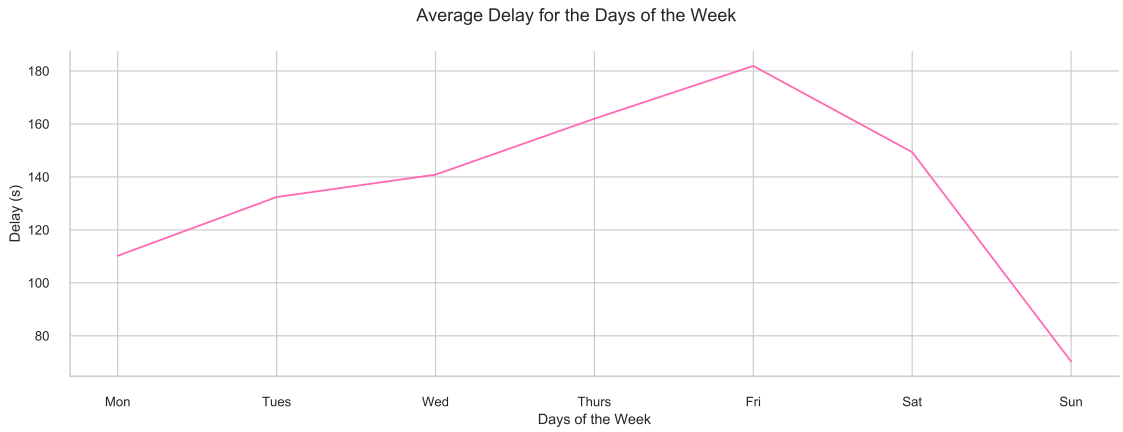


Figure 4.29: Average delay of bus 21 by the days of the week of the bus stop, Chancellor's Way

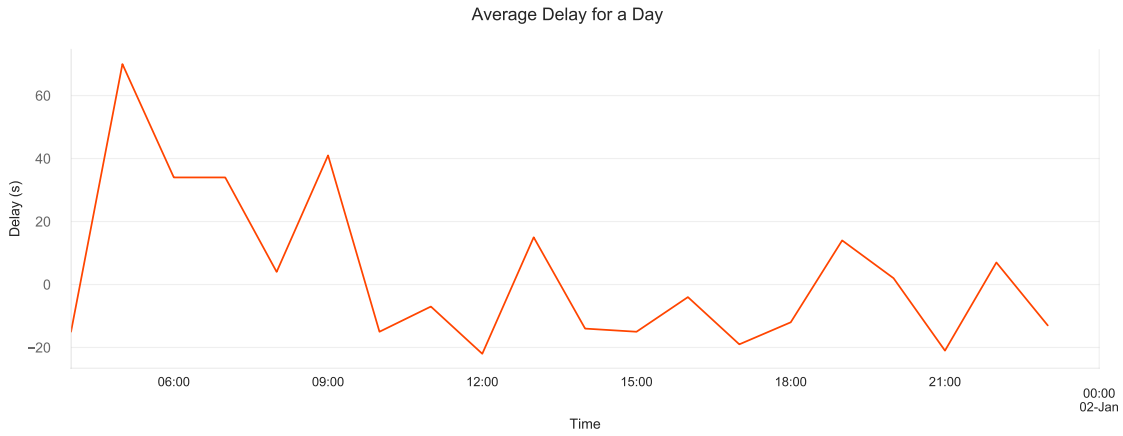


Figure 4.30: Average delay of bus 21 by the days of the week of the bus stop, Chancellor's Way

4.1.6 Bus Number 26 Arrival Times

The following displays the graphs created for the bus service number 26. The bus stop that is visualised to look more in to detail is the Calcot Ikea stop.

Figure 4.31: The average delay of bus 26 by months and overall months from January to May

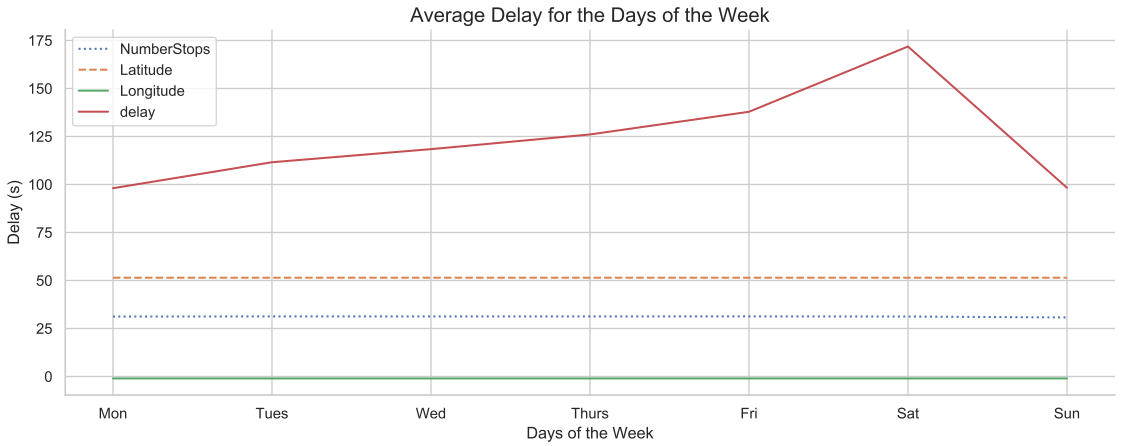


Figure 4.32: Average delay of bus 26 by the days of the week over the course of 5 months

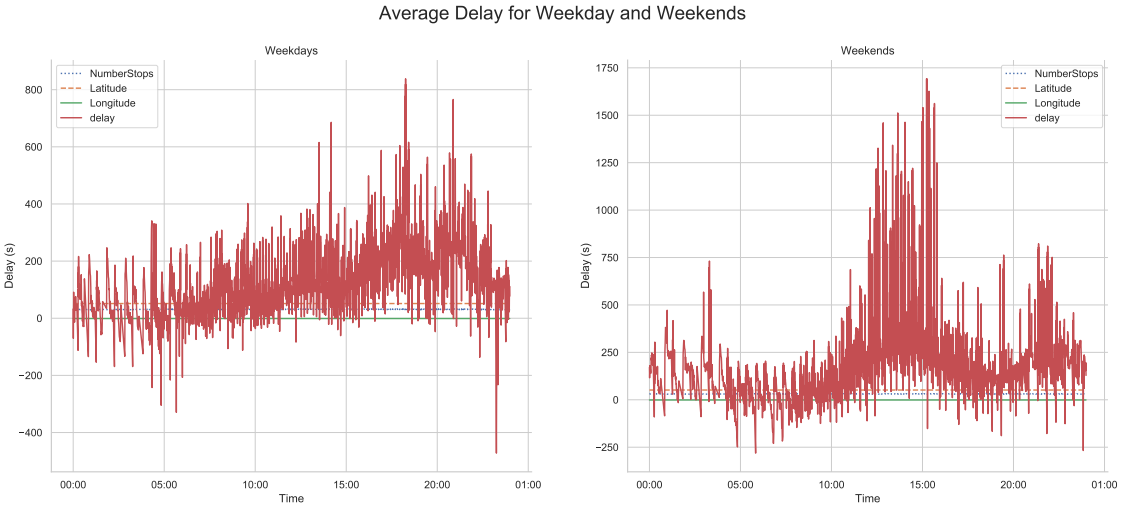


Figure 4.33: Average delay of bus 26 comparing the weekdays to the weekend

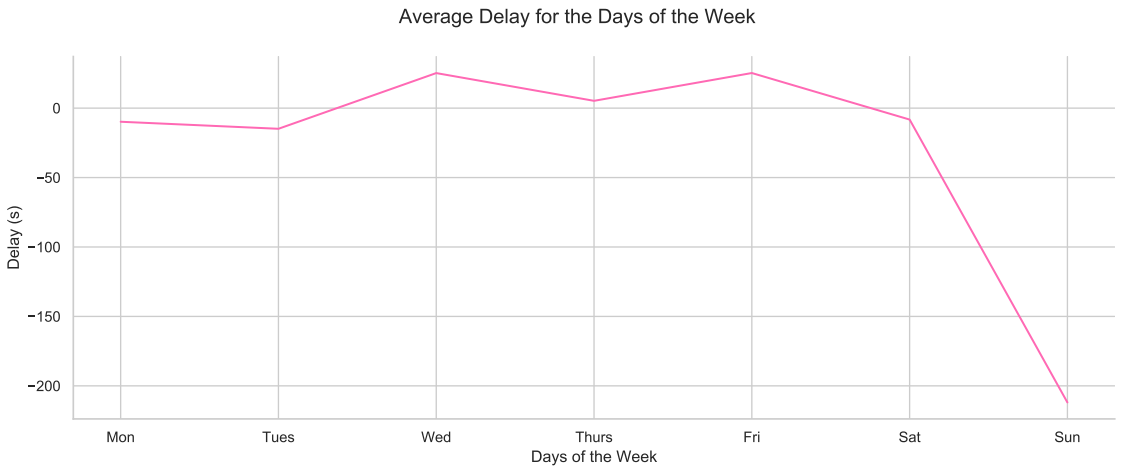


Figure 4.35: Average delay of bus 26 by the days of the week of the bus stop, Calcot (IKEA)

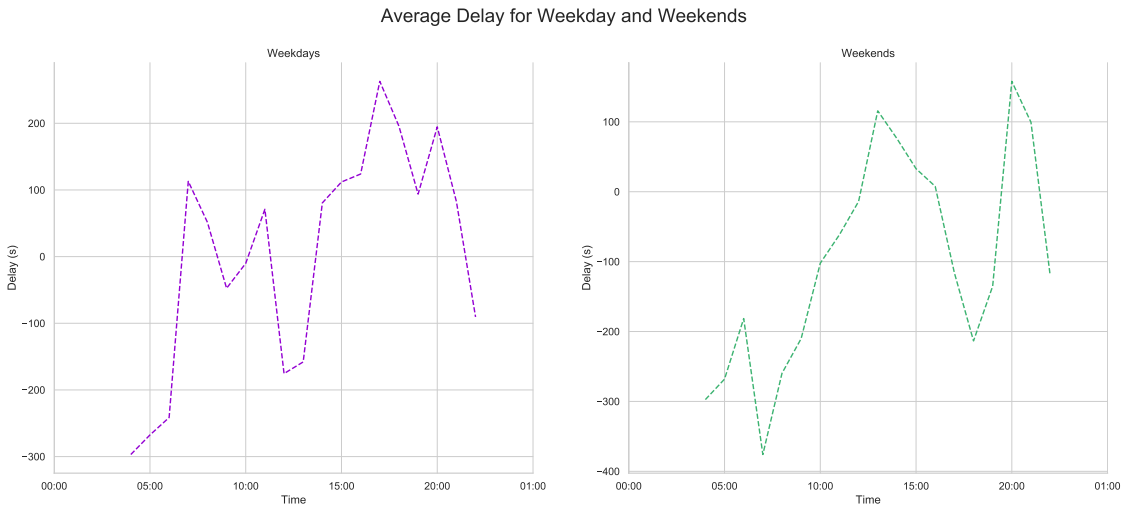


Figure 4.34: Average delay of bus 26 comparing the weekday to the weekend of the bus stop, Calcot (IKEA)

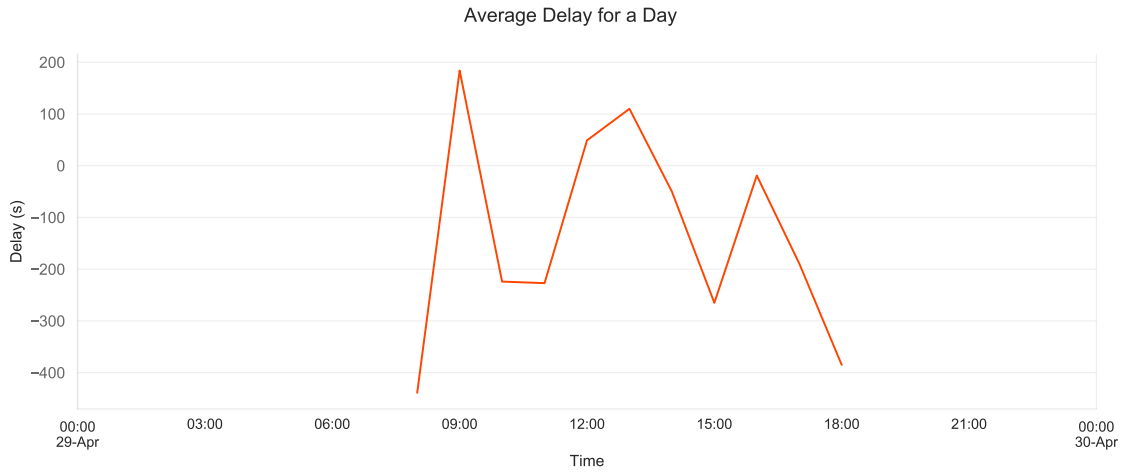


Figure 4.36: Average delay of bus 26 by the days of the week of the bus stop, Calcot (IKEA)

4.2 Arrival Delay Prediction for Bus 21

This section displays the predictions of the delays of the selected buses produced from using the months stated. The data is used to predict both a week and month of future delays.

4.2.1 Using Linear Regression Algorithm

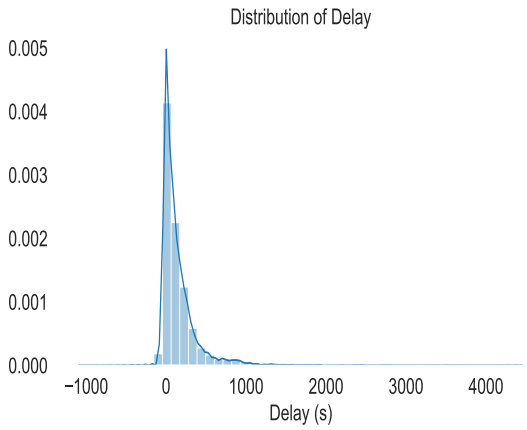


Figure 4.37: Bus 21 distribution of delay in data set

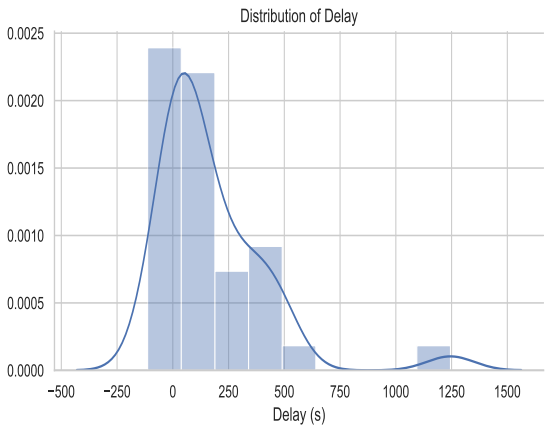


Figure 4.38: Bus 21 distribution of delay by a day

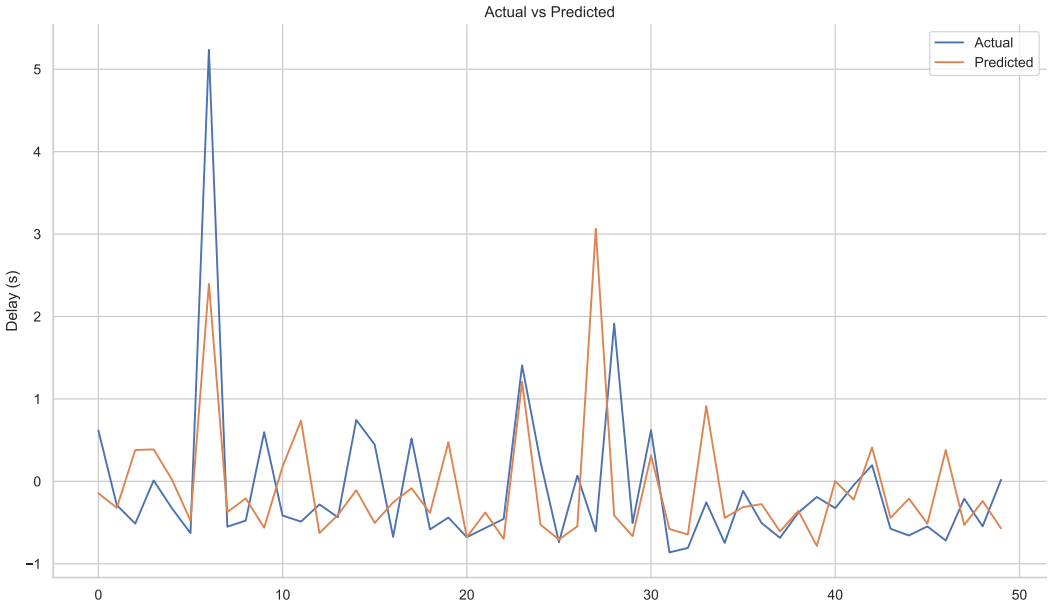


Figure 4.39: Bus 21 Chancellor's bus stop prediction based on data set

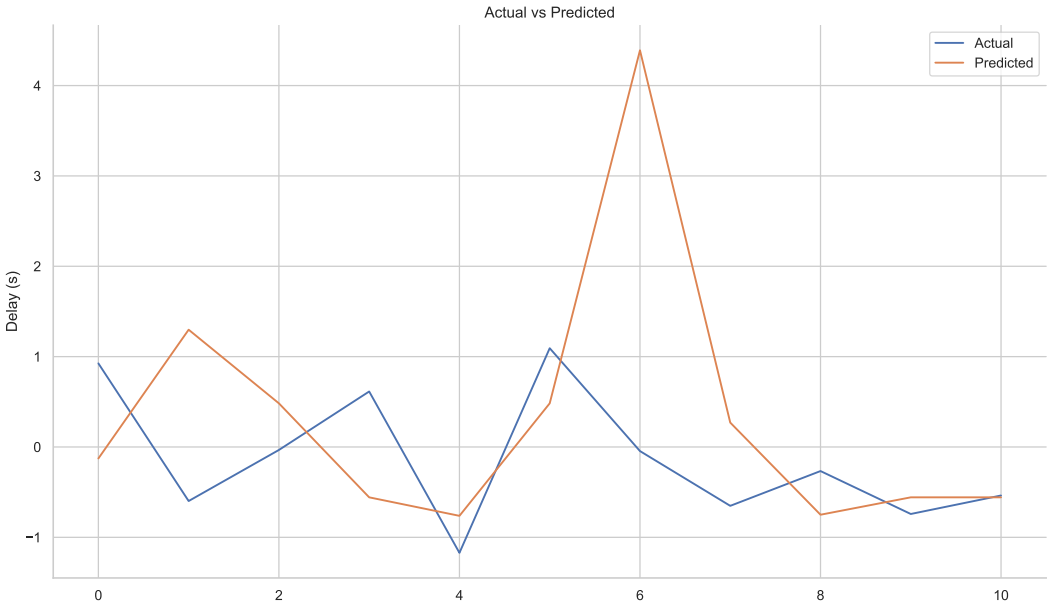


Figure 4.40: Bus 21 Chancellor's bus stop prediction based on a day

4.2.2 Using Long-short-term-memory Algorithm

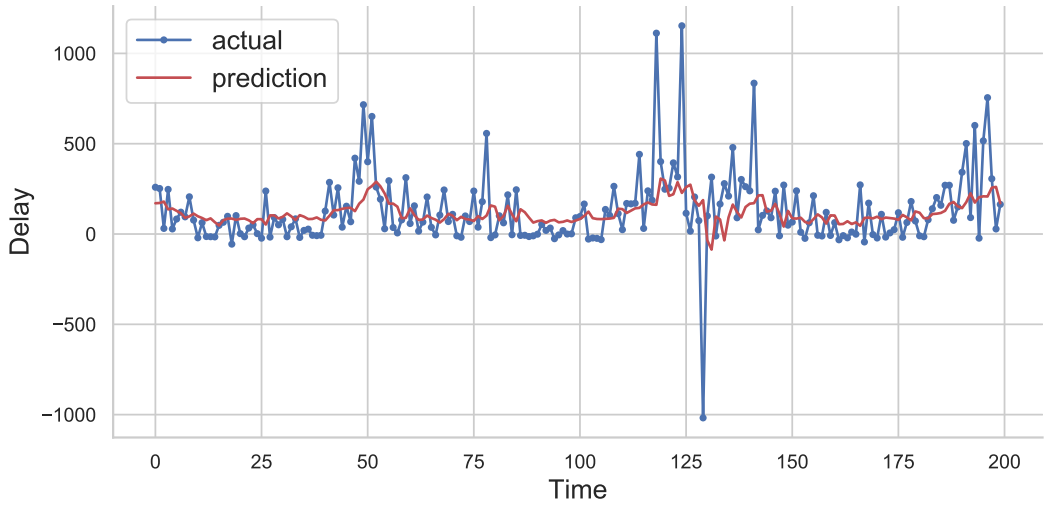


Figure 4.41: Actual and predicted delay of Bus 21 Chancellor's bus stop based on data set

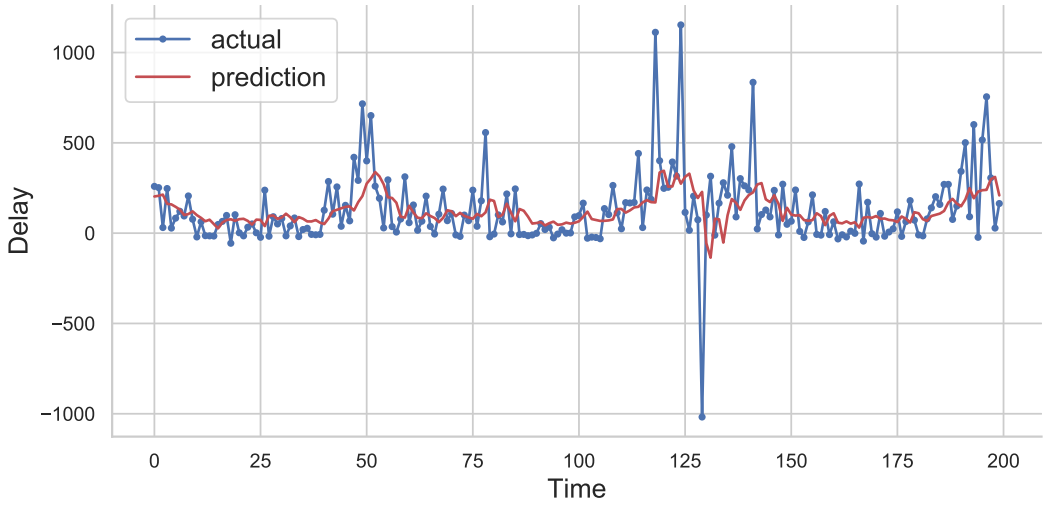


Figure 4.42: Actual and predicted delay of Bus 21 Chancellor's bus stop using second model

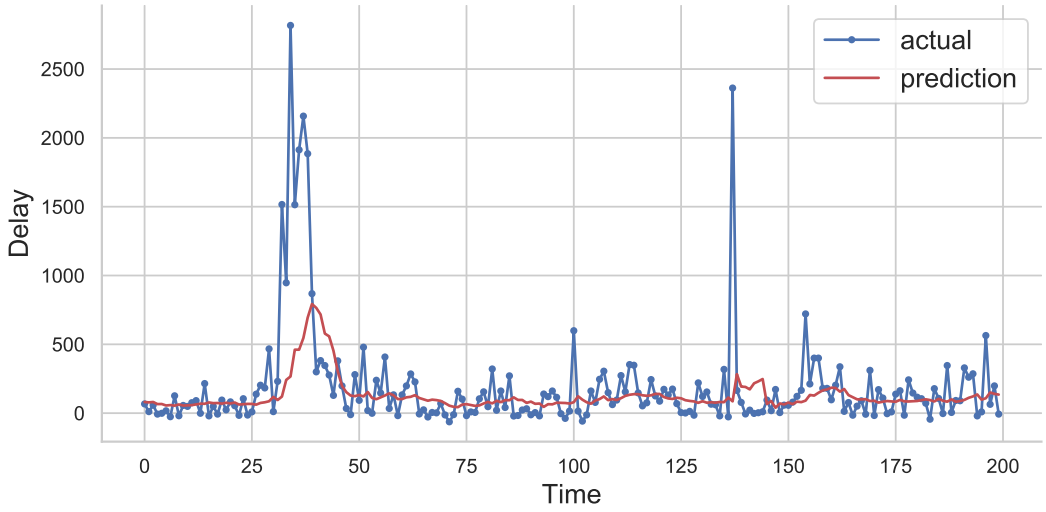


Figure 4.43: Actual and predicted delay of Bus 21 Chancellor's bus stop based on 1 month

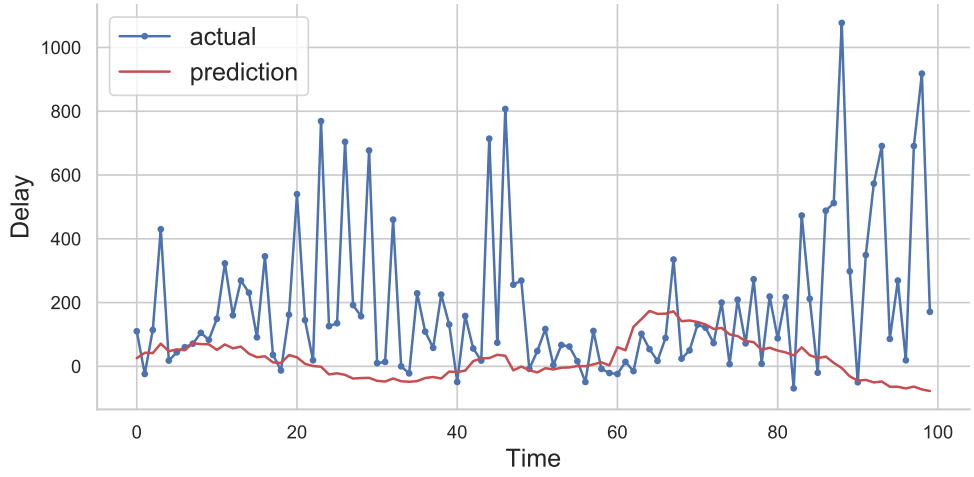


Figure 4.44: Actual and predicted delay of Bus 21 Chancellor's bus stop based on 2 weeks

4.3 Summary

This section illustrates the performance of the buses in Reading throughout the period from January to May. It reflects that the number 3 bus was the best when it came to reliability, even though it has the most bus stops on its route. The buses that are chosen were analysed further by grouping them into periods; this consisted of the single month, day of the week, weekday and weekend and single day. Other than this it illustrates the outcomes of the predictions by using linear regression and LSTM.

Chapter 5

Discussion and Analysis

5.1 Visualisation

Following the methodology of collecting data and prepossessing for visualising and predicting it, the final data set visualised included looking at the data in more detailed specified by the bus service. The need to understand the data consisted of looking at the months individually and collectively to enable the ability to distinguish any patterns in the five months.

5.1.1 Bus 3 and 10 Arrival and Delays

The number 3 and 10 bus belongs to the Leopard service group. These two buses travel on similar routes and share various bus stops. Bus 3 route travels from Reading Station to the town Wokingham and follows back on the same route. This is one of the bus services that leave the Reading area and travels to a surrounding town in Berkshire. Bus 10 travels from Reading Station to the Spencers Woods village and back on the same route. University students also use both bus services as the bus stops right outside the Reading University campus and near the accommodation. The geography of the routes is represented in Fig. 5.1 and Fig. 5.1.

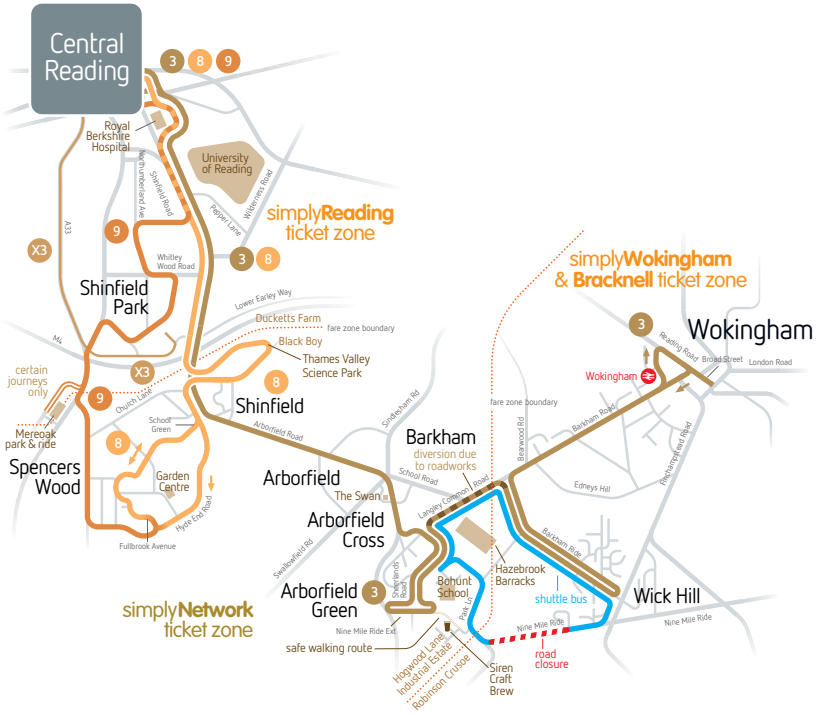


Figure 5.1: Route of number 3 and 8 (10) bus service (ReadingBuses, 2020)

Looking at the first result of bus 3 for the five months in Fig. 4.1 shows that overall, even though there are delays, the actual time of arrivals was not very far off from the scheduled arrival time. Even though this is the case, there are a few differences in the months when looking at specific bus stops. For instance, the Broad Street Stop B bus stop was on an average around 350 seconds (6 minutes) late in January but was around 450 seconds (8 minutes) early in February and was early in May. Every vehicle was late in January except for the Ratepayers Hail bus stop. The lateness indicates that January was the most detrimental month when it came to being on schedule and had the highest delays. The month that seemed to be the most reliable was in February. The graph for this month shows that there were five bus stops, Broad Street Stop B, Church Lane, Reading Station, Sheerlands Road and Station Road Stop, that has an early arrival time and two bus stops, Hogwood Lane and St James Road, that reached the bus stop exactly on time.

Even though bus 3 and 10 shares some bus stops, the performance was different but also contain results with similarities. Looking at the first graphs of bus 10 in Fig. 4.19 shows that the performances throughout the five months are steady with arrival times. Unlike bus 3 in January bus 10 was better at keeping on schedule. The rationalisation behind this was because most delays were under 200 seconds (3 and a half minutes) and only had one very late arrival, which occurred on the Albert Close bus stop. The bus was around 1200 seconds (20 minutes) late on this bus stop and continues results alike to this in other months except for March and April. In these two months, it was only around 60 seconds (1 minute and 100 seconds (1 and a half minutes) late. Even though this is the case, throughout the five months, on average, it is around 800 seconds late (13 minutes). Other than this, the bus did arrive early on some stops within the months. The two bus stops, Reading Station and Station Road show that the bus arrives earlier than scheduled every month. The performance is the same

as bus 3. From the results, it shows that the best month for this bus was in February.

When looking at a bus, it is competent to comprehend how it operates in the week. This can be referred to in Fig. 4.2 where it exhibits that the worst day for being on schedule is on Thursday as on average the delay is around 200 seconds (3 minutes). The only day that has a delay of less than a minute is Saturday. The reason for this may be because there would be fewer vehicles on the road, especially in the early morning and the evening compared to on the weekday. Even though this is the case, from Fig. 4.3 it presents that there is a spike of delay around the hours of 15:00 on the weekend with the highest delay being nearly 1000 seconds (16 minutes). This means that on the weekend it could have the same dilemma with keeping on schedule compared to the weekdays. The weekdays have a delay of 1000 seconds, but this is anticipated because it is during the rush hour period at 09:00. Though from the graph it does show that the bus has mostly a delay between 200 and 600 seconds (3 minutes and 10 minutes) there are times it is early, for instance, around the time of 20:00 the bus is 400 seconds (7 minutes) early and is around this duration often in the morning. This is not only on the weekdays as the weekends also has a surge of earliness at 20:00 and around 07:00.

Looking at the days of the week in Fig. 4.20 shows that the highest delay on average is on Wednesday and the lowest on Sunday. This can be expected as scarcer traffic and people are commuting on the weekend. Even though this might be the case for the bus being a minute late on this day, from the Fig. 4.21 on the weekends it illustrates that throughout the day it can be around 400 seconds (7 minutes) late or even around 300 seconds (5 minutes) early on schedule. It does show that the best time for arrival is between 18:00 and 20:00, for instance, the arrival after 20:00 is around 2700 (45 minutes) early compared to the timetable schedule. This may be because the traffic flow is at its minimal on the weekend, especially in the evening. Compared to the weekdays the graph displays that there are fewer spikes of delay and arrival throughout the day but shows the similarity of early arrival just after 20:00 as it is just under 3000 seconds, making it 50 minutes early. This is very different from bus 3 when it comes to the weekday and weekend.

It was vital to include the analysis of a related bus stop that is on this route to look more into detail of this bus service. The chosen stop was Cressingham Road Church. This bus stop is located near the University of Reading on Shinefield Road, which is one of the busiest roads in Reading. With this being one of the busiest roads throughout the day, it is anticipated that the bus would find it difficult being on schedule, causing the delays to be higher. Other than this factor, being on schedule is affected by the road being under construction yearly by the council. This harms the traffic flow more than usual. Looking at Fig. 4.4 the delays were highest at 15:00 within the weekday and similar outcomes at the same time within the weekend with the delay of 400 seconds (7 minutes). What was expected resulted in the actual outcome and shows that the only time the bus is on schedule or early is around 07:00 and 20:00. This insinuates that when the bus travels on route to this bus stop on the weekday and weekend, it is likely to not be on time.

To look more into this, Fig. 4.5 presents the days of the week. Looking at the graph, it determines that Thursdays and Fridays have the worst delays by increasing past 275 seconds. Saturday is the best day for being on time. This is manifested in Fig. 4.2 from the illustration displaying a decrease in a delay from Friday to Saturday. Looking even more in-depth, the last Fig. 4.6 illustrates the actual arrival time of the bus though out a chosen day, in this instance, it is Thursday 24th May. The graph explicates that in the morning during the rush hour period the delay is very high as it is 2500 seconds (42 minutes) late. This is the latest arrival time

shown from the results for this bus. The duration of delay was expected as declared before, as the bus stop is on a hectic road, and is during the time of increased traffic flow. The increase of delay is not the case for the rush hour in the evening, as the delay decreases to around 800 seconds (13 minutes). Even though this is still late compared to the scheduled time, it is an improvement. The only time the bus is on schedule during this day is around 14:00, 17:00 and again around 20:00. Using this figure helps to acquire a better insight from Fig 4.5 of how the delay of this bus on Thursday operates and how it concludes to be the worst day for delays.

For bus 10, looking more in detail, it included the analysis of St Mary's Butts bus stop. This bus stop is positioned in the town centre in front of Reading Broad Street Shopping Mall. The first results from this bus stop are illustrated in Fig. 4.22. It exhibits that the bus is only on schedule or early around 06:00 and 19:00. This is the outcome on the weekend but with a high early arrival. As expected, there are more delays than early arrivals on the weekday, with the highest delay being at 300 seconds around 16:00. It is surprising that on the weekday at 05:00 the bus is still not on schedule even though there would be no difficulty of this occurring because there would be little to no vehicles on the road at that time. The only reason to believe that the bus could not be on time is how the driver operates as they could believe that because there is no traffic flow, they would have more time and would less hasten to get to this bus stop. Looking at the weekend around 14:00 it shows that the bus is 240 seconds (4 minutes) late, this may be because it is the busiest time that people travel but performs very well considering the bus stop is in the town centre where it is the busiest. Fig. 4.23 shows that like in Fig. 4.20 the bus is latest on a Wednesday and decreases on a Saturday and Sunday. On a Monday, the bus is only late by 90 seconds (1 and a half). The Fig. 4.24 illustrates the bus performance at this bus stop on the 21st March throughout the day, this day fell on a Wednesday. In the graph it shows that at 18:00 it generates the highest delay of arrival by being 1750 seconds (29 minutes) late, this is understandable at this stop as this is during the rush hour time where people are still commuting throughout the town centre. Other than this time, the bus was around 250 seconds (4 minutes) late throughout the day and only was on schedule at 15:00. From these results, it does further determine that Wednesday can be the adversest for delays as this figure correlates with Fig. 4.20.

5.1.2 Bus 5 and 6 Arrival and Delays

The number 5 and 6 bus service belongs to the Emerald service group. The number 5 and 6 bus travels from Reading Station to Whitley Wood, were bus 5 terminates at Northumberland Avenue Terminus and follows back on the same route to the station. Bus 6, on the other hand, terminates at Engineers Court and goes back on the same route. Though these two buses terminate on different stops, they share the same bus stops, therefore, making the routes partially the same. Bus 5 route only consists of 20 to 21 stops, whereas the bus 6 route consists of 16 to 22 stops being that it goes a little further on its route. These two bus services are one of the shortest routes of Reading Bus services. The geography of the routes is represented in Fig. 5.2 and Fig. 5.3.

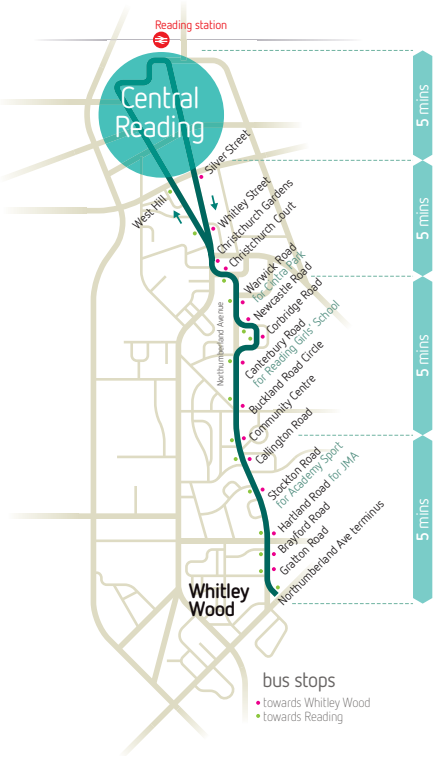


Figure 5.2: Route of number 5 bus service (ReadingBuses, 2020)

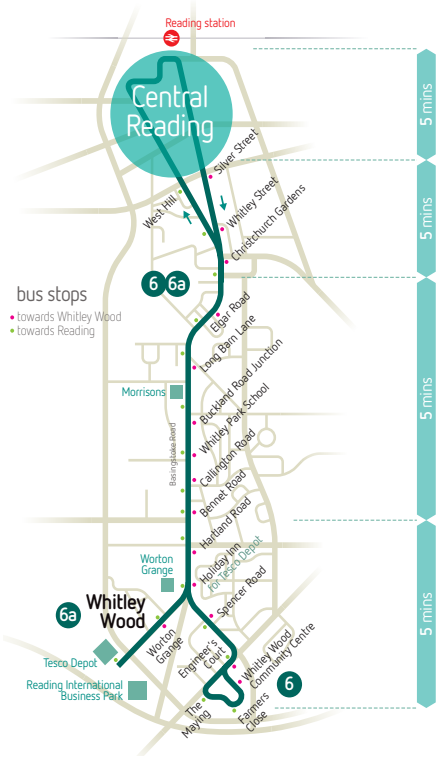


Figure 5.3: Route of number 6 bus service (ReadingBuses, 2020)

The first Fig. 4.7 looks at how the bus operates within the five months. From looking at the graphs, it shows that within most months the bus is not on the schedule. Nevertheless, the delay is no more than 200 seconds (3 and a half minutes) concluding that the bus has had the best performance out of all the analysed buses by only being a few minutes late and no more than 5 minutes. Even though these are the results, it could be effective with the satisfactory of commuters by being at this rate every month. The two bus stops that result in positive outcomes by arriving earlier in the Blagrave Street Stop and Reading University bus stop. In the graphs, it proves that every month this bus is moderately early whether this may be only 100 seconds or 300.
On the other hand, referring to Fig. 4.13 shows that bus 6 operates a similar way in April and May as it also performs on the early side when it comes to the Blagrave Street Stop and Reading Station bus stop. In January, February and March, the results show that the bus is more than 2 minutes late. For instance, the bus stop Bridge Street is always around 250 seconds(4 minutes) off schedule every month, whereas compared to bus 5, this bus stop is only ever around 100 seconds (1 minute and 40 seconds) late. This is not the only case for this stop when comparing these bus services with each other on the corresponding stops that they share.

Looking at how the bus operates throughout the five months; the days of the week is illustrated in Fig. 4.8. This graph reveals that every day of the week the bus is late is on average and not on schedule, especially on Thursday. The arrival time for each day is not drastic since at most the bus is late is by 200 seconds (3 minutes), this is only a maximum of 3 and a half minutes. This is the same case for bus 6 in Fig. 4.14 where the maximum delay is only 200 seconds

but instead of a Thursday, it is on a Saturday. In Fig. 4.9 the graph shows that the bus has an extensive delay in the morning around 08:00, during rush hour, with the delay being 1000 seconds (16 minutes). Compared to the weekend, the bus at 08:00 was nearly on schedule by only having around 100 seconds of a delay. On the weekend, the only time the bus gets to the same point as the weekday is 15:00 by being approximately 900 seconds early. The bus on the weekend is on schedule or arriving earlier than the bus on the weekday. This can be anticipated because more people are commuting by bus on the weekday. For example, just before 20:00, the bus is over 750 seconds (13 minutes) early, whereas on the weekday around this time the bus is only 400 seconds (7 minutes) early. Referring back to bus 6, in Fig 4.15 it reveals that the bus was very late on various times on the weekday by being late up to 30 minutes and only being early by less than a minute throughout the day. On the weekend the delay is later but is right on time between 05:00 and 10:00. The most prolonged delay of this bus was around the time of 15:00 by being 16 minutes late.

Looking at bus 5 in more detail was done by choosing the Whitley Street bus stop. This bus stop is located halfway between the University of Reading and the centre of Reading. The first graph in Fig 4.10 showcases the weekday and weekend delay. It indicates that the time for the highest delay is at 18:00 by being 320 seconds (5 minutes)late on arrival, this is quite different from the weekend delay where the delay for the weekend increases up to 450 seconds (8 minutes) at around 14:00. On the weekend the bus is only a few seconds early in the morning and a minute late before 10:00 whereas on the weekday it is not on schedule throughout the day. Fig 4.11 illustrates the days of the week. It explicates that the worst day for the bus was on Friday, which then decreases drastically to nearly being on time on Saturday and going back up to being late on Sunday. What is interesting is that the bus is more alike to the arrival schedule at the start of the week compared to the end, as Monday and Tuesday show a lesser delay than on Sunday. The last figure, Fig 4.12, uses Saturday 17th March to display how the bus performed on this day. Looking at the figure, it exhibits the highest delay occurring at approximately 15:00 and then again around 18:00, which is the time near the school run and the rush hour period. Looking at around the time of midnight and the early morning the bus happened to be still not able to arrive on schedule until it reached 06:00. The results displayed from this figure helps to get a more understanding of how Saturdays are the lowest when comparing it to Fig 4.11.

Referring to bus 6, the choice of a bus stop to understand the operation of this bus service is the Reading Station bus stop. This bus stop is located outside of Reading Station and is used by many commuters. Looking at the first graph in Fig 4.16 in showcases the weekdays and weekends. In this figure, the data plotted has a different outcome, as it shows that there was not much data to go on. Using the provided results shows that the bus arrived early at midnight and again the following midnight and was not on schedule around 01:00 in the morning by about a minute. This reveals similar results to the bus on the weekend as it also arrives earlier than scheduled around the similar time and 05:00. Results might have been different if there was more data on this bus to analyse to contrast the difference between the weekday and weekend. Looking at the weekday and weekend it does show the same pattern at the same times, so this can indicate that at around 05:00 the bus would show up earlier than the scheduled arrival time. Compared to bus 5 Fig 4.11, in Fig 4.17 it shows that bus 6 performed very well on this bus stop because the bus arrives earlier than scheduled. This is very different from the other buses. In Fig 4.18 it illustrates the performance of the bus on Friday 13th April. There was very little data plotted, but from what is available it shows that the bus was early in the early morning and midnight. This figure does show correlation with

Fig 4.17 by showcasing that the bus has good arrival time on a Friday.

5.1.3 Bus 21 Arrival and Delays

The number 21 bus service belongs to the Claret service group. The bus is the dedicated service for university students for commuting. The bus service route of the number 21 travels from Reading Station to Lower Early and vice versa. Between these two endpoints, the route consists of the bus going through the town centre of Reading, passing off-campus university accommodation and continues to go in the direction passing through the campus and then towards Lower Earley. The start point is the Reading Station. This travels to the endpoint bus stop, Chilcombe Way, it then turns back around to go back on the same route to the endpoint, Blagrave Street Stop. This service operates on a frequency of a 5- and 10-minutes basis and up to every 7/8 minutes during term time. The geography of the route is represented in Fig. 5.4.

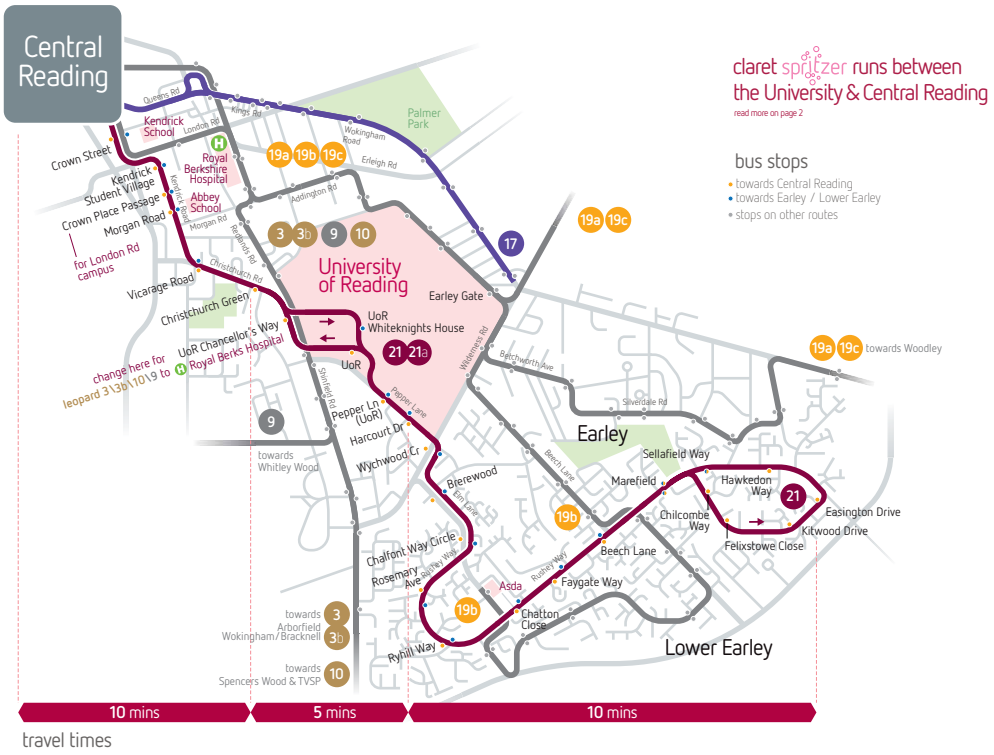


Figure 5.4: Route of number 21 bus service (ReadingBuses, 2020)

Analysing data of bus 21 by visualising the data on each bus stop from each month was consistent and was similar in the arrival time, as it did not change drastically. For instance, looking at Fig. 4.25 demonstrates that the Reading Station is not exceptional for arrival time when it comes to being on schedule, as in the figure it shows that 4 out of 5 months the bus is late because of the value of delay. Bus 21 in April and March produced the adversest results contrasted to the other months because it increased in delay. Overall, the bus stop Reading Station and Blagrave Street were the only bus stops that did not display difficulty with operating on schedule on this route. The result of this can be positive. For example, the

bus will come earlier for commuters and would decrease the waiting time at the bus stop and could allow the departure of the bus to be early and improve the arrival time of the next bus stop on the route. Other than these two bus stops there was a negative side, as the other bus stops on this route are either slightly or drastically late compared to the arrival schedule, for example, the Kendrick Student Village bus stop is very late on arrival. It indicates a high value of delay. Viewing the five months especially May in Fig. 4.25 for this bus stop the value is around 260 seconds (4 minutes), showing that bus 21 does struggle on this stop to be on schedule or even early.

Fig. 4.26 illustrates the delays of all five months by the days of the week. This helps to grasp and visualise the delays of the route according to the day of the week. Fig. 4.26 exhibits that on Sunday the delay decreases and the bus is more likely to be on schedule, this may be caused by the frequency of the bus, as buses operate on Sunday times which means that the bus may come every hour instead of every 15 minutes. Other than Sunday, there is also a decrease in delay on Saturday. This can aid the understanding of Fig. 4.27 when viewing the times for the weekend. From the Fig. 4.27 the weekend has the highest later arrival time between 14:00 and 18:00, whereas the weekday has late arrivals between 07:00 and 9:00 and a repeat at 12:00 and 16:00. Delays may persist because of rush hour in the morning and during school runs. This is when the roads are the most congested and the cause of traffic. The congestion then affects the arrival and departure of the bus. Referring to Fig. 4.26, the delay on Monday compared to the days in the middle of the week is not near the equivalent value. This is considered to be more of a delay because it is the start of the week and usually the day many people go back to work.

It was best to look at a bus stop that is mostly used by university students of way to commute. Looking more in-depth to this bus service was done by analysing the Chancellor's Way bus stop. The bus stop is located at the front of the university. The Fig. 4.28 and Fig. 4.29 presents the delays in the days of the week and the comparison between the weekday and weekend. This helped to get an insight into how efficient this bus is at this bus stop. Fig. 4.28 illustrates the delay on weekdays and weekends and displays the fluctuations that occur. Understandably, the delay is higher in the early hours on the weekday, as it is rush hour so the bus arriving at Chancellor Way would be delayed, in this instance by 400 seconds. The delay during the time around 20:00 is quite high despite the fact it would be quieter when it comes to vehicles on the road. Even though this is the case, it does decrease by the time it gets to midnight. It is also unexpected that this is also the case during the weekend. The weekday and weekend both insinuate that the bus arriving at this stop is adequate when it comes to the timetable by being no more than 10 minutes late. Fig. 4.29 displays that on Friday it is the worst when trying to keep on course with the timetable as the delay is the highest out of the seven days of the week, but on Sunday it does become better by being less than 80 seconds late. Fig. .4.30 uses the 1st January to show the delays throughout that day. The reason for looking at the performance of the bus for this date was because 1st January is New Year's Day. As this is classed as a holiday, the frequency of the bus would follow the standard bank holiday timetable, which could affect how the bus performed. The figure displays that the arrival delay was not inadequate for this day as the highest delay was only a minute late. This designates that on this day the bus was on schedule as from 10:00 it had an occurrence of arriving early.

5.1.4 Bus 26 Arrival and Delays

The number 26 bus service belongs to the Yellow service group. This bus service operates on a frequency of every 5 minutes from Reading Station to Calcot (IKEA) where it terminates and turns back to go on the same route to Reading Station. It is a service that is used by many university students at the start of the academic year to travel to IKEA to get any essentials they may need. The geography of the route is represented in Fig. 5.5.

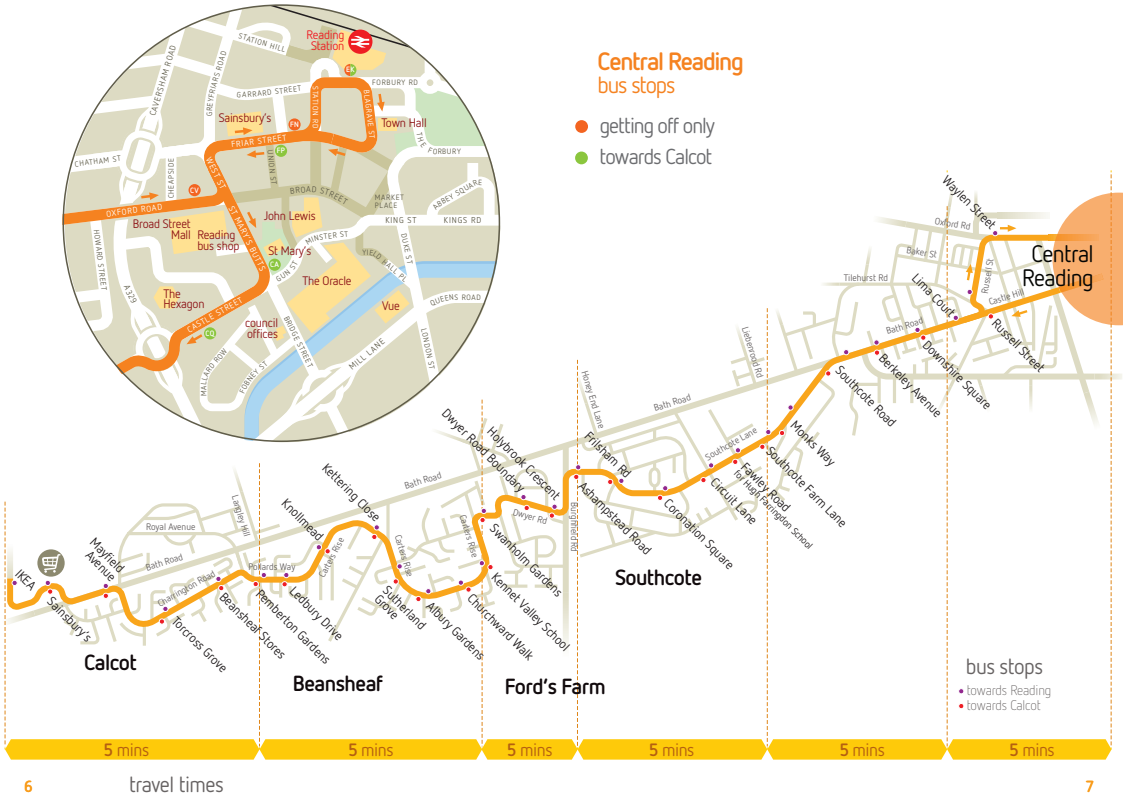


Figure 5.5: Route of number 26 bus service (ReadingBuses, 2020)

The first results are from understanding the performance of the number 26 bus illustrated in Fig. 4.31. The figure represents the bus within five months. From looking at the figure, it is effortlessly displayed that the bus performance results in it being late by the maximum of around 7 minutes throughout the months. For example, this is demonstrated in March at the Russel Street bus stop. Within the other months, the maximum lateness of the bus is 5 minutes and upkeeps a constant level of being around 100 to 200 seconds late on most bus stops. On the positive side, there have been moments where the bus has been on schedule. This has been demonstrated throughout all the months. A given example of this is when the bus either reaches Cheapside, Russell Street or Waylen Street bus stops they were precisely on schedule. Even better, the bus also had the action of arriving earlier than the schedule. This occurred at the Calcot (IKEA), IKEA NE-bound, Reading Station and Lima Court bus stops. The arrival times that occurred earlier than scheduled time consisted of being around 100 seconds (2 minutes) early. From the figure, the only time the bus was never arriving earlier was in March. It is notable in this month by the bus either being on time or late.

Looking at how this bus performed within the different months, caused interest to look at how it performed by the days of the week. This is illustrated in Fig. 4.32. The figure explicates that there is a slight increase in the delay that occurs every day throughout the week. The lowest delay starts on Monday at 100 seconds and increases every day from there until Saturday. Saturday is the latest by being off schedule by 175 seconds (3 minutes) and then decreasing on Sunday to being late by 100 seconds again. Surprisingly, the bus is later on a Saturday than any other day during the weekday, as this is mostly the busiest time for traffic flow and the number of people commuting by bus. Unfortunately, like the other figures illustrating the data of the bus by weekdays and weekend was unavailable to plot for this bus, as the data was unsubstantial to visualise.

The bus stop that was chosen to analyse was the Calcot IKEA stop. This stop is located outside the IKEA store, so that is easier for commuters. To get an insight of this bus stop, Fig. 4.34 shows the visualisation of the difference and similarities between the weekday and weekend. In the figure, it does show a few similarities, for instance, around the time of 05:00 the bus is exactly 300 seconds (5 minutes) on the weekday and weekend. Possibility of this occurrence could rely on the traffic flow as it is at its minimal so early in the morning. This is not the only similarity as after 20:00 the bus on weekday and weekend is just over a minute early. The difference does present itself, on the weekday it is twice as much of a delay for the bus, as it is more than 200 seconds late between 17:00 and 20:00.

The delay during these two times is understandable because it is during peak hours. Another discrepancy is that the bus on the weekends is more on track than the weekday as it consists of arriving at stops earlier than the specified time on the timetable. Other than looking at this figure, the insight of the performance in the days of the week is acknowledged. Fig. 4.35 reveals that throughout the week performs thoroughly by either arriving very close to the schedule or arriving earlier on most days of the week. From looking at all the days, it is only a few seconds from the scheduled time whether this may be on the early side or late side. This is particularly demonstrated on Sunday as the bus is just over 200 seconds early. There was a choice of just looking at just one day within the months; this is shown in Fig. 4.36. The choice of using Sunday 29th April was at random. From the figure, it reveals that in the morning before 09:00 the bus was more than 400 seconds (7 minutes) early and when it got to 09:00 it was delayed by around 200 seconds. This duration of delay was the longest throughout the day, and after this time, the delay transpired twice and maintained early arrivals. By 18:00, the bus went back to being around 200 seconds early. The results of this correlate with Fig 4.35 by showing how it performs adequately by occurring early on a Sunday.

5.2 Prediction

The main choice of using bus 21 for performing a prediction was because it is a bus that is used widely by university students to commute in Reading, as it is mainly used to commute to town. This is a service that is dedicated to transporting students to and from campus. Table 5.1 and table 5.2 represents the modified data set that was used for the prediction. It used the scheduled arrival time and actual arrival time by looking at the difference between the two and outputting the delay.

ScheduledArrivalTime	Delay
2018-01-01 04:40:00	15.0000
2018-01-01 05:40:00	70.0000
2018-01-01 06:42:00	34.0000
2018-01-01 07:42:00	34.0000
2018-01-01 08:42:00	4.0000

Table 5.1: Example of the data set head for bus 21 prediction

ScheduledArrivalTime	Delay
2018-05-31 19:45:00	399.0000
2018-05-31 20:12:00	142.0000
2018-05-31 20:27:00	180.0000
2018-05-31 21:12:00	364.0000
2018-05-31 22:41:00	493.0000

Table 5.2: Example of the data set tail for bus 21 prediction

5.2.1 Linear Regression

The method to implement linear regression consists of using *sklearn* imports in jupyter notebook. The data consisted of using the months January to May, just the month May and 31st May for the prediction. Linear Regression can be used by using these imports, which is shown in Listing 5.1.

```
1 from sklearn.model_selection import train_test_split
2 from sklearn.linear_model import LinearRegression
3 from sklearn import metrics
4 from sklearn.preprocessing import StandardScaler
5 from sklearn import tree
6 from sklearn.metrics import r2_score
```

Listing 5.1: Code to split the data set in to training and test sets

Pre-processing

The months and days were visualised using a distribution plot of the Chancellor's bus stop within bus 21, this is illustrated in the Fig. 4.37 and Fig. 4.38. These two figures show the distribution of delays within the five months and 31st May. Fig. 4.37 presents that most delays are between 0 and 500 seconds and some up to 1000 seconds. From this, it also shows that there is a limited amount when it comes to the delay being more than 1000 and that the value of the bus being early was minimal compared to being late. On the other hand, Fig. 4.38 shows that the 31st May, the delay was minimal by being at 1250 seconds (21 minutes) and was mostly between 0 and 600 seconds. Using distribution plot gives knowledge of how the whole data set might influence the process of detecting and processing outliers.

There needs to be the process of splitting the data into a train and test data set to train a model and make a prediction. This is accomplished by splitting the data set into two using the function *train_test_split* that is imported from *sklearn*. This is demonstrated in the Listing 5.2. The listing displays the data being held for testing by a 30% split. This is by the *test_size=0.3* inside the function. The split of the training and testing data sets to be 70% training and 30% testing. This then gives the shape of *x_train* and *y_train* to be (7089, 1), *x_test* and *y_test* to be (3039, 1). The *random_state* is assigned the value of 0, this can be modified so that it will produce the same random selection of data and is useful for reproducibility of results.

```
1 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3,
      random_state=0)
2 print (X_train.shape, y_train.shape)
3 print (X_test.shape, y_test.shape)
```

Listing 5.2: Code to split the data set in to training and test sets

Part of pre-processing the data consisted of declaring the x and y variables by assigning the two columns to each variable. The schedule arrival time was assigned to x, and the delay was assigned to y. The values of these were then reshaped to $(-1, 1)$ as providing a new shape makes it compatible with the original shape and by using *numpy* it allows it to give a new shape parameter. Reshaping the array using these values is used because the data has a single feature. The next step of pre-processing is to scale the features. This process is called scaling and is performed on the variables, x and y. The *sklearn* import *StandardScaler* uses the definition of standardisation to standardise the features of data by centring the data and removing the mean to scale it to the unit variance. This helps by normalising the values to improve the accuracy of the model. The process uses the formula $z = (x - u)/s$, where x is the variable, u is the mean, and s is the standard deviation. Scaling is demonstrated in Listing 5.3.

```
1 scaler = StandardScaler()
2 scaler.fit(x)
3 x = scaler.transform(x)
4
5 scaler.fit(y)
6 y = scaler.transform(y)
7 y
```

Listing 5.3: Code to split the data set in to training and test sets

Train and Predict

```
1 #training the algorithm
2 regressor = LinearRegression()
3 regressor.fit(x_train, y_train)
4 y_pred = regressor.predict(x_test)
```

Listing 5.4: Code to split the data set in to training and test sets

A prediction from the linear model needs to be built by using the training data set. To accomplish this, the import of *LinearRegression* class is used from *sklearn* module. The model is trained by it being fitted on the training data so that we can try and predict the test data. The algorithm is trained, as shown in Listing 5.4. The fit method in the code is called to train the algorithm on the training data using x_train and y_train. These variables are passed as a parameter to the fitted method. This is used to try to predict the test data, x_test, this is on line 4. The output of the actual data and predicted uses the variables y_test as the actual and y_pred as predicted. There was the use of regressor model score function to determine the accuracy of the algorithm to obtain an interpretative understanding of the model validity. Linear regression models can use the statistics MSE and correlation coefficient r from *sklearn.metrics* module.

5.2.2 Long Short-Term Memory

Model Training

The method has been implemented by *Keras* framework. The prediction uses the data of the scheduled arrival time and the delay to predict the selected bus stop and data, which was from January to May. Between January and May, the data was split into a training set and a test set. For machine learning, it is needed for the algorithm to be trained, then validated and tested before it can be deployed for an outcome. Training data is the data that is trained to build a model, as the model intends to learn the data set, which can consist of patterns and correlations. It can relate to the outcome of the model overfitting or underfitting. The test part of the data set is used to test the model hypothesis. The model is applied to the test data to receive an accurate measurement of its performance to predict the future. The split consists of the training data set to be 80% of the observations that are used to train the model and leaves the remaining 20% for testing the model. Listing 5.5 demonstrates how the split is accomplished.

The validation part of the data set is used to validate the fit of the model, as it can help to improve the model hyperparameters. It is not for the model to learn the validation data but to apply it to get a better state of the hyperparameters. Hyperparameters consists of the hidden units, dropout, dense, epochs and batch size. Using *Keras* Sequential model for creating the LSTM model uses a linear stack of layers. The model architecture is displayed in Listing 5.6.

```
train_size = int(len(dataset) * 0.80)
test_size = len(dataset) - train_size
train, test = dataset[0:train_size,:], dataset[train_size:len(dataset),:]
print(len(train), len(test))
```

Listing 5.5: Code to split the data set in to training and test sets

Specify the Hyperparameters

```
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
from keras.layers import *
from sklearn.metrics import mean_squared_error
from keras.callbacks import EarlyStopping

model = Sequential()
model.add(LSTM(50, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')

model.summary()
```

Listing 5.6: Python code of LSTM model hyperparameters

First, the LSTM model consists of creating an instance of the Sequential class and creating the layers by the order it should be connected. The model needs to know what input shape it should expect. As of this, the first layer receives 'X train' NumPy array, which is three-dimensional and comprises of samples, timesteps, and features. Samples are the rows in the data, and timesteps are the past observations for a feature and features are the columns in the data. From the Listing 5.6 it shows how the LSTM has one layer and is trained with 50 units in the hidden layer consisting of 1 neuron each. The dropout value is a percentage between 0 and 1 were 0 is no dropout, and 1 is no connection. In this model, the dropout is 0.2 (20%) where every hidden unit is set to 0 with a probability of 0.2. This means that there is a 20% chance that the output of the neuron will be forced to 0. The dense layer is a densely connected NN layer Keras (2020). The parameter details the number of neurons that will consist within the hidden layers and is used for outputting a prediction.

Compilation

Still referring to the Listing 5.6, before the training occurs the configuration of the learning process needs to be specified. This is achieved by the compile method. There are three arguments it receives, and this consists of the optimiser, the loss function, and the metrics. The optimiser chosen is *adam*, which is a secure method to implement, has little requirements for memory and efficient computationally. The loss function chosen is the score function of MSE measurement and gives additional metrics used to judge the performance of the model.

Model Architecture

Using the code from Listing 5.6 it displays the summary of the LSTM model architecture by detailing the layers. The completed model is illustrated in Fig. 5.6 and shows the output shape of each layer.

```
Model: "sequential_9"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_9 (LSTM)                (None, 50)                16200
_________________________________________________________________
dropout_4 (Dropout)          (None, 50)                0
_________________________________________________________________
dense_9 (Dense)              (None, 1)                 51
=================================================================
Total params: 16,251
Trainable params: 16,251
Non-trainable params: 0
_________________________________________________________________
```

Figure 5.6: LSTM

Train and Predict

The model uses *numpy* arrays of the input data, and for training the model, it uses the fit function. The Listing 5.7 exhibits the history object returning and providing a summary of the model performance during the process of training. The model is trained with a batch

size of 32 and performing by the specified number of epochs. Epochs represent the total number of iterations the data is run through the optimiser (Basnet, 2016). The training of the data can be varied in duration as it can take seconds to hours depending on the size of the network, training data, choice of epoch and batch. To show that training is occurring it displays a progress bar on the command line for each epoch while detailing the training loss and validation loss. The method of splitting is used for the test and validation data set. This can then be plotted by plotting the training and validation loss at each epoch by using the variable *history* returned by the fit function.

```
1 history = model.fit(X_train, Y_train, epochs=25, batch_size=32,
      validation_data=(X_test, Y_test),
2                     callbacks=[EarlyStopping(monitor='val_loss', patience
      =10)], verbose=1, shuffle=False)
```

Listing 5.7: Python code of fitting the model

Listing 5.8 contains the code that is used to produce the prediction after the model has gone through training. It consists of predicting the variables *x_train* and *x_test* by calling the function *model.predict* and assigning the data into the new variables *train_predict* and *test_predict*. Using the two new variables can then be used to visualise the data in a graph.

```
1 train_predict = model.predict(X_train)
2 test_predict = model.predict(X_test)
```

Listing 5.8: Python code for calling the predict() function on the model

Overfitting and Underfitting

The purpose of models is to generalise data well; this means to give realistic outputs to sets of the input that has not been foreseen. Overfitting and underfitting refer to the model's performance and any deficiencies that may suffer. "A model that generalises well is a model that is neither under fitted nor overfitted"(Al-Masri, 2019).

Overfitting refers to the model training the data set too well and fitting too close to the set of data points, resulting in the prediction to be too good to be true. Overfitting occurs when the model learns the trends and noise in the training data, which can lead the model to learn too much, therefore negatively impacting the model and becoming ineffective for the new data. The model learns the noise and variation in the training data as concepts. These concepts are not applied to the new data and lead the model to be unable to generalise. In conclusion, overfitting results in a decent performance on the training data but poor generalisation towards the other data; this leads to being worse. Underfitting refers to the model being unable to model the training data set and generalise it towards new data; this leads to an unsubstantial model with poor performance on the data. Unlike overfitting, underfitting is more natural to recognise by a performance metric. In conclusion, underfitting results in both a poor performance on the training data and generalisation towards the other data.

A model that is overfitted has a low training error, but a high testing error whereas a model that is under fitted has a high training and testing error. For substantial results in predictions,

the model should be between overfitting and underfitting. It can be hard to accomplish this as it is common for models to be overfitted. There are strategies to help achieve a good model, and this is done by using a validation data set. Validation data set is a subset of the training data that is held back from the learning algorithm until the end of the project. Once the learning algorithms are modulated and tuned on the training data set, the learned models can be evaluated on the validation data set to acquire the performance of the model when it comes to the unseen data. Using cross-validation is sufficient for estimating the model accuracy on unseen data. Other than this, other strategies include training with more data, removing features and early stopping.

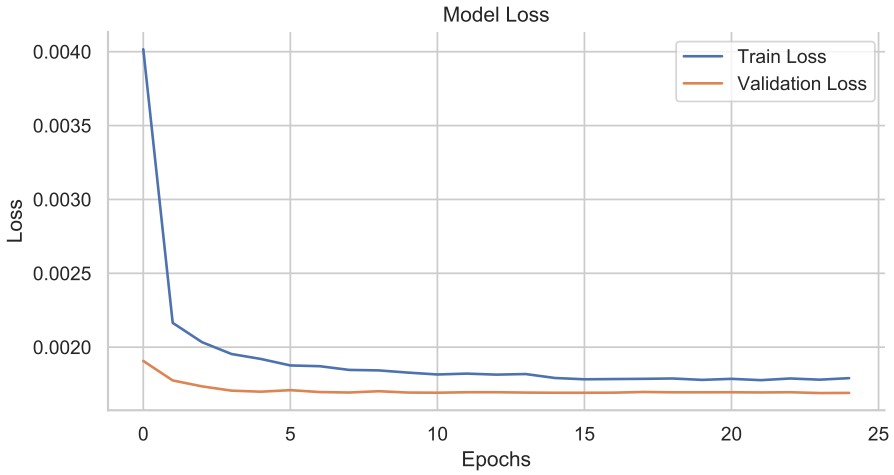


Figure 5.7: Complete data set with 25 epochs

The results from this are illustrated in Fig. 5.7 where it shows the training and validation loss coverage. This model consists of the train size of 8102, test size of 2026. The data results in the train shape of 8071 samples, 1-time step, 30 features and the validate size of 1995 samples. The shape displays the whole data set. From the figure, it shows that it is an under fitted model. It is observed by focusing on the first epoch of the validation loss. In the beginning, the validation loss reduces by a fraction but then has idle progress after three epochs. This is when the model begins to underfit and stays above the validation loss. The training loss at the start starts at a high value and reduces at epoch two nearly reaching to 0. The training loss continues to decrease in loss and nearly reaches the same value as the validation loss by the last epoch. Even though the model presents these results, the training loss value was still low meaning the prediction is reasonable. It does show that it does not need to run for 25 epochs as after epoch 20 the loss is the same.

Observing the model concluded trying to improve it since an underfit model is not beneficial. It is good to adapt by changing hyperparameters, as stated before, to improve the model. Observing the figures reveals how the change of parameters in the model can affect the train and validation loss and the performance of the model when it comes to an overfit or underfit model. Below shows the changes to the model and the outcomes.

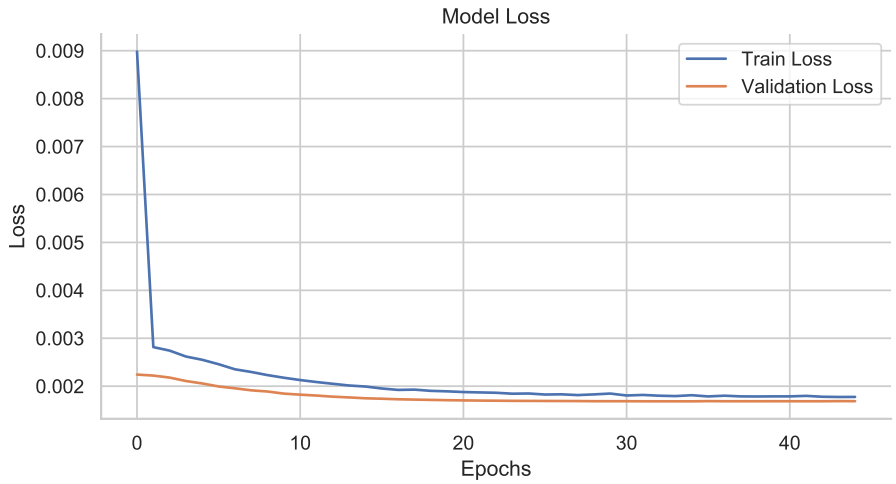


Figure 5.8: Data set using sigmoid activation and dropout layer at 0.1

The change of hyperparameters for this model in Fig. 5.8 consists of including the complete data set running 45 epochs with the dropout layer at 0.1 and specifying *sigmoid* as the activation function. The figure presents the model as an under fitted model. It indicates that this model will not produce a good quality output of values for the prediction. The use of the *sigmoid* layer and dropout layer of 0.1 has changed the results from Fig. 5.7. The model possesses a better fit even though it is classed as an underfit model. The validation error is always smaller than the training error when the model is underfitting the data which needs to be avoided to produce a predicted outcome that is beneficial. Observing the validation loss, no matter the increase of epochs, the model accuracy remains the same. The drawback of this is that with the more significant number of epochs and no accuracy increase, the only action occurring is the inefficacious waste of computation time and resources.

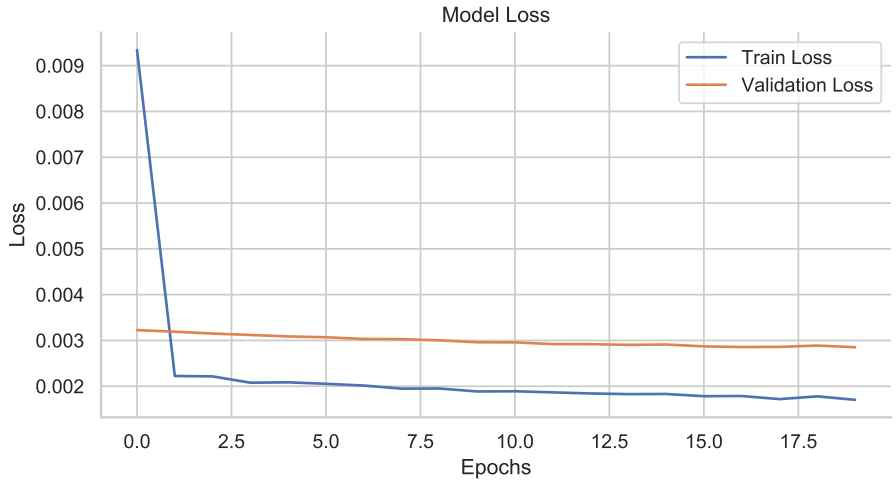


Figure 5.9: 1 month of data with 20 epochs and dropout layer of 0.2

Fig. 5.9 consists of using May as the one month of data running 20 epochs and dropout layer set to 0.2. This model consists of the train size of 1512, test size of 379, resulting in the train shape of 1481 samples, 1-time step, 30 features and the validate size of 348 samples.

The figure presents an overfitted model. The reason for this is because the training loss drops much lower than the validation loss during training. This is observed when the minimum value of the training loss reaches 0.002, and the validation loss stays around the same until the last epoch. There could be a possibility of the loss being able to continue decreasing if the number of epochs ran for longer but could also continue overfitting the data with more significant epochs. Other than this, the results indicate that using a prediction based on one month of data has an outcome of a better prediction than the first model but still would not produce such accurate values.

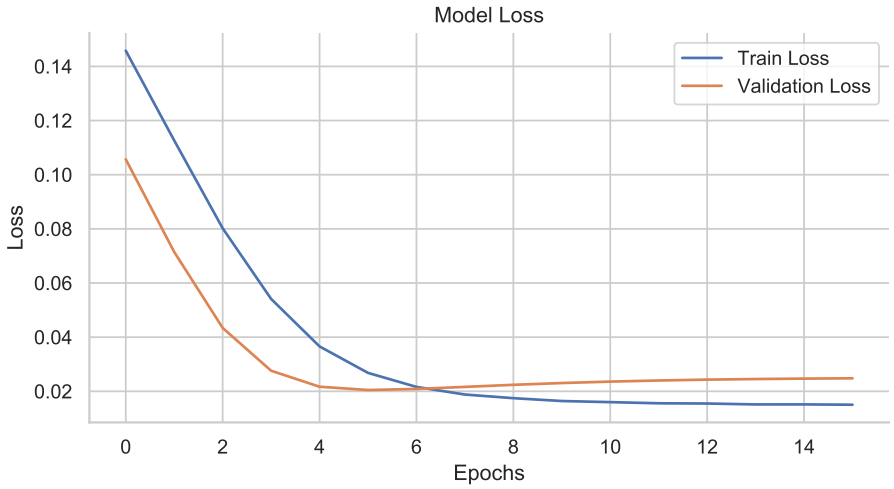


Figure 5.10: 2 weeks of data using sigmoid and dropout layer at 0.5

The last model in Fig. 5.10 uses the dates between January 1st to 14th for two weeks of data. The hyperparameters in this model consist of using the *sigmoid* layer, dropout of 0.2 and running 20 epochs. The number of epochs specified to run was 20, but the model only managed to run 15. This model consists of the train size of 736, test size of 185, resulting in the train shape of 720 samples, 1-time step, 15 features and the validate size of 196 samples. As the model contains two weeks of data, there had to be a decrease in the lookback window because as a reduced amount of data, this is exhibited in the train shape. The figure illustrates an overfit model, from the observation that the training loss increases after epoch six and begins to overfit the data by the more epochs. Looking at the figure indicates that it would be best to stop the training at epoch six as after that amount the model starts to overfit. This does give an understanding of the performance because it does not learn any patterns and therefore, instead, it memorises the training data presented. The validation error is larger than the training error, which needs to be avoided. This is the presented result when overfitting the data occurs.

5.2.3 The Results

Linear Regression

Viewing Fig. 4.39, it displays the prediction for bus 21 by detailing the actual bus delay and the predicted on the Chancellor's Way bus stop through January to May. The figure includes

a key to illustrates what each line represents on the graph. The y axis is the months from January to May using numbers, and the x axis is used to show the value of delay. In some points on the graph, it indicates that the prediction is following the actual bus arrival. For instance, there is a big spike in delay between 0 and 10. The presented result shows that the prediction is following the pattern. This is not the only pattern the algorithm was able to follow as the prediction point at 30 follows the actual delay by being near the same angle and value of delay. Unfortunately, these are the only points where the algorithm was able to achieve a near prediction that is accurate or realistic to the value.

Looking at Fig 4.40 from the results, it displays the actual and predicted delays from using the 31st of May as a data set. This was used to see if by using one day of data would result in a difference compared to using the numerous months stated. The y axis is the hours of the day, and the x axis is used to show the value of delay. Looking at the figure illustrates that the prediction for the delay comes to a conclusion of not being substantial and not fitting with the actual delay. It can be visible by looking at the different points in the figure where it shows the prediction is unable to follow the pattern of the actual data and is very imprecise with showing a substantial prediction for the future.

The two figures, Fig. 4.39 and Fig 4.40 indicates that the algorithm was unable to follow a pattern and produce reasonable results for a prediction of the number 21 bus service on Chancellor's Way. Changing the data structure for each of the models implies that modification and data do have much effect in hindering how well the model performs. Using all the data did produce better results in a prediction, which does conclude that for linear regression it is best for the performance and results to use as much data as possible, so that can recognise any patterns.

Linear regression has been a wide choice for producing predictions such as for stock markets as this model in machine learning can evaluate the precision and bias of predictions. Other than that it can be utilised to predict the arrival of a bus. The results give evidence that linear regression was not the right choice for the implementation of the model and for a time series analysis and prediction. It was difficult to understand how to implement the variables within the algorithm properly and how to cater the algorithm to cope with using time series with various intervals of the bus arriving. From research, there were not many studies that used linear regression to predict future values as the data sets used were mainly timestamps. For instance, this study contains data with purely numeric data stored in the format of timestamps and contained less categorical values. From using this model, it was hard to establish whether there was a significant relationship between the two variables used, scheduled arrival time and delay. It is an essential factor to determine as relationships or correlations between variables is critical to understand if we want to use the values of one of the variables to have the ability to predict the value of another. This was not implemented to the extent where the relationship was comprehensible to understand.

Long Short-Term Memory

LSTM was also utilised to predict the delays of the number 21 bus for the Chancellor's Way bus stop using various data sets to predict the next day.

From Fig. 4.41 it displays the actual data of the months and the prediction. It is illustrated by colour-coordinated key, showcasing the actual data and the prediction. The y axis values

represent the delay in seconds, and the x axis represents the time of the day. The prediction line follows the pattern of the actual data and showcases how the predicted data would appear. For instance, it shows the interval time 125 follows the same spike in the bus arriving earlier than scheduled. From looking at the visualisations of bus 21, it is noticeable that this occurrence is at the Reading Station bus stop. In Fig. 4.42 it appears that the graph looks identical, but there are some differences in how the prediction line is displayed. For instance, at the same time interval of 125, the predicted line does have a slightly earlier arrival time than Fig. 4.41 and at most intervals has a higher delay time, such as the time intervals at 50 and 200. These graphs are from the results of using different hyperparameters in the model.

Fig. 4.43 showcases the actual and predicted delay of the bus service using May as one month for the data set. The data displayed in this figure shows that it does not consist of a drastic time of early arrivals, as shown in figures 4.41 and 4.42. It displays the delays of the bus by 2500 seconds (41 minutes) at the interval time 30. The predicted line at this time predicts that the bus will have an occurrence of delay a fragment after this time by being delayed around 800 seconds (13 minutes), therefore, showing that this model for prediction is not exactly accurate and is far off from the actual delay time of the bus. The model demonstrates inconsistency with the accuracy of the prediction by not being proximate to the actual arrival time of the bus.

Fig. 4.44 showcases the actual and predicted delay of the bus service using two weeks of data in May for the data set. Like Fig. 4.43, this figure displays a prediction that is unsubstantial, but is feasible because the model is based on two weeks of data. This is displayed by the prediction line not following trend with the actual data and is easily noticeable from how the prediction line is correlating with the actual line. The outcome additionally indicates that it is best to use more data in the future for more accurate predictions.

Even though there were models that resulted in being incapable when it came to the accuracy of a prediction, using LSTM was beneficial in trying to accomplish a prediction for a bus compared to linear regression. The research on this neural network allowed the understanding of why it is widely used for time series forecasting as it adds excellent benefits where regression, such as linear, can find difficulty in adapting. The choice of using LSTM over the other RNN was because it appeared to be the right model when it came to the format of the data and from previous research producing results with somewhat accurate predictions. LSTM allows the flexibility of being in reasonable control over various parameters of the time series. The flexibility is extensive. For instance, it gives the ability to have various relationships in models from the combinations of sequence to sequence models. Furthermore, changing the size of the lookback window to predict the next step, this was very beneficial when changing the amount of data to use in training. Therefore, from the results, LSTM is the better fit for the given problem statement.

5.3 Performance Metrics for Analysis

5.3.1 Mean Squared Error

It is necessary to evaluate the performance of each of the models in terms of prediction accuracy. Performance can be regulated by using Mean Squared Error (MSE) as this measures

the models' performance, representing the average squared difference between the observed value(scheduled arrival time) and the predicted value (predicted arrival/delay time at the bus stop). Here it is used for the evaluation of the linear regression, and LSTM models. The formula is formalised in Eq. 5.1.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^{2} \tag{5.1}$$

From Eq. 5.1, Y_i is the i_{th} observed value and $\hat{Y}_i$ is the i_{th} predicted value.

MSE is detailed by the lower the MSE, the higher the accuracy of prediction, and the larger the MSE, the larger it is in errors and lower the accuracy in the model. It squares each difference between the observed and predicted so that the positive and negative do not cancel each other out. Even though some may assume otherwise, there is no correct value for MSE, as if the MSE had an outcome of 0 it would determine that the model predicts the training data flawlessly which would cause the unlikeliness of being capable of predicting any other data perfectly. Therefore, too low can result in being over-refinement.

5.3.2 Coefficient Determination

Coefficient determination denoted by r^2 can be applied in linear regression as a metric for performance. This is a key output for regression analysis, as it is interpreted as the variance in the dependent and independent variable. The coefficient determination is the square correlation (r) between the predicted and actual scores by measuring the durability and direction of the linear relationship between these two variables. A definition of this is "(total variance explained by model) / total variance" Rowe (2018). The correlation (r) is ranged from 0 to 1. 1 indicates the model being of a perfect fit and 0 indicating no linear relationship and a low level of correlation within the model, this outcome is not always the case. Given an example, if the r^2 score were of a value of 0.60, it would mean that the independent variable predicts 60% of the dependent variable. The coefficient determination (R^2) for linear regression with one independent variable is formulated in Eq. 5.2 illustrated by (Yau, 2020).

$$r^2 = \frac{\Sigma\left(\hat{y}_i - \bar{y}\right)^{2}}{\Sigma\left(y_i - \bar{y}\right)^{2}} \tag{5.2}$$

From the 5.2 y_i is the i_th observed value of the dependent variable, $\bar{y}$ is the mean and $\hat{Y}_i$ is the i_th fitted value.

5.3.3 Model Comparison

This section shows the model comparison within linear regression and LSTM models using the metrics described above. Within the two methods of prediction, it includes different parameters to the models which produced different outcomes in prediction and measurement of performance.

Model	MSE	r2 Score
Using complete data set	1.53	-0.4811552679698152
Using 31st May as data set	2.52	-4.207106773473997

Table 5.3: Linear regression model performance

Linear Regression

From Table 5.3 there was the inclusion of two models for linear regression. Looking at the first model, which consists of using all the months in the data set, it results in the MSE value of 1.53 and a r^2 score of -0.48 to two decimal place. Firstly, the results for the MSE metric indicates that the model is low in error because it is adjacent to 0. Even though this is the case, the r^2 score shows a negative linear relationship which indicates that the model is poorly defined and does not follow the trend of the data. It reveals that the model fits the data poorly. The negative value occurred by the discovery of using a time series can lead to this outcome IrishStat (2011). This is the case for the model consisting of using the 31st of May as the data set. The results for this model includes the MSE value of 2.52 and the r^2 score to be -4.21 to two decimal place. The MSE value is slightly higher compared to the first model, the reason for this may be because there is a lot less data for the model to utilise for training and to grasp the patterns to make a proper prediction fully. This proves that only using a day for a prediction is not substantial enough as it being further away from 0 designates that there are more errors in this model to the first one. Other than the MSE having a more mediocre performance on this model, the r^2 also scores poorly. The r^2 score is higher by having a negative number further from 0 showing that the linear relationship is also negative and does not contain any correlation with the two variables.

Long Short-Term Memory

Table 5.3 summarises the MSE results for four prediction models performed by the chosen hyperparameters in each model to compare how well each model performed using the train and test variables.

Model Hyperparameters	MSE Train	MSE Test
Base model	229.58	222.27
Sigmoid activation and dropout layer at 0.1	224.78	222.06
Dropout layer at 0.2	211.98	283.23
Sigmoid activation and dropout layer at 0.5	227.60	294.22

Table 5.4: Long short term memory model performance

The first is the base model which consists of five months of data, dropout of 0.2, dense of 1, and no activation layer. The results from this model include the MSE train being 229.58 and the MSE test being 222.27. These values are very high when it comes to the MSE, which indicates that the model has many errors and is low with accuracy—using these metrics with the train and test variables further knowledge of whether the model is overfitting or underfitting. It can be distinguished of overfitting if the test MSE is higher than the train MSE and underfitting if the test MSE is lower than the train MSE. The results clarify that

this is an underfit model from the train being higher than the test results. Unfortunately, this is the same case for the second model that includes the use of the *sigmoid* activation and the dropout layer of 0.1. The results for this model is the MSE train being at 224.78 and the MSE test being at 222.06. Even though it is a underfit model, it does reveal that its performance was more significant than the base model because the train and test values being closer in value. The model could have then been at the same value when performing, constructing it to produce a reliable prediction.

The third model consists of using one month of data with the hyperparameters, including the dropout layer at 0.2 and no activation layer. The results for this model signify the MSE train is at 211.98, and the MSE test is at 283.23. The difference between the train and test values is very high, with a difference of 72. From the results, it exhibited that the model is overfitted, which is better than the previous models but still not good enough. This model has the lowest MSE train value, which might indicate that it does perform imperceptibly better than the others even though it is not hugely substantial. Like the third model, the last model persists to being overfitted. This model has hyperparameters that consist of using the *sigmoid* activation and a dropout layer at 0.5 while using data of 2 weeks. With this model, it produces the highest value for the MSE test and the highest value for the MSE train. Overall it indicates that using two weeks of data is not concrete enough for a prediction and would not endure in the future. Therefore, the last model is ineffectual in comparison to the other model

The table shows that all the models are very high in error; this concludes to knowing that the models are not accurate in a prediction and not very substantial as it is aspired to be. Therefore, it would be could to decide how much data to use next time and what hyperparameters would corporate best amidst the results.

5.4 Summary

In this chapter of discussion and analysis, the models within linear regression and LSTM were analysed and statistically tested. The LSTM models gave better results visually than the linear regression model across the various models in terms of the accuracy of the prediction. It is discovered that the MSE is very high when using LSTM in comparison to the models using linear regression, but when looking at the graphs for visualisation, the LSTM models appear to have a better prediction. Other than this, it is perceived that the implementation using the prediction methodologies could be implemented more efficiently by ensuring the variables used for both linear regression and LSTM are substantial and contain a relationship. In the next chapter, it will discuss findings and issues related to the bus arrival time under conclusions and recommendations.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This paper presents a bus arrival prediction method that transpired based on the proposed machine learning methods, linear regression and LSTM by analysing the live data of Reading buses transport times. It presents the different machine learning approaches for times series forecasting. Using these methodologies demonstrates that the obtained aim was achieved by applying data science tools and techniques to predict the arrivals and delay of Reading bus services. The required data for this proposed system was collected from the output of the AVL system used for public transport within the Reading Bus Company and widely used in other various cities. The total of 6 buses was used for analysis within the months from January to May and focused on using the scheduled arrival time and the delay. The data set was classified into two time periods; this consisted of the weekday and weekend so that there was an understanding of the congestion and different travel patterns. From the chapter, Results, it shows that the bus stop that stood out in all buses when coming to being on schedule transpired to be the number 3 and 5 bus. The number 3 bus showed that during the time between January and May, it was mostly on time and when it was late, the delay was not of a high quantity.
Additionally, it produced the earliest arrival times compared to the arrival schedule. The number 5 bus concluded having a minimum duration of delay by having a maximum delay of 5 minutes. Even though it is not on schedule, the small quantity of time for waiting is more salutary in performance than the other buses when they are delayed. Here presents that as a university student, it is best to rely on these two bus service when commuting by public transport.

The study consisted of six different prediction models that were developed and tested using the stated methodologies. This involved linear regression containing two models and LSTM containing four models. These models were tested on the route of the number 21 bus service.

The proposed system used a linear regression model that was able to capture any patterns within the variables in the arrival and delay times. The model could overview any patterns that persisted so that it was able to learn from it. From the results, some elements of the linear regression remained moderately accurate to the real data presented. The outcome of this methodology approach attested that for an improved prediction, there could be more further data to base it off compared to the periods used currently. The results showed that using

linear regression for this variety of prediction did not produce an accurate outcome. Changing the quantity of data that the model could learn from did not produce many changes when it came to accuracy. This was mentioned in the statistical measurement of using MSE to detail the errors in the model and using coefficient determination. It was ascertained that the linear regression models gave the smallest prediction errors in terms of prediction accuracy meaning that the model was not high in errors but also persisted in showing that the variability of the variables did not have any correlation of relationship to one another, therefore not resulting to a good fit.

The proposed system also used a neural network model consisting of LSTM layers that were also able to capture correlations of variables in the bus arrival and delay times. This allowed the model to generalise any patterns learned in the predictions across the months. The approach allowed predictions further into the future. The increased accuracy, when compared to the baseline model, is also better when the peak hours of the bus were present and under the stress of being on schedule. The implementation of various models was performed and occurred by relying on the techniques of using the look-back method that is used in the machine learning community when it comes to producing predictive models. There was a naivety understanding of how LSTM algorithm operates when looking back at the results at the start of the study. The conclusion of the predictions based on the first LSTM model, in 5.7, showed that the original LSTM model was under fitted and from the statistical measures detailed the same outcome. By observing this outcome of underfitting, the model architecture consisted of changing the hyperparameter was essential to improve the model, but this led to the conclusion of overfitting the model. This conclusion could have been lead by the hyperparameter not being optimised correctly in line with the data as tuning the hyperparameters can be challenging. Using this methodology showcased that one month of data was the ideal approach to go to produce a sufficient prediction.

The performance of evaluation of the linear regression and LSTM occurred by comparing all models using conventional and statistic measures between one another. The results perceived presented came to the verdict that both models developed were not particularly useful and producing results that lead to insufficient accuracy. The prediction results from both methodologies attested that predicting arrival and delay times is a time series problem than a regression problem. The use of time series approaches can produce better results compared to regression methods. Time-series methods such as LSTM assume that the patterns from the data can be remembered and repeated in the future for a further accurate prediction. The outcome of the study reveals that if it were implemented again, the use of time series method would be the ideal approach.

6.2 Future Work

As for future work, I would like to extend the presented system based on the identified findings of the study to proceed in the directions of these following recommendations:

Part of the future work should include focusing on collecting more data. The data would contain a year of real-time for the studied buses so that patters in the data are more apparent to discover and train in the model. Extending the period of the data is, therefore, a necessary process as it would later provide a build of a model that is a strong predictor. Collecting a year of data can be time-consuming; therefore, there would be the use of collecting data in an

automated method. This would make the methodology of data collection more manageable and reduce the time required to collect data. The process consists of pulling data from the API automatically using the already written python script to collect the data and Microsoft Windows Task Scheduler Application. This application enables the ability to set a schedule for a program or in this case, the required script.

As the LSTM model is the best approach for time series data, the focus of future work would also include improving the model. It would include optimising the hyperparameters to deal with the problem of underfitting and overfitting the data. There is proposed and developed hybrid algorithms to improve prediction accuracy that would benefit from such models Altinkaya and Zontul (2013). Therefore, it is recommended to study other factors that contribute to the variation of the arrival time. This would be acclimated in as variables in the model. It would be great to aim to develop this approach by adding more features of real-time data. This would consist of data from an API that provides data for the weather, the duration and frequency of traffic signals that would include temporary lights as this could affect the travel time of buses, persisted road works and the duration of passengers boarding and disembarking. In addition, the inclusion of using the ID of vehicles as a variable as this can be an intriguing factor in the prediction. Anomalous may transpire because of technical problems of the bus, such as it breaking down, this then can be a recurrence with certain buses. A classifier could be able to identify the buses that are likely to have technical problems from specific patterns included in the data. Further investigation could adhere and include identifying the performance of each driver on their routes. All these variables can, therefore, improve the model further and deliver a more accurate and reliable prediction of the arrival times during the year.

Even though improving the already developed models, there is the opportunity to implement the data on a different methodology in machine learning. An example of this is using SVM. This method for performing predictions is widely used in the machine learning community and has been utilised for predicting bus times before. This has been presented in Yang et al. (2016) and Osman et al. (2020). Both these researchers used SVM with genetic algorithm and culminated that the model performed better than ANN models. This would then be a future study to conduct with the collected data and would be interesting to implement and analyse any differences of results compared to using LSTM or linear regression. Other than this, another model such as ARIMA could be implemented. ARIMA is a simple method to fit time series data to predict future points, therefore forecasting. ARIMA has been used before in the case of predicting bus times and has been researched and demonstrated in Napiah and Kamaruddin (2010).

Developing a web application will allow exercising the study even further. The development of the web application as an implementation of the prediction can support utilising the data alongside the live bus travel times presented at the bus stops. The features of the web application would require the capability to exhibit the given bus to display the predicted arrival time so that the user can view it. The data would be presented as a timetable so that it is straightforward and visible to the user. This would help to achieve the ability to allow users to search for a specific bus and history of its performance and predictability of arrival times. For this to be possible, the implementation of the recommended future work beforehand must be completed first.

Chapter 7

Reflection

The work I have done has helped me to understand the concepts of data science further. I started this project uncertain and not confident in the future work that was required because it was a new subject for me. I can now unequivocally say that I am confident in further using aspects of data science tools and techniques to implement in future projects. From this, in the future, I would love to teach myself and accumulate more knowledge about data science to produce projects in my spare time, such as developing an Artificial Intelligence assistant. Data science is a topic that is becoming more popular by creative developers and in work environments. The project has left an impression on my future choice as it has impelled me to seek employment in this sector. I have learnt most aspects of data science; this was aided from the Artificial Intelligence module, that helped me to understand machine learning more from the different types of learning to various machine learning networks, such as RNN. The results have lead me to aspire to have a more in-depth knowledge of using linear regression and LSTM for predictions and to search for more examples of implementing this method for time series data.

I believe from the project I worked well with implementing the models, but I do believe that from more extensive research and examples of machine learning for time series data I would have the capability to implement a reliable predictor model. Reaching to this outcome has driven me to practise coding skills thoroughly to accomplish an accurate model that I would be valuable and used in a real-life situation for the bus in Reading. The entire duration of the project, I managed to be organised by ensuring that I was on course with the Gantt chart developed in the PID. The project was managed and achieved by the recurrent meetings that occurred with my supervisor. This helped me with any questions I had and confirmed if I was on track with the development of the project and within the report.
Due to the global circumstances of COVID-19 circulating since January of 2020, it has resulted in disruption in day to day life. This caused the University of Reading to conclude closing the campuses, therefore cancelling the individual meetings with supervisors. The Government instructed strict rules to uphold social distancing measures to prioritise the populations' health. Therefore, meeting every week with individuals outside of a household was prohibited. Even though this was an unfortunate result, I still had the capability of recurrent weekly discussions regarding the report via Zoom. This was beneficial in reaching my potential within my report to assure that I produce a compelling report and that I can accomplish the highest marks possible. Throughout the entirety of this project, the resources and extensive amount of work, I was very interested in composing what I possessed and created within the findings

when it came to the buses in Reading. As someone that commutes via bus, it was interesting to know how the buses in Reading operate on a day to day basis and obtaining information about which bus was the least reliable.

What I could have done better was to understand different machine learning methods more in-depth so that I would be able to comprehend which model is more beneficial for a time series data. Understanding the methods would have enabled me to develop a more suitable model. This could be done by researching more information about the different activations and have more knowledge of adding layers and what they can do.

References

Abkowitz, M., Slavin, H., Waksman, R., Englisher, L. S., Wilson, N. H. et al. (1978), Transit service reliability, Technical report, United States. Urban Mass Transportation Administration.

Agafonov, A. and Yumaganov, A. (2019), 'Bus arrival time prediction using recurrent neural network with lstm architecture', *Optical Memory and Neural Networks* **28**, 222–230.

Al-Masri, A. (2019), 'What Are Overfitting and Underfitting in Machine Learning?'.
URL: *https://towardsdatascience.com/what-are-overfitting-and-underfitting-in-machine-learning-a96b30864690 (accessed on 23 Apr. 2020)*

Altinkaya, M. and Zontul, M. (2013), 'Urban bus arrival time prediction: A review of computational models', *International Journal of Recent Technology and Engineering (IJRTE)* **2**(4), 164–169.

Basnet, B. (2016), 'Lstm Epoch Size Choice'.
URL: *https://deepdatascience.wordpress.com/2016/11/18/lstm-epoch-size/ (accessed on 21 Apr. 2020)*

Brown, A. and Wilson, G. (2012), *The architecture of open source applications, volume ii*, Vol. 2, Lulu. com.

Chien, S. I.-J., Ding, Y. and Wei, C. (2002), 'Dynamic bus arrival time prediction with artificial neural networks', *Journal of Transportation Engineering* **128**(5), 429–438.

Council, R. B. (2011), 'Local transport plan 3: Strategy 2011-2026'.

Gurmu, Z. K. and Fan, W. D. (2014), 'Artificial neural network travel time prediction model for buses using only gps data', *Journal of Public Transportation* **17**(2), 3.

Hochreiter, S. and Schmidhuber, J. (1997), 'Long short-term memory', *Neural computation* **9**(8), 1735–1780.

IrishStat (2011), 'When is r squared negative?'.
URL: *https://stats.stackexchange.com/q/12905 (accessed on 27 Apr. 2020)*

Jeong, R. and Rilett, R. (2004), Bus arrival time prediction using artificial neural network model, *in* 'Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749)', IEEE, pp. 988–993.

Johar, A., Jain, S. and Garg, P. (2016), 'Prediction of bus travel time using ann: A case study in delhi', *Transportation Research Procedia* **17**.

Keras (2020), 'Keras: The python deep learning library'.
URL: *https://keras.io/ (accessed on 21 Apr. 2020)*

Kumar, V., Kumar, B. A., Vanajakshi, L. and Subramanian, S. C. (2014), Comparison of model based and machine learning approaches for bus arrival time prediction, *in* 'Proceedings of the 93rd Annual Meeting', Transportation Research Board, pp. 14–2518.

Napiah, M. and Kamaruddin, I. (2010), 'Arima models for bus travel time prediction', *Journal of the Institution of Engineers Malaysia* **71**.

Ojha, V. K., Dutta, P., Saha, H. and Ghosh, S. (2012), Linear regression based statistical approach for detecting proportion of component gases in manhole gas mixture, *in* '2012 1st International Symposium on Physics and Technology of Sensors (ISPTS-1)', IEEE, pp. 17–20.

Olah, C. (2015), 'Understanding Lstm Networks'.
URL: *http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed on 31 Mar. 2020)*

Osman, A., Mohd Hashim, S., Anwar, T. and Ahmed, A. (2020), *A Robust Hybrid Model Based on Kalman-SVM for Bus Arrival Time Prediction*, pp. 511–519.

Patnaik, J., Chien, S. and Bladikas, A. (2004), 'Estimation of bus arrival times using apc data', *Journal of public transportation* **7**(1), 1.

ReadingBuses (2020), 'Reading Bus Network'.
URL: *https://www.reading-buses.co.uk/services (accessed on 30 Mar. 2020)*

Rowe, W. (2018), 'Mean Squared Error & R2 Score Clearly Explained'.
URL: *http://www.r-tutor.com/elementary-statistics/simple-linear-regression/coefficient-determination (accessed on 27 Apr. 2020)*

Seber, G. A. and Lee, A. J. (2012), *Linear regression analysis*, Vol. 329, John Wiley & Sons.

Shalaby, A. and Farhan, A. (2004), 'Prediction model of bus arrival and departure times using avl and apc data', *Journal of Public Transportation* **7**(1), 3.

Strathman, J. G., Dueker, K. J., Kimpel, T., Gerhart, R., Turner, K., Taylor, P., Callas, S., Griffin, D. and Hopper, J. (1999), 'Automated bus dispatching, operations control, and service reliability: Baseline analysis', *Transportation Research Record* **1666**(1), 28–36.

Turnquist, M. A. and Blume, S. W. (1980), 'Evaluating potential effectiveness of headway control strategies for transit systems', *Transportation Research Record* **746**(1), 25–29.

Yang, M., Chen, C., Wang, L., Yan, X. and Zhou, L. (2016), 'Bus arrival time prediction using support vector machine with genetic algorithm', *Neural Network World* **26**, 205–217.

Yau, C. (2020), 'Coefficient of Determination'.
URL: *http://www.r-tutor.com/elementary-statistics/simple-linear-regression/coefficient-determination (accessed on 27 Apr. 2020)*

Appendix A

Linear Regression model in Python

```
import seaborn as seabornInstance
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
%matplotlib inline
from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters()
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score
from sklearn import tree

# Bus 21 data set for chancellor bus stop
bus_21p = bus_21
bus_21p.head()

# Describe data set
bus_21p.describe()

# Plot distribution graph
plt.figure(figsize=(8,4))
plt.tight_layout()
seabornInstance.distplot(bus_21p['delay'])
sns.despine()
sns.set_context("paper")
sns.set(style="whitegrid")
plt.title('Distribution of Delay')
plt.xlabel('Delay (s)')
#plt.savefig('1day_dist.pdf', bbox_inches='tight', transparent='true')

# Reshape scheduled arrival time and delay
x = bus_21p['ScheduledArrivalTime'].values.reshape(-1, 1)
y = bus_21p['delay'].values.reshape(-1, 1)
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3,
    random_state=0)
print (x_train.shape, y_train.shape)
print (x_test.shape, y_test.shape)

x_train

# Scale x and y variables
scaler = StandardScaler()
```

```
scaler.fit(x)
x = scaler.transform(x)
scaler.fit(y)
y = scaler.transform(y)
y

# Train the algorithm
regressor = LinearRegression()
regressor.fit(x_train, y_train)
y_pred = regressor.predict(x_test)

# Metrics for model
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred)) #
    MSE
print('Intercept: \n', regressor.intercept_) # To retrieve the intercept
print('Coefficients: \n', regressor.coef_) # For retrieving the slope
r2_score(y_test, y_pred) # R2 score

clf = tree.DecisionTreeRegressor()
clf = clf.fit(x_train, y_train)
y_pred = clf.predict(x_test)

# Print actual and predicted values
bus_21pc = pd.DataFrame({'Actual': y_test.flatten(), 'Predicted': y_pred.
    flatten()})
bus_21pc.head(15)

# Plot actual and predicted values
bus_21pc1 = bus_21pc.head(50)
bus_21pc1.plot(kind='line',figsize=(14,8))
sns.despine()
sns.set_context("paper")
sns.set(style="whitegrid")
plt.title('Actual vs Predicted')
plt.ylabel('Delay (s)')
#plt.savefig('linpred1day.pdf', bbox_inches='tight', transparent='true')
plt.show()
```

Listing A.1: Linear Regression model

Appendix B

Long Short-Term Memory model in Python

```
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
from keras.layers import *
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from keras.callbacks import EarlyStopping

# Data set for bus 21 chancellor bus stop
df_21 = pd.read_csv('LSTM.csv')

# Specify dates that will be in data set
date_after = pd.Timestamp(2018, 6, 1, 0)
date_after
df_21 = df_21.loc[df_21['ScheduledArrivalTime'] < date_after]

# Print head and tail of new data set
df_21.head()
df_21.tail()

dataset = df_21.delay.values #numpy.ndarray
#dataset = dataset.astype('float32')
# Reshape and scale dataset
dataset = np.reshape(dataset, (-1, 1))
scaler = MinMaxScaler(feature_range=(0, 1))
dataset = scaler.fit_transform(dataset)
train_size = int(len(dataset) * 0.80)
test_size = len(dataset) - train_size
train, test = dataset[0:train_size,:], dataset[train_size:len(dataset),:]
print(len(train), len(test))

def create_dataset(dataset, look_back=1):
    X, Y = [], []
    for i in range(len(dataset)-look_back-1):
        a = dataset[i:(i+look_back), 0]
        X.append(a)
```

```
        Y.append(dataset[i + look_back, 0])
    return np.array(X), np.array(Y)

look_back = 30
X_train, Y_train = create_dataset(train, look_back)
X_test, Y_test = create_dataset(test, look_back)

# Reshape input to be [samples, time steps, features]
X_train = np.reshape(X_train, (X_train.shape[0], 1, X_train.shape[1]))
X_test = np.reshape(X_test, (X_test.shape[0], 1, X_test.shape[1]))
X_train.shape

# Model of lstm
model = Sequential()
model.add(LSTM(50, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
#model.add(Activation('sigmoid'))
model.compile(loss='mean_squared_error', optimizer='adam')

history = model.fit(X_train, Y_train, epochs=25, batch_size=32,
    validation_data=(X_test, Y_test),
                    callbacks=[EarlyStopping(monitor='val_loss', patience
    =10)], verbose=1, shuffle=False)

model.summary()

# Predict
train_predict = model.predict(X_train)
test_predict = model.predict(X_test)
# Invert predictions
train_predict = scaler.inverse_transform(train_predict)
Y_train = scaler.inverse_transform([Y_train])
test_predict = scaler.inverse_transform(test_predict)
Y_test = scaler.inverse_transform([Y_test])

# Print mse of train and test prediction
print('Train MSE:',np.sqrt(mean_squared_error(Y_train[0], train_predict
    [:,0])))
print('Test MSE:',np.sqrt(mean_squared_error(Y_test[0], test_predict[:,0])
    ))

# Print actual and predicted values
dataset = pd.DataFrame({'Actual': Y_test.flatten(), 'Predicted':
    test_predict.flatten()})
dataset.head(10)

# Plot train and validation loss
plt.figure(figsize=(8,4))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
sns.despine()
sns.set_context("paper")
sns.set(style="whitegrid")
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(loc='upper right')
#plt.savefig('losscomsig01.pdf', bbox_inches='tight', transparent='true')
plt.show();

```

```
# Plot actual and predicted values - first 200 rows
aa=[x for x in range(200)]
plt.figure(figsize=(8,4))
plt.plot(aa, Y_test[0][:200], marker='.', label="actual")
plt.plot(aa, test_predict[:,0][:200], 'r', label="prediction")
plt.tight_layout()
sns.despine()
sns.set_context("paper")
sns.set(style="whitegrid")
plt.subplots_adjust(left=0.07)
plt.ylabel('Delay', size=15)
plt.xlabel('Time', size=15)
plt.legend(fontsize=15)
#plt.savefig('predcomsig01.pdf', bbox_inches='tight', transparent='true')
plt.show();
```

Listing B.1: Long Short-Term Memomry model