BI

WISEflow
Europe/Oslo(CEST)
30 Jun 2022

Handelshøyskolen BI

GRA 19703 Master Thesis

Thesis Master of Science 100% - W

Predefinert informasjon

Startdato:	16-01-2022 09:00	**Termin:**	202210
Sluttdato:	01-07-2022 12:00	**Vurderingsform:**	Norsk 6-trinns skala (A-F)
Eksamensform:	T		
Flowkode:	202210||10936||IN00||W||T		
Intern sensor:	(Anonymisert)		

Deltaker

Navn:	Nithun Manoharan og Philip Anthony Hauger

Informasjon fra deltaker

Tittel *:	Analysis of the Demand for Bike Sharing in Oslo with Machine Learning
Navn på veileder *:	Wei-Ting Yang

Inneholder besvarelsen konfidensielt materiale?:	Nei	**Kan besvarelsen offentliggjøres?:**	Ja

Gruppe

Gruppenavn:	(Anonymisert)
Gruppenummer:	79
Andre medlemmer i gruppen:	

BI Norwegian Business School -

GRA1974 - Master Thesis

Analysis of the Demand for Bike Sharing in Oslo with Machine Learning

Philip Hauger and Nithun Manoharan

Supervisor:

Wei-Ting Yang

Campus:

BI Oslo

Programme:

Master of Science in Business Analytics

June 30, 2022

Abstract

This paper aims to create accurate predictive models for the bike-sharing system operated by Oslo City Bikes. The three different machine learning methods are used to predict user demand within a specified area of Oslo. Furthermore, the paper intends to discover which factors that most influence bike-sharing usage pre- to mid-pandemic. Different factors of bike-sharing systems will be evaluated to create a reliable model. Recommendations for further research topics, as well as possible business implementations for the model will be explained. The machine learning method with the best performance was GRU with an MAE score of 14.30, RMSE of 20.80 and R^2 of 0.77. Multiple COVID-19 features indicating varying intensities of lockdown were tested, however they did not have as much of an effect as expected.

Table of Contents

List of Figures

List of Tables

Abbreviations

GRU – Gated Recurrent Unit

LSTM – Long-Short Term Memory

MAE – Mean Absolute Error

NN – Neural Network

OCB – Oslo City Bike

R^2 – R-Squared

RF – Random Forest

RMSE – Root Mean Squared Error

RNN – Recurrent Neural Network

1 Introduction

Bike-sharing systems (BSS) is a sharing service that provides the public with short-term bike rentals, usually within a municipality. In order to rent the bikes, a fee or a subscription is required. They also tend to have docking stations where users can pick up and return the bikes. The municipality can choose the model of operation, ranging from governmentally run to privately run.

Many cities across the globe have a BSS. BSS tend work complementary with public transport. Bike-sharing is considered a public good, reasons being that it reduces car usage and has an added health benefit related to cycling. In addition, it improves connectivity to other modes of transportation by reducing travel time and providing a solution to the first/last-mile problem (Du et al., 2019). With the occurrence of a pandemic, many businesses and forms of transportation were negatively impacted. The BSS was one of these businesses that saw an impact due to the pandemic. The pandemic led to stay at home orders which significantly decreased the amount of people who commuted to work, school, or social gatherings. It also led to a decrease in the overall bike usage for BSS. Given the rapidly changing environment, to better allocate resources and design strategies to better suit current needs, it is important to analyse the factors that influence bike usage. The demands of the Oslo bike-sharing system, Oslo City Bikes (OCB), will be the focus of this thesis. The proposed predictive model can be further be applied to other systems, which will be discussed in the last chapter.

OCB is a station-based system located in the city of Oslo, Norway, and operates around the city centre. Demand prediction is particularly relevant for station-based systems, as they require an adequate supply of bikes at each station. The research question is to find the factors that influence the demand for bikes in Oslo. In this study, Grünerløkka, the area with the greatest demand, will be investigated first. Many factors that have been studied in the existing literature are included in the proposed model, such as temporal factors and weather conditions. In this paper, some novel factors related to the pandemic are considered and their potential contributions to model accuracy and robustness are analysed. Based on several data sources, a machine learning model is built to identify the most influential factors and predict future demand. With the proposed model, OCB may benefit by allocating resources based on more accurate demand forecasts. The influencing factors found in this model can also be used as a reference for the design of pricing strategies, hoping to further increase the utilization rate of city bikes.

BSS demand prediction is a commonly researched topic with many different articles covering it. Recent research has found that machine learning methods have high efficiency when predicting demand in comparison to traditional forecasting models. Demand forecasting models have multiple approaches within the field of BSS. Most of the research share a common thread, where they focus on station specific levels for their forecasting. This might lose some factors of influence from neighbouring stations.

With the introduction of COVID-19 in early 2020, there has not been much BSS research on how pandemics effect bike demand. With such, it would be of interest for the city to see how the pandemic influences their BSS and changes in user behaviour. Public mobility was in large part effected by measures taken by the various countries. Norway did not have as restrictive of a lockdown as some other countries, and this led to BSS usage throughout the pandemic. The constant usage of bikes in Oslo leads to an opportunity to see how different stages of lockdown influence the BSS.

This paper creates models by using previous techniques used in BSS research. Three machine learning models are chosen due to their high performance when compared with other models and approaches. The features used are proven to have high importance for BSS demand prediction. The intention is to test how different COVID-19 measures compare to these features for BSS demand prediction.

This thesis is structured as follows. In Chapter 2, the background of bike-sharing systems and Oslo City Bikes is described. Chapter 3 reviews the related literature, including bike-sharing demand and machine learning models for demand forecasting. Chapter 4 covers data preparation and modelling. Chapter 5 presents the results of the predictive models and performance evaluation. In Chapter 6, the potential benefits of business implementation for the model are explained. The conclusions and future work are summarized in Chapter 7.

2 Background

DeMaio (2009) states that there have been three generations of bike-sharing and describes a fourth and future generation. The first generation of a BSS was introduced in the 1960s, where bikes could be used for free and placed in random locations in the city. However, the system failed when bikes were thrown in the canals or stolen for private use. The second generation brought better bikes revenue generation in the form of advertising and coin deposit to unlock them. Despite the improvements, the second-generation system still struggled with the earlier problem of bike theft. The third generation used advanced technology, combined with policy incentives which led to rapid growth of the bike-sharing system (Garcia-Gutierrez et al., 2014). Smartcards, fobs, or mobile phones could keep track of user profiles, and additional safety features were implemented with electronically locking racks being one of them. As more advanced technology improved bike security and user-experience, the concept grew in popularity in the 2000s (DeMaio, 2009).

Among different BSS, the station-less bike-sharing, also called free-floating bike-sharing, is popular because of its ease-of-access for users. This becomes more viable as a system due to the improved technology used to avoid misappropriation of the bikes when the system was first introduced. However, this system could lead to bikes being dropped off in inconvenient places for the public, which is why station-based systems are sometimes preferred. There is a trade-off between the flexibility and cost-effectiveness of a free-floating BSS versus the predictability and running costs of stations of a station-based system. One major disadvantage of a station-based system is that the company running the system must balance the inventory at each station to meet demand.

DeMaio (2009) summarized different BSS models, from public to private, as shown in Figure 2.1. There are benefits and disadvantages to each system. The "Advertising Company" is most relevant to our case as it is the one used in Oslo. This BSS is feasible thanks to a beneficial arrangement between the advertising company and the municipality. The advertising company can arrange favourable terms with the city public advertising space, in exchange for running the BSS.

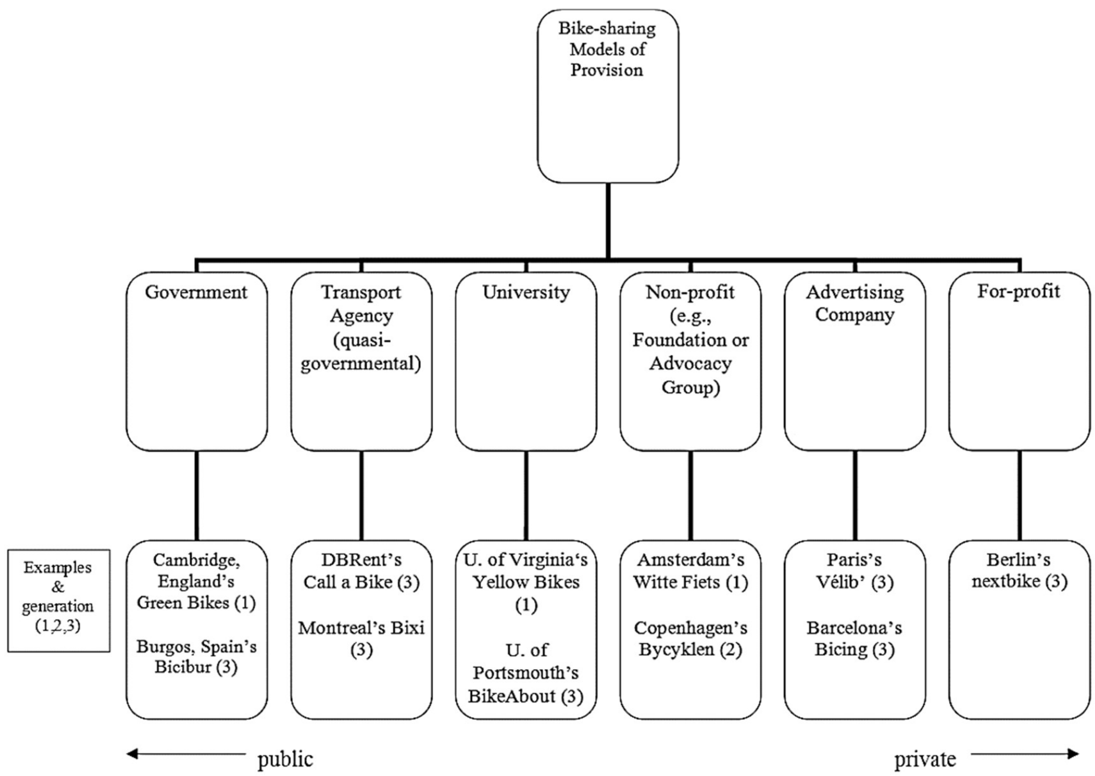

Figure 2.1: Models of Provision by DeMaio (2009)

2.1 Oslo City Bikes

The city of Oslo and Clear Channel Norway have collaborated to create the city bike scheme[1]. UIP is the urban infrastructure company that runs the city bike scheme for Oslo, as well as schemes in Bergen and Trondheim. UIP has physical ownership of all the bikes and stations and does all the operational activities. The city bike scheme is financed through advertising, subscriptions, and a sponsorship. In order to use and unlock a bike, a person must have the "Oslo Bysykkel" app. Through the app, the user will be able to purchase a daily, weekly, monthly, or seasonal pass which will allow them to use the bikes. There are around 254 stations across the city. With the large number of stations, the travel time to the next closest station is minimal. This leads to a reduction in travel time for longer trips, but also trips within walking distance.

As of 2021, there are around 1 million people living in the city of Oslo. Of those 1 million people, about 100 thousand people are city bike users. These 100 thousand people make around 2.7 million trips a year.

3 Literature review

Schuijbroek et al. (2017) identify four research substreams in bike-sharing literature: strategic design, demand analysis, service level analysis, and rebalancing operations. Within these sub streams, this article will focus on the demand analysis. More specifically demand prediction of bikes based on public data. In this section, a literature review of factors considered in demand forecasting is conducted, followed by an introduction to various machine learning models proposed in related research.

[1] This follows the "Advertising Company" structure mentioned in section 2.1, with Oslo being the municipality and Clear Channel Norway being the advertising company.

3.1 Key factors for bike demand forecasting

For BSS with stations, demand forecasting and prediction are crucial. For truck-based rebalancing there are two approaches one being station level target intervals, while the other being station level demand prediction. With an accurate station level demand prediction there is improvement on the bike rebalancing seen in both studies Guo et al. (2019) and Liu et al. (2016)

Adequate supply, benefits the users by ensuring that they can pick up and drop off bikes during their preferred time at their intended station. There are constant fluctuations in demand for city bikes. These fluctuations are influenced by many factors with some of the most influential factors being weather, time, and day of the week.

Many studies have investigated the impact of weather conditions on bike demand (Nankervis, 1999; Bean et al., 2021). According to research by Nankervis (1999), wind, rain, and temperature were found to be the most critical factors affecting the number of riders. Bean et al. (2021) further identified precipitation as the most important weather factor. The higher the amount of precipitation the less bikes are used. In this case, people tend to take public transport. Bean et al. (2021) also found that there was a turning point on demand at around 22.5 degrees Celsius. This means that the demand increases as the temperature increases up until 22.5 degrees Celsius, and then starts to decline when the temperature rises above 22.5.

Another common factor used to analyse demand is the time of day (Nankervis, 1999; Kim, 2018; Du et al., 2019; Bean et al, 2021). Bean et al. (2021) found that the hour of day was the most significant variable. They found a bimodal usage frequency with one peak being early in the day, and another peak being in the early evening. These peaks are due work and school commuters. The peak starts early in the morning when people are biking to work and shows up again in the evening when people are heading home from work. This pattern of bike usage can be seen during the weekday in several other studies.

In addition to the expected factors mentioned above, other unexpected factors may also have a significant impact on demand, such as government policies or epidemic outbreaks. Since 2020, the COVID-19 pandemic has severely affected all aspects of daily life around the world. BSS have also been affected by the pandemic and have seen significant changes in usage and demand during this period. According to Kubaľák et al. (2021) there was a 46.25 percent decrease in bike usage during the pandemic period in Slovakia, with the summer months having the largest decrease in usage. This reduced usage was due to infection control measures, which meant that people worked and studied from home and thus they did not need to commute by bike. When the restrictions were lifted in December, the bike usage was similar to pre-pandemic figures. Furthermore, Kubaľák et al. (2021) found that the average length of rides increased, which they attributed to a change in purpose of bike usage. This purpose change would be from using the bikes as a method to get to work or school, to using the bikes to travel more leisurely around the city. Chai et al. (2021) notes that there was a 64.8 percent drop in the shared bike usage in Beijing, China during the pandemic. They found that the mobility in high tech areas, subway stations and shopping plazas was greatly reduced due to the quarantine restrictions. The overall bike usage was significantly influenced by the pandemic, and it still has some lasting effects.

3.2 Demand prediction model

In an effort to improve BSS, different analytical methods have been used over time and in different areas. Almannaa (2019) notes that bike prediction commonly uses one of these four approaches: statistical models, exploring and clustering algorithms, machine learning algorithms, and time series models. The focus will be on machine learning algorithms in the form of recurrent neural networks and random forest. Previous articles on this subject tend to either choose a specific time interval (Yang et al., 2018), or compare several of them (Boufidis et al., 2020).

Random forest (RF) is a supervised machine learning method for classification and regression (Du et al., 2019). RF creates an ensemble of decision trees. Each tree has randomly selected features, which are then used to compare the resulting mean squared error (Ashqar et al., 2017). According to Biau & Scornet (2016) the popularity of RF is greatly contributed by fact they can be applied to a wide range of prediction problems and have few parameters to tune. Within the topic of bike-sharing, RF has been used to find useful features of bike use characteristics regarding bike usage demand (Du et al., 2019), as well as predicting demand (Ashqar et al., 2017; Boufidis et al., 2020; Feng & Wang, 2017; Zeng et al., 2016).

In recent years, deep learning approaches have been used for various prediction problems including bike usage prediction (Boufidis et al., 2020; Wang & Kim, 2018; Xu et al., 2018; Zeng et al., 2016). The recurrent neural network (RNN) is one of the typical deep learning models used for sequential prediction problems, such as language modelling, speech recognition and machine translation (Zaremba et al., 2014). RNN is a recently developed deep learning method for time series data modelling. As opposed to artificial neural networks, RNN accounts for temporal dependencies in the model structure. This is done by recurrently connecting hidden layers at different timestamps. However, RNNs are known to have suffered from the gradient vanish problem, which means that it is difficult for a RNN model to learn the weights after many timesteps. To overcome this problem, an extension of RNNs, called long-short-term memory (LSTM), was developed. More details can be found in the literature Xu et al. (2018).

LSTM and GRU have previously been compared with other models and approaches within time series prediction. In a comparison with three different varieties of RNN as well as other nonparametric and parametric approaches[2], LSTM outperformed the other algorithms in terms of accuracy and stability (Ma et al., 2015). Similar

[2] The models compared were the RNN models Elman NN, Time-delayed NN, Nonlinear Autoregressive NN, with the other approaches being SVM, ARIMA and Kalman filter.

results were found in traffic flow predictions by (Fu et al., 2016), who in their comparison of LSTM, GRU and ARIMA saw better performance for the RNN models. LSTM thus shows a proficiency in time series prediction with longer temporal dependencies. GRU is an RNN based on LSTM, and saves more computational resources than LSTM with similar performance (Wang & Kim, 2018). Both LSTM and GRU have been used in bike-sharing demand forecasting (Xu et al., 2018), and the results show that the two models perform similarly. However, the results of GRU are more accurate while the training time is also faster than LSTM.

LSTM, GRU and RF are three models that have been commonly used when predicting bike usage demand due to their high accuracy. While studies such as Drevland & Finseth (2018) and Xu et al. (2018) have used similar models with the intention of predicting demand, they predict for shorter intervals between 10 to 60 minute intervals. The paper by Drevland & Finseth (2018) also predicts for the BSS in Oslo, however with the intention of predicting demand station by station. The BSS system predicted by Xu et al. (2018) is a station-free system which is likely to have different user behaviour and demand.

4 Research methodology and design

4.1 Data preparation

4.1.1 Data collection

This study used three data sources, namely bike trip data, weather data, and COVID-19 data. The details of each data source and its limitations are explained below.

The main dataset was collected from OCB and includes historical trip data from April 2019 to December 2021. The data is published under the Norwegian License for Open Government Data (Norwegian Digitalisation Agency, 2020). This data complies with GDPR regulations and does not include sensitive or confidential customer information. The machine learning models and datasets used in the work can be found on GitHub[3]. The raw data for each month is split into separate files going back to April 2016. The April 2019, data format was updated to include geolocation variables. This gives a natural cut-off point for the data used in the analysis. This bike data will later be merged with the corresponding weather data. However, weather information for some parts of Oslo is not sufficient. In this case, in order to better match the corresponding weather information, the bike demand of Grünerløkka, one of the busiest areas in Oslo, will be the focus of this study.

The historical meteorological data for Oslo is gathered from the Norwegian Centre for Climate Services (NCCS, n.d.). There are 38 weather stations in Oslo, each of which may collect different weather elements. The dropdown selection on this website offers the added benefit of being able to choose the time resolution, time period, weather station, and various weather information. On the webpage, up to five weather elements can be selected from the following categories: temperature (daily mean temperature), wind (highest wind speed of the day), snow (snow depth), and precipitation (daily precipitation). In this study, the hourly precipitation, minutes of precipitation within an hour and temperature will be used. The data is collected at the Blindern station, as this station provides the most accurate weather for the city of Oslo with the desired variables for this study.

Lastly, the final data source is related to COVID-19, as it has led to a reduction in mobility due to lockdown and government recommendations. The data was created based on a timeline of these measures taken by the (Oslo Municipality, 2022).

The open data used in this thesis is limited in terms of explanatory features. This led to the need for additional data from varying sources. Also, there was missing data during data collection which had to be worked around. Another challenge is that weather stations for certain parts of Oslo lacked relevant weather information gathered on an hourly basis. This led to the choice of Blindern station for weather

[3] https://github.com/nithun96/BSSThesisOslo

information. There is also a lack of unified data on COVID-19 restrictions in Oslo. Which led to a manual input of the COVID-19 feature.

4.1.2 Data exploration

For the data that was collected, Python was used to properly prepare it. The first piece of data that was changed was data from OCB. As the datasets were composed of individual trips, code was made to create rows for each hour. Data was then created to fill the times there were no trips. There was also an analysis done on the data to see the influential features. Figure 4.1 shows the user demand for the time of day. OCB has a bimodal usage frequency like the usage frequencies from the articles mentioned in section 3.1. The demand in the figure has a small peak around 5 am to 6 am and another peak from 2 pm to 3 pm. These peaks are for when people start heading to work and when people start heading home.

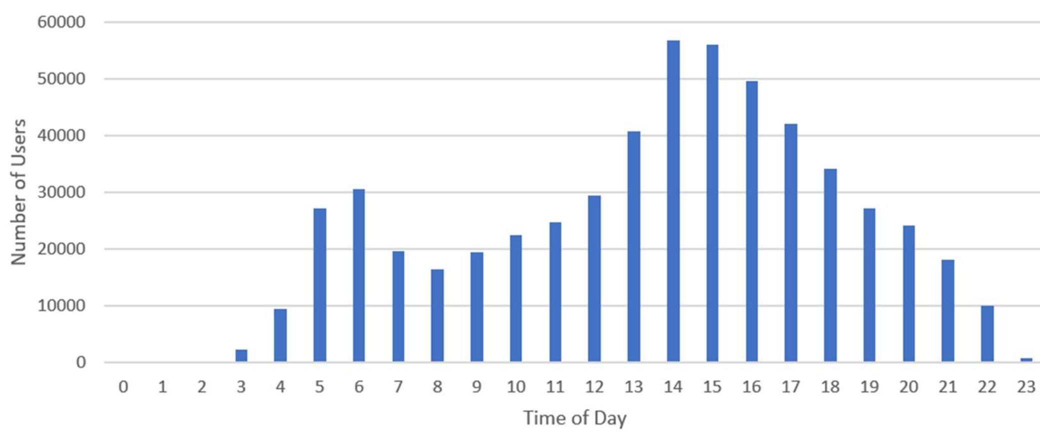

Figure 4.1: Users per time of day

Figure 4.2 shows the demand levels from Monday to Friday. It is clearly seen that there are more users during the weekdays than there are during the weekends. This is also explained by the change in trip purpose from weekday to weekend. Figure 4.3 shows the demand per time period and day of week. There are six, four-hour time periods. In the figure, the hourly demands are identical from Monday to Friday. The weekend hourly demand differs from the weekday as seen by the bell curved shaped demand on Saturday and Sunday. The weekends are also characterized by its low overall demand in comparison to the weekdays, with mornings (04-07) being heavily reduced. This also influenced the *DemandTime* feature for the model which will be mentioned in more detail later in the paper.

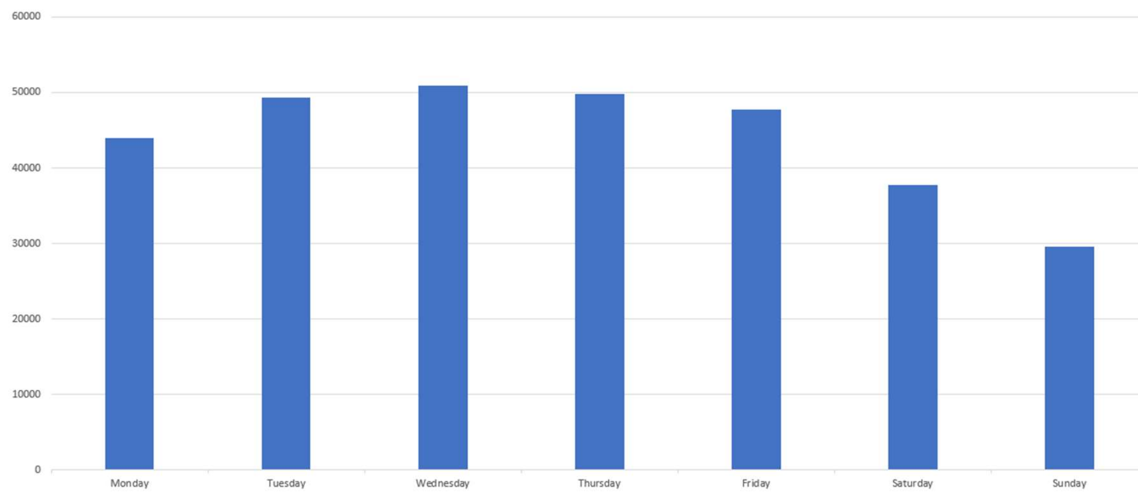

Figure 4.2: Day of the week

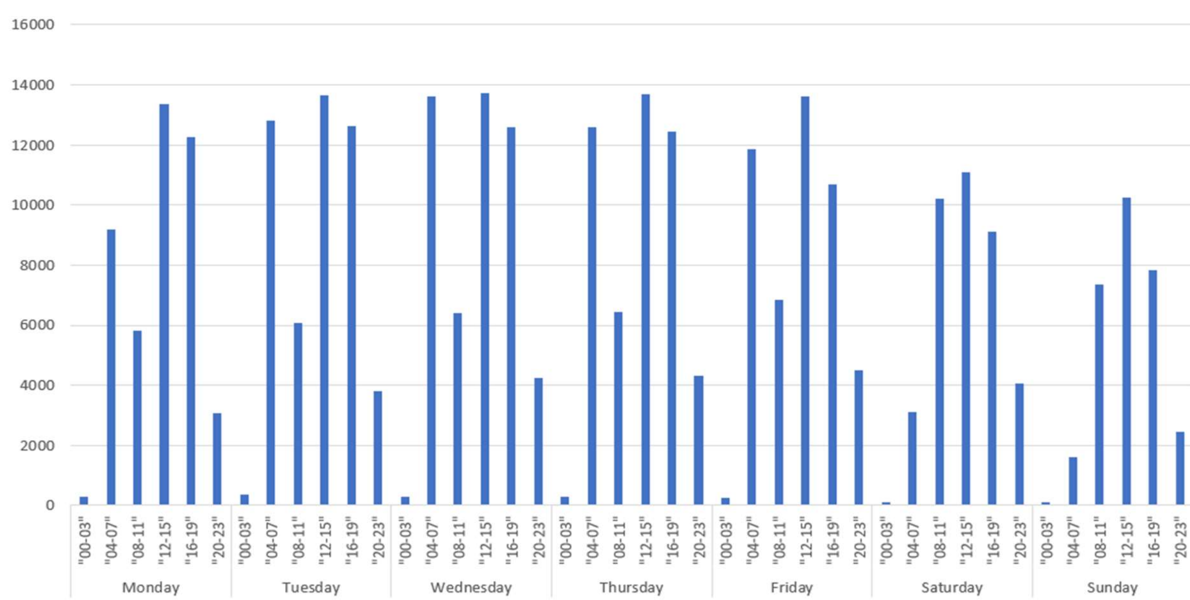

Figure 4.3: Users per time and day of week

Figure 4.4 shows the user demand for temperature in Celsius. Just like mentioned in section 3.1 it is seen that an increase in temperature up to around 17.5 degrees leads to an increased bike demand. Temperatures higher than 18 degrees leads to a decrease in bike demand. A similar decrease can be seen at lower temperatures that likely correlate with winter months.

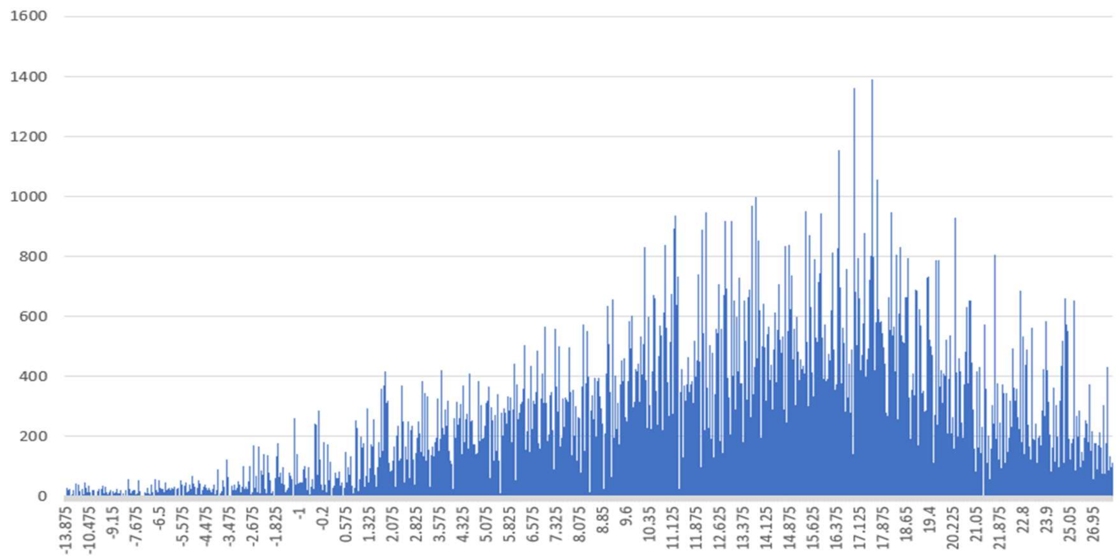

Figure 4.4: Users by temperature

4.1.3 Data pre-processing

For any data driven model to work as intended, it is necessary to have data that has been cleaned and prepared properly. Data with missing values or incorrect symbols could throw off the model and lead to inaccurate results. There are several different methods used to prepare the data.

The OCB data records details of each trip, such as start time, end time and the corresponding station name. After selecting trips that were departure from the Grünerløkka region, only the start time of each trip was kept, and other information not relevant to this study was excluded from the following analysis. Based on the starting time, several temporal features are generated and will be used as input data for modelling. These features are hour, month, day of month, day of week, time period, weekend and season as shown in Table 4.1.

Data source	**Features**	**Description**
Weather	Precipitation (Numeric)	Average precipitation during the time period.
Weather	PrecipitationMinutes (Numeric)	Total minutes of precipitation.
Weather	Temperature (Numeric)	Average temperature in Celsius during the time period.
Weather	TempLimit (Binary Dummy)	Binary variable for temperature limit, with 0 being below 4 degrees and above 21 degrees Celsius. And 1 being temperatures between 4 and 21 degrees Celsius
OCB	Day (Categorical)	Categorical value between 1-31 showing the day of the month the observation occurred.
OCB	Day (Binary Dummy)	Seven variables for each day (Monday-Sunday) showing the day the observation occurred.
OCB	Month (Binary Dummy)	Twelve variables for each month (January-December) showing the month the observation occurred.
OCB	Time period (Binary Dummy)	Six variables for each time period the data set is split into (0-3, 4-7, 8-11, 12-15, 16-19. 20-23).
OCB	DemandTime (Categorical)	3 different values. 0 being for the times with the least amount of demand (0-3,20-23). 1 being the time periods with the second most demand (4-7,8-11). And 2 being the periods with the most demand (8-11 and 12-15).
OCB	Weekend (Binary Dummy)	Binary variable with 0 being Monday to Friday, and 1 being Saturday to Sunday
OCB	Season (Binary Dummy)	Binary variable. November to March being 0 and April to October being 1
Oslo Municipality	Restrictions (Binary Dummy)	Five binary variables showing the intensity of the restrictions. These levels are none, mild level, medium level, high level, and total lockdown.

Table 4.1: Table of features

The model in this paper will be predicting BSS demand, which is the number of users that start a bike ride within a given period of time. As shown in figure 4.1, there are multiple peaks and troughs in demand throughout a day. A way to accurately portray them is by splitting the day into several time periods. The focus will be on specific time interval of four hours, thereby splitting the day into six different sections. The hourly data was combined into six time periods as shown in figure 4.5. These time periods were then explained by dummy variables split into 6 periods: 12 AM – 3 AM, 4 AM – 7 AM, 8 AM – 11 AM, 12 PM – 3 PM, 4 PM – 7 PM and 8 PM – 11 PM. The creation of the days into 6 time periods also led to some engineering of other features.

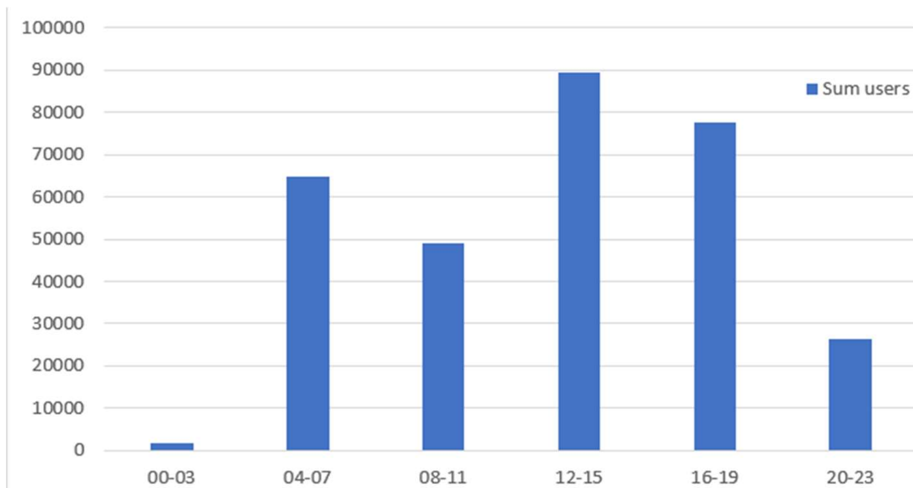

Figure 4.5: Time periods of final dataset

Both the precipitation and temperature variables had to be modified in order to fit into the specific time periods. The most reasonable way to modify it was to create the sum of precipitation as well as an average temperature for each time period. The season variable was chosen based on the number of users per time of year. From November to March there is a reduction in the supply of bikes and in the demand for the bikes. As for the COVID-19 lockdown features, binary variables were created by selecting the time period when restrictions were in place for Oslo and the intensity of restrictions. None is before covid with zero restrictions. Mild is when there are few restrictions consisting mostly of recommendations to keep distance from people and to stay away from public transport. Medium is when restaurants and stores are mostly closed and recommended to work from home. Strong is when schools and workplaces are mostly closed. And total lockdown is when there is a lockdown on any social interaction and only grocery stores are open. To determine the demand of city bikes, it is necessary to find which variables effect the demand. The variables that were chosen based on the analysis of the features, were the weather variables (Temperature and Precipitation), time variables (Time of Day, Day of week), and COVID-19 variable (if there were COVID-19 restrictions in place or not).

The final step in data pre-processing is feature scaling the values within a range of 0 to 1. (Nkikabahizi et al., 2022) explain that feature scaling is a pre-processing method of data that consists of transforming numerical attributes with different ranges into the same scale, between 0 and 1. This improves numerical input stability, reduced the time for learning the predictive model and can make a significant improvement in prediction model performance. The scaling is based on the target variable and the features in the training set, which is then applied to all four sets of data.

4.1.4 Data split

The dataset is divided into two subsets: the first 75 percent of the dataset is considered as the training set, which is used to train the models, and the remaining 25 percent is the test set, which is used to evaluate the performance and robustness of the model. This splits the dataset with 5282 rows into a training dataset with 1057 rows and a training dataset with 4225 rows. The split data is shown in figure 4.6. The x-axis of the graph represents the observation period, and the y-axis indicates the number of users (i.e., trips) recorded in the BSS system during that period. The time period of the whole dataset is from April 2019 to December 2021, with the data being split at the 13. of June 2021.

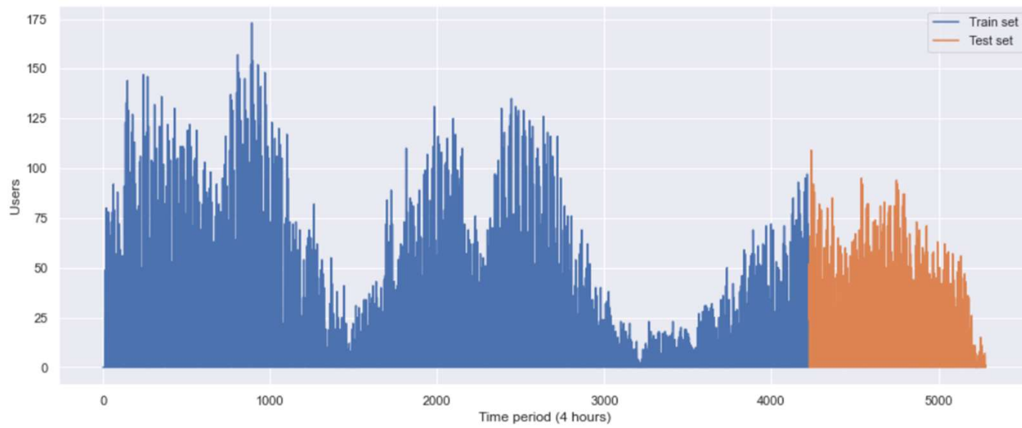

Figure 4.6: Train-test data split

4.2 Modelling

4.2.1 Model description

The models chosen are RF, LSTM and GRU as mentioned in section 3.2. RNNs were initially used for language models, yet they have also in recent years been used for time series predictions due to their performance. In this paper, LSTM and GRU have multiple, yet the same set of hyperparameters that are run with the same layers. RF on the other hand, has the benefit of having fewer parameters to tune. Compared to other classifiers the advantages of RF include: "(1) very high classification accuracy; (2) a novel method of determining variable importance; (3) ability to model complex interactions among predictor variables; (4) flexibility to perform several types of statistical data analysis, including regression, classification, survival analysis, and unsupervised learning; and (5) an algorithm for imputing missing values" (Cutler et al., 2007).

The model used is in the scikit-learn package in Python (Pedregosa et al., 2011). For LSTM and GRU, Keras is used to run and create the model (Chollet, 2015) with scaling done by scikit-learn. Both matplotlib and seaborn are used to create the graphs shown later. Pandas and NumPy are used for data manipulation.

As for the features used, feature importance of multiple features was looked at before choosing the final set of features shown in table 4.1. There were several other features that could have been selected, but when looking at the feature importance for the model it was seen that those features did not have any significant effect. When other weather variables such as amount of snow, cloudiness, and amount of humidity were looked at, it was seen that it was unnecessary to include in the model.

The relevant metrics in this study are mean absolute error (MAE), root mean square error (RMSE) and R^2. These metrics are commonly used within this subject, among them Yang et al. (2018). The metrics are chosen due to them having different focuses, with MAE explaining by how many pickups the predictions are off and RMSE penalizing predictions far away from the actual observations. R^2, being the coefficient of determination, represents the goodness of fit for the models used.

4.2.2 Model creation (GRU and LSTM)

With the dataset fully pre-processed (section 4.1.3) and split into training and test sets (section 4.1.4), there are some additional steps remaining when using GRU and LSTM. These normalized datasets are then used to create a three-dimensional matrix:

- Timestep (*t*): Number of observations required to predict the next value.
- Features (*p*): 38 input features including 29 temporal features, 4 weather features and 5 covid-related features.
- Samples (*k*): $k = N\text{-}t\text{+}1$, where N is the number of observations before transformation.
- Learning rate (l): Controls the rate of which a model adjusts after each iteration.
- Unit (n): The number of nodes in a layer.

The value of timestep, is one of multiple hyperparameter value that have to be decided. The baseline value for timestep is one. Yet, as the timestep value is related to time, it is logical to test 24-hour intervals. Figure 4.5 shows how each observation in the dataset is split into six. Each sixth observation would be the equivalent of a day and is used as a basis for the values tested. However, only a couple of timestep values are tested and those are: [1, 12, 24]. Two other hyperparameters are tested, while others were chosen by acquiring a better understanding of the model behaviour.

The two other hyperparameters tested were learning rate and unit. The learning rate is vital as it determines how quickly a model forgets what it learned in its previous iterations. This is determined by how relevant the older data is when predicting newer observations. A balanced learning rate can be found when looking at the loss of a function over time after running the model. Examples of different learning rates can be seen in figure 4.7. A high learning rate will be too quick at trying to adjust and will finish before being able to find a good solution. If it is far too high, it will cause too drastic updates in its search for the solution. A low learning rate, however, will be adjusting at a too slow rate to the point it will get stuck at a local minimum without being able to find a better solution. The tested learning rates were [0.01, 0.001, 0.0001]. This will be shown in section 5.1 with the results of the model.

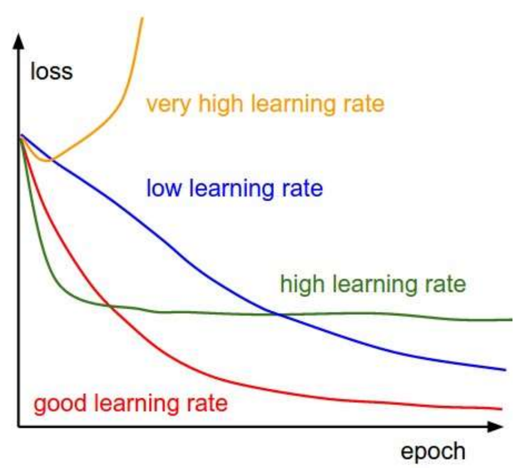

Figure 4.7: Loss function (Stanford, n.d.)

The final hyperparameter tested was the number of units in each layer. Units are also called neurons and defines the dimension of hidden states. This hyperparameter, along with timestep, determines the time it takes for the model to learn. 64 is a common value for units, and the tested values are: [16, 32, 64, 128].

To determine suitable hyperparameter values the following experiment were performed. The model is run three times for every combination of the hyperparameters mentioned earlier. The metrics are gathered for each run, and the table shows the mean of those three runs to avoid any possible inaccuracies. The desired result is a high R^2 (R-squared) as well as low RMSE and MAE values. Table 4.2 shows 36 different combinations of hyperparameters with varying performance for LSTM and GRU. The results are colour coded to show distinguish better between the different values.

The best performance can be found for index of 4 and has the best combination of values for both models. The optimal hyperparameters would be a timestep of 1, learning rate of 0.001 and unit of 16. In this model, the runtime is not taken into consideration, however that would disadvantage higher values for timestep and unit. Coincidentally, the optimal model in this case also has a low runtime.

	Time step	Learning rate	Units	GRU RMSE	GRU MAE	GRU R-Squared	LSTM RMSE	LSTM MAE	LSTM R-Squared
0	1	0.0100	16	24.83	16.68	0.72	28.51	18.50	0.63
1	1	0.0100	32	25.82	16.70	0.69	26.45	17.53	0.68
2	1	0.0100	64	27.06	17.65	0.66	26.08	16.94	0.69
3	1	0.0100	128	26.04	17.54	0.69	26.39	17.42	0.68
4	1	0.0010	16	23.91	15.80	0.74	24.36	16.07	0.73
5	1	0.0010	32	25.14	16.61	0.71	25.63	16.82	0.70
6	1	0.0010	64	25.29	16.42	0.71	25.90	16.96	0.69
7	1	0.0010	128	24.79	16.50	0.72	25.41	16.77	0.70
8	1	0.0001	16	24.46	16.31	0.73	24.62	16.29	0.72
9	1	0.0001	32	24.76	16.46	0.72	24.31	16.20	0.73
10	1	0.0001	64	24.56	16.37	0.72	25.11	16.64	0.71
11	1	0.0001	128	24.78	16.45	0.72	24.09	16.01	0.73
12	12	0.0100	16	26.03	17.14	0.69	25.06	16.10	0.71
13	12	0.0100	32	25.91	16.74	0.69	24.80	16.35	0.72
14	12	0.0100	64	28.00	18.98	0.64	27.24	17.54	0.66
15	12	0.0100	128	25.26	16.75	0.70	25.79	17.04	0.69
16	12	0.0010	16	25.74	16.96	0.70	28.57	18.69	0.62
17	12	0.0010	32	24.66	16.22	0.72	27.52	17.88	0.65
18	12	0.0010	64	26.88	17.96	0.67	27.41	18.06	0.65
19	12	0.0010	128	24.41	16.42	0.73	26.26	17.44	0.68
20	12	0.0001	16	24.91	16.54	0.71	30.67	21.18	0.56
21	12	0.0001	32	25.60	16.72	0.70	29.99	20.87	0.58
22	12	0.0001	64	26.41	17.28	0.68	29.67	19.91	0.59
23	12	0.0001	128	24.25	15.84	0.73	28.98	19.51	0.61
24	24	0.0100	16	26.48	17.58	0.68	25.37	16.54	0.70
25	24	0.0100	32	28.78	18.97	0.61	25.52	16.98	0.70
26	24	0.0100	64	24.95	16.81	0.71	26.92	17.47	0.67
27	24	0.0100	128	28.13	18.97	0.64	25.35	16.77	0.71
28	24	0.0010	16	25.10	16.37	0.71	24.65	16.37	0.72
29	24	0.0010	32	25.18	16.67	0.71	26.18	17.39	0.69
30	24	0.0010	64	26.57	17.66	0.68	25.18	16.76	0.71
31	24	0.0010	128	25.40	16.83	0.70	25.78	16.61	0.70
32	24	0.0001	16	24.19	16.15	0.73	31.65	22.08	0.54
33	24	0.0001	32	25.04	16.51	0.71	32.30	21.81	0.52
34	24	0.0001	64	25.82	16.85	0.69	32.19	22.18	0.52
35	24	0.0001	128	26.55	17.28	0.68	29.71	19.79	0.59

Table 4.2: Time step comparison

A typical setting for LSTM and GRU is used in this study, which include an input layer, a hidden layer, and output layer. The Glorot uniform[4] initializer is used for weight initialization and the Rectified Linear Unit[5] (ReLU) is used as the activation function. ReLU is commonly used for neural networks because it runs quickly and reduces the likelihood of a vanishing gradient. The model also includes two dropout layers to prevent overfitting.

Several hyperparameters also should be determined to ensure the effectiveness of the learned model and avoid overfitting. First, an optimization technique called *EarlyStopping*[6] is used in the model. It monitors whether a valuation loss is no longer decreasing. This stops the model from running the set number of epochs if there is no improvement in the model over a longer period of time. An additional hyperparameter, called patience, can be used in this approach is added to give the model more lenience at runtime, thereby improving the model. The value for patience is set to 20, which means that if the model runs for 20 epochs without any improvement, it stops training.

The next set of hyperparameters and their values are listed in Table 4.3, with some of them being reliant on each other. The number of epochs is the number of iterations that the model will work through the entire training dataset (Bell & Gaillard, 2017). In this case, the *EarlyStopping* function is likely to stop the model from reaching the max number of epochs set. The batch size on the other hand, determines the number of training samples utilized in one epoch (Murphy & Gaillard, 2017). Steps per epoch is the total number of batches the model uses before declaring an epoch finished and starting the next (Team Keras, n.d.). The value is commonly set to be the number of batches required to run the entire training dataset, which is

---

[4] https://www.tensorflow.org/api_docs/python/tf/keras/initializers/GlorotUniform
[5] https://keras.io/api/layers/activation_layers/relu/
[6] https://keras.io/api/callbacks/early_stopping/

automatically done with Keras. While it is possible to adjust both batch size and steps per epoch, the value calculated does not need optimization.

Hyperparameter – Model fit	**Value**
Max number of epochs	200
Validation split	0.2
Batch size	default
Steps per epoch	default

Table 4.3: Hyperparameters used in the model fitting function

Based on the above setting, the model is finally compiled with an optimizer as the last step. The Adam[7] algorithm was chosen as the optimization method.

4.2.3 Model creation (RF)

After the data split mentioned in 4.3.2 the data is ready to be used on the RF model. The model used is RandomForestRegressor in the scikit-learn package[8]. Several hyperparameters can be tuned to learn a random forest model. The number of trees in the forest is set to 1000. The remaining hyperparameters are the default setting for the model. The mean square error is used as evaluation criterion. Mean absolute error (MAE) is an option for evaluation criterion, however it increases the runtime of the model drastically while not providing any improvement in model performance. The minimum samples required to split a node is set to 2. And there is no limit to the depth of tree.

The model is then built based on the training dataset. In the next step, the model is used to predict the target variable.

---

[7] https://keras.io/api/optimizers/adam/

[8] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

5 Model Evaluation

This paper aims to create an accurate predictive model for OCB that can describe bike demand in both pre-pandemic and post-pandemic period. The results of different predictive models are evaluated and discussed in this Chapter.

5.1 Result

5.1.1 LSTM and GRU

Graphs showing training and validation loss and predictive comparisons have been used to gauge the accuracy and performance of the LSTM and GRU models. This is in in addition to key metrics in the form of MAE and RMSE and R^2. These metrics gives the user a better understanding of the model performance. The following graphs in this section is from a randomly chosen run of the model.

The purpose of these training and validation loss graphs in figure 5.1 and 5.2, is to show the model performance and to identify parts of the model with room for improvement. It gives an understanding of model fit, and whether model is underfitting or overfitting. If the training loss is much more than the validation loss, the model is underfitting and is not accurate in its predictions. If the training loss is less than the validation loss it would be overfitting and not be a reliable model for general situations. As can be seen in the two figures, the train loss never reaches the validation loss, yet it is not underfitting. There is room for improvement in terms of model fit, however it does not seem like there is any underfitting or overfitting in the model. On the x-axis of the two graphs, there is a difference in the number of epochs due to the early stopping mentioned in 4.3.4. This is meant to decrease the likelihood of overfitting. If the training and validation loss have similar values, then the model predicts well and that it does not overfit or underfit.

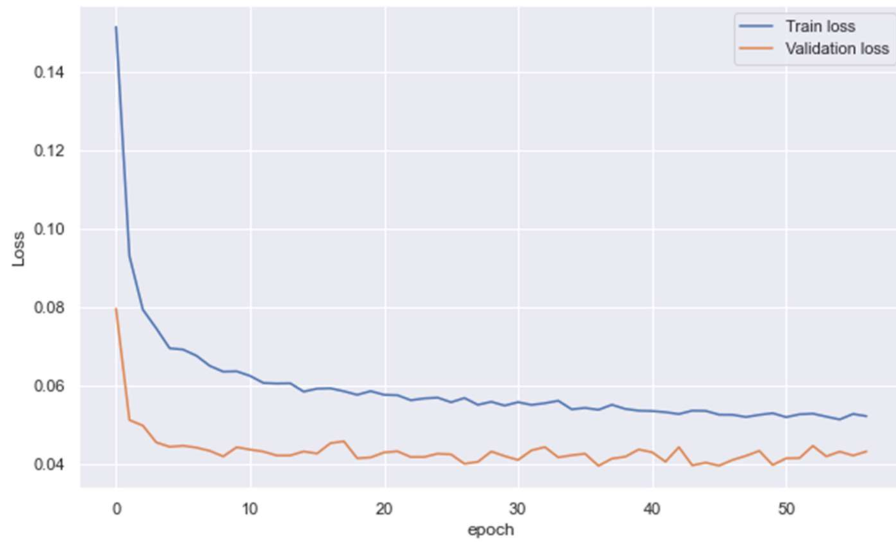

Figure 5.1: LSTM training and validation loss

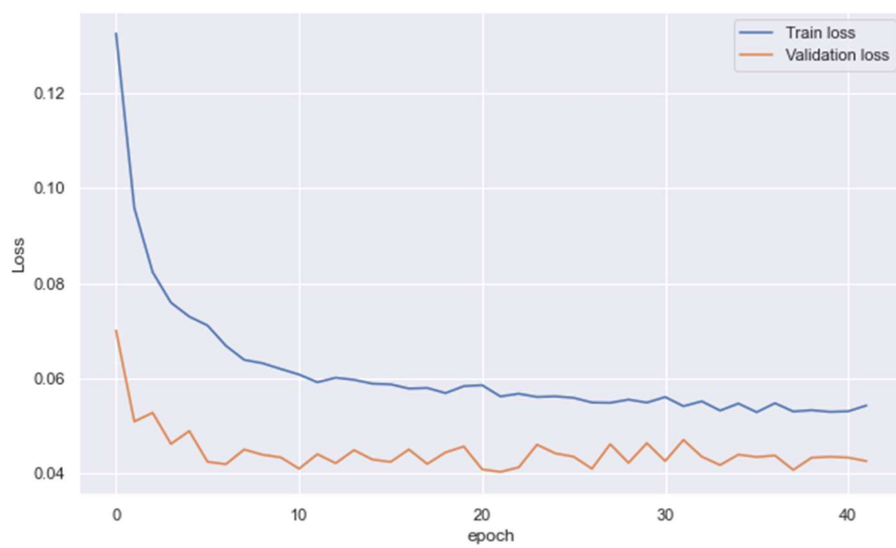

Figure 5.2: GRU training and validation loss

In figure 5.3 and 5.4, the predicted values are overlayed over the true values. In the figures it is seen the predictions are overestimating between the hours of 300 – 600 and that it underestimates from 600 – 900. However, both models follow the general trend when compared to the true future.

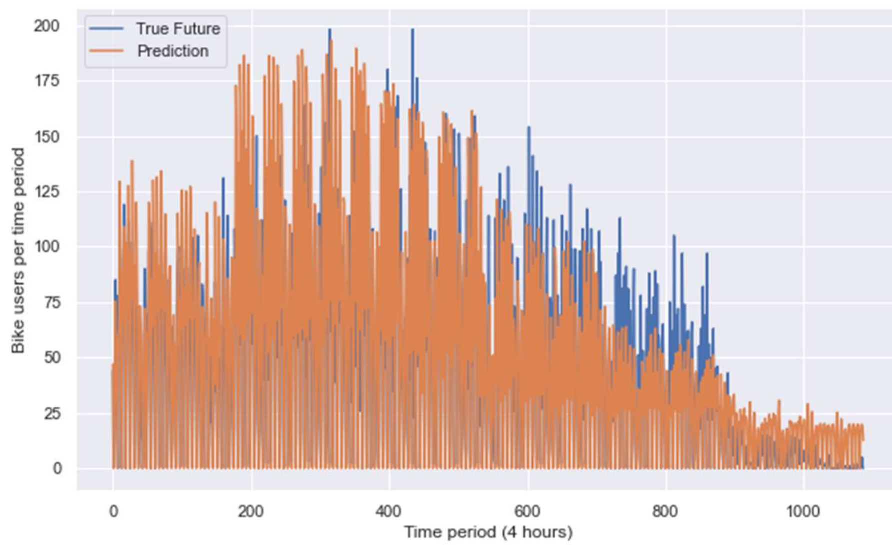

Figure 5.3: LSTM prediction compared to true future

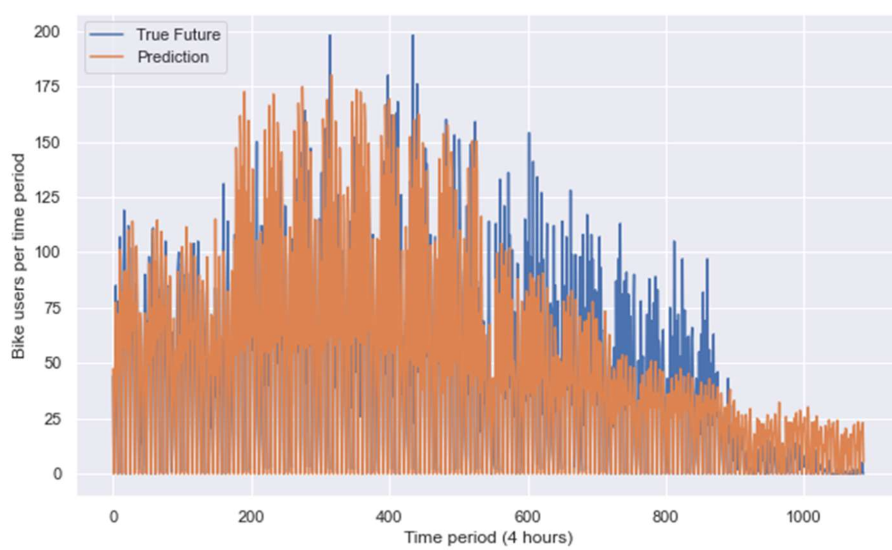

Figure 5.4: GRU prediction compared to true future

5.1.2 RF

A random forest model was also created as a comparison to the GRU and LSTM models above. The graph below follows a similar format with predicted values over true values. The random forest predictions follow the general trend as seen in GRU and LSTM. However random forest unlike the two previous models mostly overestimates in the early predictions and underestimates from time period of 800 – 1100. The random forest prediction overall is accurate in its demand prediction.

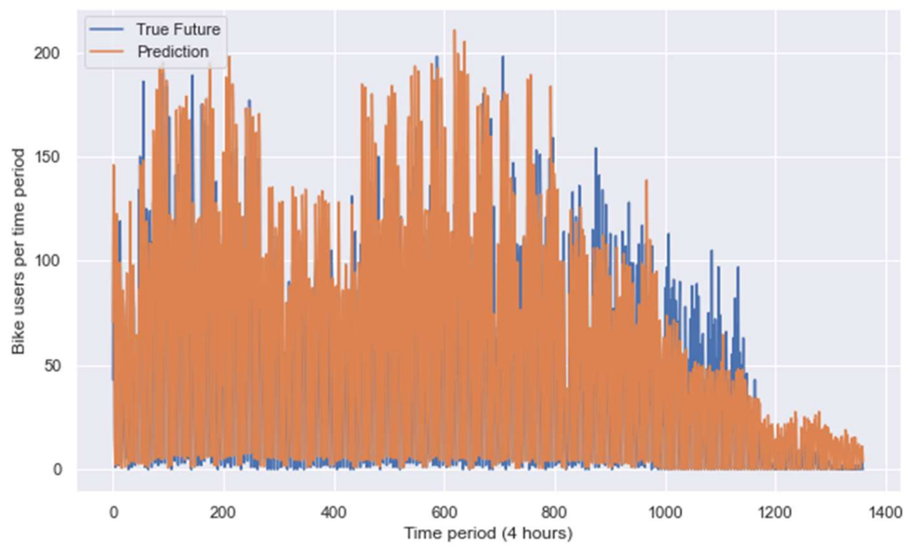

Figure 5.5: RF prediction compared to true future

Due to the number of nodes of this RF model being 1000, and there not being any limits to the depth of the tree, any visualization of the trees in the model would be too large to visualize as well as uninformative. However, the feature importance created with the RF model will be looked into in section 5.1.4.

5.1.3 Performance comparison

Table 5.1 shows the different metrics that were used to score the accuracy and reliability for the machine learning models. The three metrics are MAE, RMSE, and

R^2, which are mentioned in section 4.2. To add context to the MAE values, it is worth looking at table 5.2. It shows that the highest variation in user demand is 321 with the mean at 56.84. The MAE for the models shows the number of users that the model can over or under predict for each time period. Here, this means that if the true value is 56.84 and the MAE score is 14.30, then the prediction score will be within the range of [42.54, 71.14]. All the models have a fairly high R^2 score with the lowest being LSTM at 0.7 and the highest being GRU with an R^2 score of 0.77. GRU has the best scores for MAE and RMSE as well and is seen as the best model of the three. LSTM has better MAE and RMSE but has a lower R^2 score compared to RF.

Metrics	LSTM	GRU	RF
MAE	16.25	14.30	17.01
RMSE	23.61	20.80	24.41
R-Squared	0.70	0.77	0.73

Table 5.1: Comparison of LSTM, GRU and RF

Descriptive statistics	Value
Mean	56.84
Max	321.00
Min	0.00

Table 5.2: Descriptive statistics of user demand

RMSE treats the larger errors more harshly compared to MAE which treats all errors equally. For both RMSE and MAE the lower the number is, the better the model is at predicting. R^2 is measured separately to RMSE and MAE and gives an intuition on how accurate the features are at predicting demand.

The results show that the models perform to a high degree of satisfaction and can reliably be used. From the results it can be see that they are not 100 % accurate. This indicates that they are not overfitted to the testing data which would lead to it not being reliable to use with new data. The model that performs the best and has the most accurate predictions, is the GRU model. It outperforms both RF and LSTM in all three metrics used to judge the models. This falls in line with GRU's performance shown in table 4.2 where different hyperparameters were tested. The performance of the two other models is fairly similar, with small margins in the different metrics. With the best performing model, it could be used for area-specific demand prediction by OCB in order to make some tweaks and changes within their operations. Even though the level of performance for this model is satisfactory, it can be further improved with better data input.

5.1.4 Feature importance

Figure 5.6 is created by using the RF model's feature importance function[9]. The figure shows that the most influential features are the weather and time related features. *DemandTime* is the most influential feature in the model with *Temperature* being the second most important. Most of the top features for the model are expected, but one of the less expected features is the *Day* feature. The numerical day of the month seems to have some significance in BSS usage. Compared to the other features, the five COVID-19 restriction features do not seem to have as much as an effect as expected. *Restrictions None,* however, has a high significance on the model meaning that the bike usage was affected by the lockdown, but the intensity of the lockdown had less of an effect. Other features of note would be the *Month* binary variables having varying levels of significance, while the *Season* feature achieves better results by aggregating months.

[9] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#sklearn.ensemble.RandomForestRegressor.feature_importances

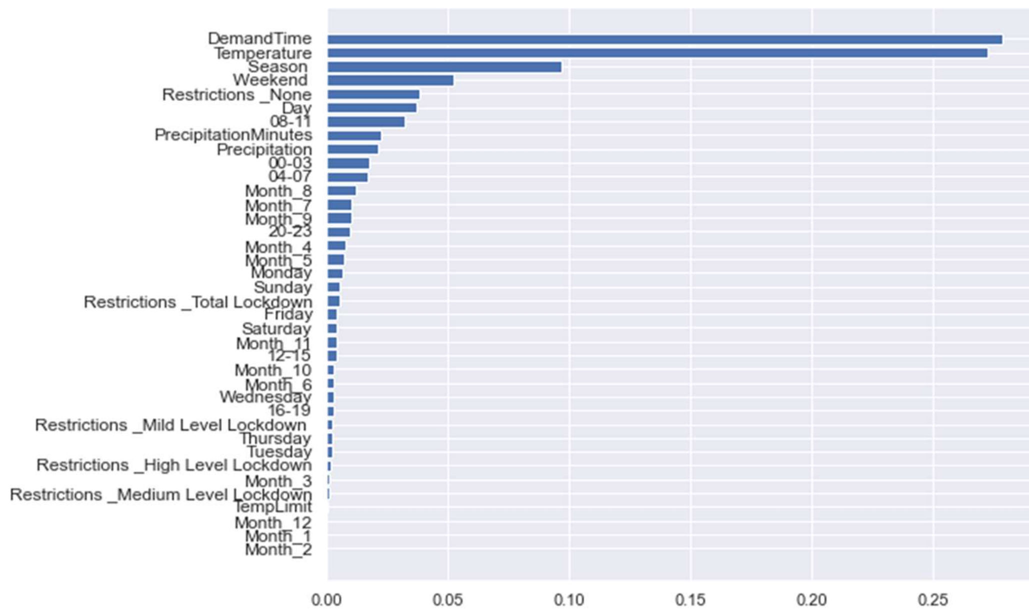

Figure 5.6: Feature importance for RF

5.2 Discussion

In their paper, Wang & Kim (2018) studied the same machine learning models however with short-term forecasting of station usage. They discovered that the three models achieve good performance with acceptable error and comparative accuracies. While LSTM and GRU are similar, GRU has more accurate results and faster training time than LSTM. This behaviour is consistent with the results in 5.1.3. The paper compares the models in different time intervals and discover that RF is better when the time interval is short. The time interval in this case is not short, however it is likely that model performance would be achieve similar results with a short time interval. Overall, the findings in this paper confirm the findings in previous research with these models. The results of all three models are strong given the hyperparameter tuning done on the RNN models. RF did not require similar work, due to the model structure and the scikit-learn package's ease of use. Extensive testing of hyperparameters for RF could lead to additional improvements to the model.

There are several limitations identified regarding this study. The majority of the data is affected by COVID-19 lockdown and measures, possibly making it less suited for bike-usage post-pandemic.

Additional relevant features worth researching in the future include public transportation station location as well as their schedules. Further information on public transportation could lead to more accurate results as public transport and BSS complements each other. Public demographic data as well as user demographic data from OCB are also potentially relevant features which would be influential when using the model across several different regions at the same time. With more data directly from OCB features such as pass type and price could lead to a prediction model for optimal pricing. The model could then provide a price recommendation based on predicted demand.

Implementing features without knowing its accuracy and relevancy in advance can be time consuming and leads to prioritizing known quantities. This leads to there being a number of features that were not included due to time constraints.

With the extensive research present in this field, there is much that can be done. This would include looking deeper into additional features for demand prediction. Or could include modification to the model to create a better supply rebalancing method or even a dynamic pricing model. Accessing additional data that is not publicly available could also lead to either more accurate models, or the ability to look into different yet adjacent subjects. Further hyperparameter tuning is possible, in addition to implementing new methods that improves it. One example would be to implement a learning rate scheduler with Keras, which would adjust the learning rate as the model runs. Tensorboard is also identified as a tool possibly able to improve hyperparameter tuning, modelling, and visualization.

6 Potential benefits of business implementation

OCB are likely to be using multiple demand models for different purposes already. Traditionally such models would be used to know the demand for stations and determine the number of bike racks needed. It would also be vital for OCB to stock bikes evenly across the city by creating separate models for bike supply rebalancing. These uses of the models have their focus on the logistics involved in the BSS system, yet they could be used to improve the pricing as well. One option would be dynamic price changes based on predicted demand. This would mean the city bike could predict the demand in a specific area and then with such predicted demand they could reduce or increase the price it would cost to rent the bike for a single trip. This would lead to some financial benefits, as well as give them a way to control number of bikes within a specified area. Dynamic pricing and incentives can also be used at a station-specific level to improve the distribution of bikes. Decrease in price would increase the amount of people that take out bikes from a specific area and vice versa with an increase in price. In areas with high demand, transport is usually arranged to move bikes from station to station. If this could be solved by the user themselves, it would save OCB money and reduce the environmental impact of city bikes. This could justify a heavier discount or incentives for the user. This incentive-based system would either give users a discount or some coupons for dropping off or taking their bike from certain stations.

The model in this paper could also be used in conjunction with other models to look into smaller areas such as at a station-specific level. Demand forecasting could also be used for longer time periods in order to find potential station locations. For the potential station benefits features such demographics, public transport, and property information would have a large influence.

7 Conclusion

The data gathered from OCB's public database was lacking in some areas but with additional data pre-processing and feature selection, informative enough to create an accurate predictive model. The Random Forest and Recurrent Neural networks used in the model predictions had high accuracy. The best model was GRU and gave an MAE of 14.30, RMSE of 20.80 and R^2 of 0.77. The introduction of COVID-19 influenced user demand from April 2020 to December 2022 and is a feature that would be important for city bikes when predicting demand during future pandemic or lockdowns. The results from the model are accurate and reliable, which means that this model could potentially be integrated into the current bike sharing system.

Many trips were gathered with the OCB data from early 2019 to the end of 2021. With such data there was a clear pattern on the behaviour of BSS users in Oslo. This behaviour was very similar to behaviour seen in previous BSS research. These patterns pointed towards users using the bikes as a complement to public transportation as well as a way to get to and from work or school. There was an expectation that the intensity of lockdown measures would have a large influence on the overall model, but it ended up not being as significant of a feature for the prediction model as expected.

Collaborating directly with OCB and spending further time on feature implementation could lead to improved scores for the model. This would also lead to more options in the model creation and could lead a model which uses demand forecasting that would influence other parts of OCB's organization.

This is a relevant and interesting topic for us due to the city BSS's role as a short-distance public transport. Emerging competitors in short-distance transport is an important challenge going forward. Private companies providing sharing-systems for electric vehicles (scooters and bikes), as well as users buying these vehicles for personal use will give OCB increased competition and an impetus to keep improving. By improving the existing BSS, there is the potential to reduce frustration for the city's populous and lead to greater benefit for the city.

Reference List

Almannaa, M. (2019). *Optimizing Bike Sharing Systems: Dynamic Prediction and Rebalancing Using Machine Learning and Statistical Techniques*. https://doi.org/10.13140/RG.2.2.26034.43202

Ashqar, H. I., Elhenawy, M., Almannaa, M. H., Ghanem, A., Rakha, H. A., & House, L. (2017). Modeling bike availability in a bike-sharing system using machine learning. *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 374–378. https://doi.org/10.1109/MTITS.2017.8005700

Bean, R., Pojani, D., & Corcoran, J. (2021). How does weather affect bikeshare use? A comparative analysis of forty cities across climate zones. *Journal of Transport Geography*, *95*, 103155. https://doi.org/10.1016/j.jtrangeo.2021.103155

Bell, D., & Gaillard, F. (2017). Epoch (machine learning). In *Radiopaedia.org*. Radiopaedia.org. https://doi.org/10.53347/rID-56141

Biau, G., & Scornet, E. (2016). A random forest guided tour. *TEST*, *25*(2), 197–227. https://doi.org/10.1007/s11749-016-0481-7

Boufidis, N., Nikiforiadis, A., Chrysostomou, K., & Aifadopoulou, G. (2020). Development of a station-level demand prediction and visualization tool to support bike-sharing systems' operators. *Transportation Research Procedia*, *47*, 51–58. https://doi.org/10.1016/j.trpro.2020.03.072

Chai, X., Guo, X., Xiao, J., & Jiang, J. (2021). Analysis of spatiotemporal mobility of shared-bike usage during COVID-19 pandemic in Beijing. *Transactions in GIS*, *25*(6), 2866–2887. https://doi.org/10.1111/tgis.12784

Chollet, F. (2015). *Keras* [Python].

Cutler, D. R., Edwards, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., & Lawler, J. J. (2007). RANDOM FORESTS FOR CLASSIFICATION IN ECOLOGY. *Ecology*, *88*(11), 2783–2792. https://doi.org/10.1890/07-0539.1

DeMaio, P. (2009). Bike-sharing: History, Impacts, Models of Provision, and Future. *Journal of Public Transportation*, *12*(4), 41–56. https://doi.org/10.5038/2375-0901.12.4.3

Drevland, L., & Finseth, P. (2018). *Evaluating Machine Learning Methods for City Bike Demand Prediction in Oslo*. Norwegian University of Science and Technology.

Du, Y., Deng, F., & Liao, F. (2019). A model framework for discovering the spatio-temporal usage patterns of public free-floating bike-sharing system. *Transportation Research Part C: Emerging Technologies*, *103*, 39–55. https://doi.org/10.1016/j.trc.2019.04.006

Feng, Y., & Wang, S. (2017). A forecast for bicycle rental demand based on random forests and multiple linear regression. *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, 101–105. https://doi.org/10.1109/ICIS.2017.7959977

Fu, R., Zhang, Z., & Li, L. (2016). *Using LSTM and GRU neural network methods for traffic flow prediction* (p. 328). https://doi.org/10.1109/YAC.2016.7804912

Garcia-Gutierrez, J., Romero-Torres, J., & Gaytan-Iniestra, J. (2014). Dimensioning of a Bike Sharing System (BSS): A Study Case in Nezahualcoyotl, Mexico. *Procedia - Social and Behavioral Sciences*, *162*, 253–262. https://doi.org/10.1016/j.sbspro.2014.12.206

Guo, R., Jiang, Z., Huang, J., Tao, J., Wang, C., Li, J., & Chen, L. (2019). BikeNet: Accurate Bike Demand Prediction Using Graph Neural Networks for Station Rebalancing. *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 686–693. https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00153

Kim, K. (2018). Investigation on the effects of weather and calendar events on bike-sharing according to the trip patterns of bike rentals of stations. *Journal of Transport Geography*, *66*, 309–320. https://doi.org/10.1016/j.jtrangeo.2018.01.001

Kubaľák, S., Kalašová, A., & Hájnik, A. (2021). The Bike-Sharing System in Slovakia and the Impact of COVID-19 on This Shared Mobility Service in a Selected City. *Sustainability*, *13*(12), 6544. https://doi.org/10.3390/su13126544

Liu, J., Sun, L., Chen, W., & Xiong, H. (2016). Rebalancing Bike Sharing Systems: A Multi-source Data Smart Optimization. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1005–1014. https://doi.org/10.1145/2939672.2939776

Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, *54*, 187–197. https://doi.org/10.1016/j.trc.2015.03.014

Murphy, A., & Gaillard, F. (2017). Batch size (machine learning). In *Radiopaedia.org*. Radiopaedia.org. https://doi.org/10.53347/rID-56140

Nankervis, M. (1999). The effect of weather and climate on bicycle commuting. *Transportation Research Part A: Policy and Practice*, *33*(6), 417–431. https://doi.org/10.1016/S0965-8564(98)00022-6

NCCS. (n.d.). *Klimaservicesenter*. Retrieved 12 May 2022, from https://seklima.met.no/observations/

Nkikabahizi, C., Cheruiyot, W., & Kibe, A. (2022). Chaining Zscore and feature scaling methods to improve neural networks for classification. *Applied Soft Computing*, *123*, 108908. https://doi.org/10.1016/j.asoc.2022.108908

Norwegian Digitalisation Agency. (2020). *Norwegian Licence for Open Government Data (NLOD) 2.0*. https://data.norge.no/nlod/en/2.0/

Oslo Municipality. (2022). *Status reports on coronavirus measures*. Oslo Kommune. https://www.oslo.kommune.no/english/coronavirus/status-reports-on-coronavirus-measures/

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., & Cournapeau, D. (2011). Scikit-learn: Machine Learning in Python. *MACHINE LEARNING IN PYTHON*, 6.

Stanford. (n.d.). *CS231n Convolutional Neural Networks for Visual Recognition*. Retrieved 23 June 2022, from https://cs231n.github.io/neural-networks-3/

Team Keras. (n.d.). *Keras documentation: Model training APIs*. Retrieved 11 June 2022, from https://keras.io/api/models/model_training_apis/

Wang, B., & Kim, I. (2018). Short-term prediction for bike-sharing service using machine learning. *Transportation Research Procedia*, *34*, 171–178. https://doi.org/10.1016/j.trpro.2018.11.029

Xu, C., Ji, J., & Liu, P. (2018). The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets. *Transportation Research Part C: Emerging Technologies*, *95*, 47–60. https://doi.org/10.1016/j.trc.2018.07.013

Yang, H., Xie, K., Ozbay, K., Ma, Y., & Wang, Z. (2018). Use of Deep Learning to Predict Daily Usage of Bike Sharing Systems. *Transportation Research Record: Journal of the Transportation Research Board*, *2672*(36), 92–102. https://doi.org/10.1177/0361198118801354

Zeng, M., Yu, T., Wang, X., Su, V., Nguyen, L., & Mengshoel, O. (2016). *Improving Demand Prediction in Bike Sharing System by Learning Global Features*.