# Master Thesis

Optimizing Bike Sharing System Flows using Graph Mining, Convolutional and Recurrent Neural Networks

---


Davor Ljubenkov (910418-3018)

davorl@kth.se


---


Academic Examiner: Šarūnas Girdzijauskas

Academic Supervisor: Amir Hossein Payberah

External Supervisors: Carlo Ratti, Fábio Duarte, Paolo Santi



Degree program: TIVNM - DASC

Subject department: EECS

Course code: II226X

EIT Digital Master School

June 3, 2019


# Abstract

A Bicycle-Sharing System (BSS) is a popular service scheme deployed in cities of different sizes around the world. And although docked bike systems are its most popular model used, it still experiences a number of weaknesses that could be optimized by investigating bike sharing network properties and evolution of obtained patterns. Efficiently keeping bicycle-sharing system as balanced as possible is the main problem and thus, minimizing or predicting the manual transportation of bikes across the city is the main objective in order to save logistic costs for the operating companies. The purpose of this thesis is two-fold; Firstly, it is to visualize bike flow using data exploration methods and statistical analysis to better understand the mobility characteristics with respect to distance, duration, time of the day, spatial distribution, weather circumstances, and other attributes. Secondly, by obtaining flow visualization it is possible to focus on specific directed sub-graphs containing only those pairs of stations whose mutual flow difference is the most asymmetric. By doing so, we are able to use graph mining and machine learning techniques on these unbalances stations. Identification of spatial structures and their structural change can be captured using convolutional neural network (CNN) that takes adjacency matrix snapshots of unbalanced sub-graphs. Generated structure from the previous is then used in the LSTM recurrent neural network in order to find and predict its dynamic patterns. As a result, we are predicting the bike flow for each node in the possible future sub-graph configuration which in turn informs bicycle-sharing system owners in advance to plan accordingly which prospective areas they should focus on and how many bike relocation phases are to be expected. Methods are evaluated using k-fold cross validation, RMSE and MAE metrics. Benefits are identified both for urban city planning and saving money (and time) for bike sharing companies.

**Keywords:** Data Science, Data Visualization, Bike-Sharing Systems, Graph Mining, Time Series Prediction, Machine Learning, Deep Learning, Recurrent Neural networks, Convolutional Neural Networks

# Acknowledgment

//TO-DO...

# Contents


**1 Introduction** **5**
- 1.1 Problem . . . 6
- 1.2 Purpose . . . 7
- 1.3 Goals . . . 7
- 1.4 Hypotheses . . . 8
- 1.5 Ethical Considerations . . . 8
- 1.6 Sustainability . . . 9
- 1.7 Limitations . . . 9
- 1.8 Thesis Outline . . . 9

**2 Related Work** **10**
- 2.1 Spatiotemporal Patterns . . . 10
- 2.2 Operations Research and Optimization of Docks . . . 10
- 2.3 Collaborative Visual Analytics . . . 11
- 2.4 Community Structures . . . 11
- 2.5 Comparing Cycling Patterns . . . 11
- 2.6 Mobility Prediction using Random Forest . . . 12
- 2.7 Mobility Prediction using Recurrent Neural Networks . . . 13
- 2.8 Predicting Station Level Demand using Recurrent Neural Networks . . . 13

**3 Data Exploration & Statistical Analysis** **14**
- 3.1 Data Preprocessing . . . 14
- 3.2 Framework and Libraries . . . 15
- 3.3 Case Study . . . 15
- 3.4 Mobility Flows . . . 19

**4 System Architecture** **22**

**5 Predicting Dynamic Patterns with RNN** **22**
- 5.1 RNN Data Preparation . . . 23
- 5.2 Linear Regression . . . 23
- 5.3 ARIMA . . . 25
- 5.4 Simple RNN . . . 29
- 5.5 Deep RNN . . . 31
- 5.6 RNN LSTM . . . 32
- 5.7 RNN Validation Metrics . . . 33

**6 Identifying Spatial Structures with CNN** **34**
- 6.1 Adjacency matrices . . . 35
- 6.2 Motivation . . . 40
- 6.3 Convolutional neural network . . . 40

**7 Discussion** **45**
7.1 Results . . . . . 46

**8 Conclusion** **46**
8.1 Future Work . . . . . 46

# Abbreviations & Definitions

ACF = Auto-Correlation Function
ADAM = ADAptive Moment estimation
API = Application Programming Interface
ARIMA = Auto-Regressive Integrated Moving Average
BN = Batch Normalization
BPTT = Back-Propagation Through Time
BSS = Bike Sharing Scheme (Service)
CHS = Cycle Hire Scheme
CNN = Convolutional Neural Network
CV = Cross Validation
D.C. = District of Columbia
DTW = Dynamic Time Warping
EDA = Exploratory Data Analysis
ELU = Exponential Linear Unit
FNN = Feedforward Neural Network
GPA = Generalized Procrustes Analysis
GRU = Gated Recurrent Units
GUI = Graphical User Interface
HCA = Hierarchical Cluster Analysis
IP = Integer Programming
LCHS = London Cycle Hire Scheme
LSTM = Long Short-Term Memory
MAE = Mean Average Error
MAPE = Mean Absolute Percentage Error
MIT = Massachusetts Institute of Technology
ML = Machine Learning
MSE = Mean Absolute Error
OD = Origin-Destination
OLS = Ordinary Least-Squares Regression
PCA = Principal Component Analysis
PIP = PIP Installs Packages
PLoS = Public Library of Science
RBM = Restricted Boltzmann Machine
ReLU = Rectified Linear Unit
RSS = Residual Sum of Squares
RF = Random Forest
RMSE = Root Mean Squared Error
RMSLE = Root Mean Squared Logarithmic Error
RNN = Recurrent Neural Network (not to be confused with Recursive Neural Networks)
RSS = Residual Sum of Squares
SCL = Senseable City Lab(oratory)
TF = TensorFlow

TfL = Transport for London
T-SNE = T-distributed Stochastic Neighbor Embedding
UDF = User Dissatisfaction Functions
VGP = Vanishing Gradient Problem

# List of Figures

//TO-DO...

# List of Tables

//TO-DO...

# 1 Introduction

A Bicycle-Sharing System (BSS) is a popular service scheme deployed in cities of different sizes around the world. It is a service in which bicycles are made available for shared use to individuals on a short term basis for free or for a price. The user borrows and returns the bike by placing it in a "dock". If the service doesn't use docks, then it is referred to as "dockless". Using these Bike Sharing systems, people rent a bike from one location and return it to a different or same place on need basis. People can rent a bike through membership (mostly regular users) or on demand basis (mostly casual users). This process is controlled by a network of automated stations across the city.

First BSS had its inception in 1965, when Amsterdam city councilman Luud Schimmelpennink proposed it as a way to reduce automobile traffic in the city center. After the city council rejected the proposal, Schimmelpennink's supporters distributed fifty donated white-painted bikes for free usage around the town. (The bike sharing planning guide, ITDP) The police, however, impounded the bikes, claiming that unlocked bikes incited theft [1].

In 1991, a second generation BSS was conceived in Denmark, offering a few hundred coin-operated bikes. In 1996, a third generation, now based on magnetic cards and several technological advances was initiated in England and continued to evolve within following years. But it was only when Lyon in 2005, and later Paris in 2007. made their wise deployments of several thousand shared bikes that these systems started to become known worldwide. A few years after that, similar programs spread throughout other continents and, now, there are estimates that more than 18 million bikes are actively used in a variety of BSS systems worldwide.

An exponential growth has been observed in developed and developing countries, in large and small, dense and sprawling cities, One of the main arguments for the implementation of BSS is that they provide an effective alternative for the first- and last-mile problem, mainly when integrated with public transport. Data from the USA Department of Transportation's 2017. National Household Travel Survey indicates that 35% of all car trips in the US were shorter than 2 miles (3.218688 kilometers), and almost 50% or half of all car trips were less than 3 miles (4.828032 kilometers) - a distance that could usually be covered with a reasonable amount of cycling. Thus, there are plenty of motivations and opportunities for the expansion of such systems both to new cities and within the cities that already have an existing basic BSS implementation. BSS have been assembled around the world in *ad hoc* manners - with little or no scientific, evidence-based planning. The complex dynamics of such systems and their interaction with the city life rhythm and other means of transportation is not yet fully understood. There are multiple business models, and public or private forms of funding BSS, Within the past few years, several BSS companies have gone bankrupt and most cities worldwide are still reluctant in considering bike sharing as an integral part of their mobility portfolio. However, with more data obtained, dynamics of such systems are slowly being

investigated by scientists using research methods that inspect mobility flows, optimization algorithms and predictions.

[**?**]
The real expansion did not take the place until the $21^{st}$ century when first municipal plans or larger scale business ventures that offered service as we know of today had been created. In general, cycling as a means of transportation in modern cities has grown significantly in the past ten years. The appearance of large-scale bike-sharing systems and an improved cycling infrastructure are two of the factors that enabled this growth. An increase in non-motorized modes of transportation makes our cities more humane, decreases pollution, traffic, and improves the quality of life. In many cities around the world, urban planners and policymakers are viewing cycling as a sustainable way of improving urban mobility. Nevertheless, most cities still rely on $20^{th}$ century tools and methods for planning and policy-making. Recent technological advances enabled the collection and analysis of large amounts of data about urban mobility, which can serve as a solid basis for evidence based decision making.

The use of bicycles for short trips (defined as trips with distance below 5 kilometers) in medium to large cities for commuting, occasional, and leisure trips presents multiple proven benefits at the global, local, and personal level, In global terms, substituting motor vehicles with bicycles reduces carbon emission and energy consumption as well as negative environmental impact. With respect to local benefits to the city, an increase in the number of cycling trips in substitution of motorized trips helps mitigating traffic congestion, decreasing air and noise pollution, and the amount of required parking space. In addition, it also brings several personal benefits for both mental and physical health. Research shows that commuting to work on a bike also presents an advantage in relation to other active modes of transportation such as walking, since its higher cardiorespiratory intensity is associated with health benefits. However, both pedestrians and cyclists are more exposed to accidents and injuries compared to a car or transit passengers. In the case of cycling, the risk is aggravated when dedicated bike lanes are not available.

## 1.1 Problem

There are several problems that arise with such sharing systems.

Firstly, there is a fleet management problem. In order to keep BSS as balanced as possible, bikes are manually transported across the city at peak times. In priority areas docking stations are continually replenished with bikes or the bikes being continuously removed from docking stations. (Roger Beecham, Jo Wood, Audrey Bowerman - 2013) This is an expensive endeavour that BSS owners have to enforce in order for the system to run smoothly. This cost function are hard to calculate and is currently being optimized by the operations research scientists. Their model focuses on the number of docks,

where each station is revised and later physically changed by adding or removing the docks. However, this method does not take the full-fledged prediction into an account. More specifically, every day in the week is observed as indistinguishable property-wise from the same day in any other week, month or year regardless of any external factors such as the weather, holidays, special events etc.

Secondly, in previous years the amount of data which was made available for researchers to work with was not sufficient enough, and in most cases the data time-span period covered was not more than a couple of months or up to a year. Of course, this varies on the specific BSS whereabouts but even the older systems investigated were not explored to their full potential.

Moreover, even though most of the visualization methods have already been covered in existing papers, there had been a lack of comparative studies that would try and investigate things such as: the underlying distribution laws of graph structures, prediction performances, visualization patterns or conclusions drawn about the mobility flow scenarios.

## 1.2 Purpose

The academic purpose of this work is to (1) explore specific properties of Bike Sharing Systems through a statistical exploration analysis, and (2) to assess the relative strengths of different implementations of predictive models and their potential combination.

Analogously, the commercial purpose is to (1) obtain a model with powerful predictive capabilities, and to (2) reduce costs of bike relocation strategy by using an efficient label prediction, and (3) obtain a high-quality correctness score.

## 1.3 Goals

The goals of the work, in chronological order, is to:

- prepare and clean the Blue Bikes Boston Bike Sharing dataset
- find suitable secondary data such as weather and use data wrangling methods to combine it with the primary data source
- use data exploration, statistical analysis and visualization to investigate bike sharing networks in collaboration with other researchers in order to get a better domain knowledge of bike sharing systems and problems that need to be addressed
- compare different recurrent neural network prediction methods on the complete bike sharing dataset to find the best one to be used for data floe prediction where data flow is defined as the aggregated number of bike check-outs for each day

- define most unbalanced or asymmetric pairs of stations in the network for each month and create a subgraph containing this nodes stored as an origin - destination matrix
- use convolutional neural network to predict the label of the next subgraph that is most likely to emerge based on the preliminary data for the upcoming month
- utilize the chosen recurrent neural network on the output of the previous step to define the predicted flow and approximate the best strategy for the bike relocation in that specific configuration
- present the results by using appropriate validation metrics and conclude the thesis with some proposition and references for future work

## 1.4 Hypotheses

Prior to data exploration and uncovering variable relationships, it is necessary to gain the domain knowledge and use structured thinking about the problem. This form of problem inspection helps forming better features and eliminate possible biases. Some of the hypotheses that could influence bike demand:

Due to the hourly trend, a higher demand for bikes must exist during the rush hours. For example, late night period should have significantly lower demand compared to lunch hour.

On a daily trend scale, weekdays would need to have a much richer network compared to weekends or holidays.

Weather and season should highly influence bike demand numbers: rainy days, windy periods, higher humidity, and lower temperatures will probably have a positive correlation with bike demand. At least, this should be true in America and Europe. Things could be differently correlated in places with different climate like some Asian countries where correlation with humidity and temperature could be negative.

Some additional bike sharing influences could be city pollution levels or traffic congestion distribution.

However, main hypothesis to be examined in this thesis is the claim that we could use current bike sharing system data to predict future bike flow, especially for those stations that are considered to be problematic in a sense of their high relocation frequency. Of course, this prediction is expected to perform within a certain degree of accuracy.

## 1.5 Ethical Considerations

//TO-DO...

## 1.6 Sustainability

//TO-DO...

## 1.7 Limitations

The very first limitation is noticeable in the usage of exclusively docked bike systems data. Originally, it was planned to have a slightly broader study were dockless bikes would be investigated as well but that did not come to fruition as American and European companies that own such systems are not comfortable with sharing data. It is important to mention that even in the case of a successful collaboration with such companies, the process of obtaining data involves a complicated legal procedure and takes a couple of months in total which was not feasible regarding the time restrictions imposed upon the completion of this thesis in a timely manner.
Dockless data and policies differ in China, but such approach was not taken into consideration due to a high volume of papers already written and specifically based upon this areas. Also, dockless systems, or fourth generation bike systems, are much more popular in China compared to the rest of the world. There are over 30 private companies operating there, while dockless bikes are still in an experimental phase in both of the Americas and Europe.

Regarding the second limitation, it is not possible to produce the model with a perfect prediction accuracy or retrieve the highest precision of label assignment. This means that there will always be an error present depending on the volume of our data, implementation details of the specific model used, computation complexity and a variety of other variables, some of which are stochastic in their nature and therefore, out of our control. Still, establishing a performance baseline, defining model setups, and knowing our lower and upper bound is supposed to make a define our limitations empirically and mitigate any unwanted performances.

## 1.8 Thesis Outline

The following thesis report is organized as follows:
This first chapter introduced a short overview of Bike Sharing Systems and its ...

The second chapter reviews the recent related work in the area of Bike Sharing Systems highlighting both strengths and shortcomings, as well as how the presented work is connected to the work contained in this very thesis.

Third...

# 2 Related Work

## 2.1 Spatiotemporal Patterns

In the paper written by Grant McKenzie (2018)[2], docked and dockless bike sharing system had been compared. Because of a sudden explosive growth in dockless bike-sharing services, limited time was provided for municipal governments to set regulations and assess their impact on docked bikesharing programs. This was a motivation behind the paper to presents an exploratory understanding of the differences in activity patterns between these two services. Results can be used to better inform urban planners, transportation engineers, and the general public. However, paper focuses exclusively on Washington, D.C. and most of the analysis is just exploratory, while results of the paper are preliminary due to lack of data. Comparisons were made between Lime (dockless) and Capital Bikeshare (docked). Lime is a private company and Capital Bikeshare is owned by the municipial government of D.C. (also Virginia and Maryland). Data analyzed included only a month of March in 2018 (238,936 individual trips). Temporal aspects were observed by calculating: mean duration, median duration, bike trip aggregation to the nearest hour of a week and independently normalized, with pattern subtraction. For spatial aspects Voronoi tessellation was used to partition town map into polygons, with subtraction and intersection of these polygons with land use data from D.C.s Office of Planning. Regrading the network analysis, K-means algorithm was used for clustering the dockless locations with a number of clusters, and the conclusion made was that the existing docks are well situated. On top of that, Dijkstras algorithm for routing analysis was also implemented. In conclusion, suggestions made mentioned that other modes of transportation should be taken into account, as well as behavioral motivation of users for selecting certain services.

## 2.2 Operations Research and Optimization of Docks

Daniel Freund et al.(2019)[3] uses a case study of Motivate (owned by Lyft and managing a vast number of U.S. Bike Sharing systems such as Blue Bikes, Citi Bike, etc.) and collaborates with Cornell University in order to optimize the number of docks in New York. This is done from the point of view of operations research viewpoint and uses optimization models. This is done with stochastic modeling, defining UDFs (User Dissatisfaction Functions) being a convey function, Poisson processes, M/M/1 queues, integer programming models, discrete gradient descent algorithm, and Kolmogorov's backward equation. Optimization formulation is written like this:

$$\begin{aligned} minimize_{\vec{b}} \quad & \sum_i c_i(b_i, K_i) \\ s.t. \qquad & \sum_i b_i \qquad \leq B, \end{aligned}$$

$$\forall i \qquad 0 \leq b_i \qquad \leq K_i.$$

where the number of docks at Station i are denoted by $K_i$, number of bikes are $b_i$ , the UDF is $c_i(b_i, K_i)$, and the total number of bikes available is B. When the docks are being moved, $K_i$ becomes the decision variable in addition to $b_i$ in which case $\bar{K}_i$ is the number of docks at each station and we can write the constraint like:

$$\sum_i |K_i - \bar{K}_i| \leq 2k$$

In conclusion, this technique is very successful and already implemented in New York. However, the author himself admits that this method does not differentiate the temporal aspect which can affect bike stations when predicting the future bike tidal flows and does not taken any secondary datasets into account.

## 2.3 Collaborative Visual Analytics

Beexham et al.(2014)[4] discuss automatic label classification of commuting behavior and inferring workplace of individuals in London (LCHS). Methods that are described include: weighted mean-centres, K-means clustering, kernel density estimation and community detection. They identify a fleet management problem and closed peak-time 'loops' but do not attempt to solve it. Contributions are present in deriving customers workplace areas and labelling commuting journeys, based on a spatial analysis of travel behaviours. Data observed includes trips between September 14 2011. and September 14 2012. which makes a total of 5,048,000 journeys. Some new attributes have been created: e.g. distance from users home to the closest docking station, RecencyFrequency (RF) segmentation. Regrading the observation of spatio-temporal analysis they used lines on a map (visual saliency) and fluctuations for each day of the week. Workplace centres for each cyclist have been derived by calculating: frequency of weighted centroids for docking station locations, using K-means clustering, hierarchical cluster analysis (HCA), and density-estimation method[5].

## 2.4 Community Structures

Munoz-Mendez et al.(2018) address a time-varying networks of bike stations and communities in London, where different motifs (loop, chain, star) and temporal evolution dynamics with extended time windows could potentially provide deeper insights into inherent relationships of spatially heterogeneous nodes (stations) or sub-networks (communities). They also suggest that instead of pure unsupervised learning, extended layers of urban systems should be used with an amenities to draw meaningful conclusions.

## 2.5 Comparing Cycling Patterns

Sarkar et al.(2015)[6] identified the problem of balancing between system usage and demand, which leads to a lack of available bicycles or free parking spaces at stations at

various times of the day. Data used i this studies had time span of 4.5 months, included 10 different cities with a total of 996 stations and 108 samples. Focus of this paper was solely on the fullness of stations and not on mobility flows. Unsupervised learning was used to show the intrinsic similarities between the cities by utilizing predictability of stations occupancy and comparing cross-city error for each. What they found was that heterogeneity is observed only in bigger systems. Random forest and neural network were used to compare the accuracy of forecasting how many bicycles will be at a given station and time.
Their paper also discusses how studies of shared bicycle systems have recently appeared in the data mining literature, and how Froehlich et al.(2009)[7] were the first to apply clustering techniques and forecasting models to identify patterns of behaviour in stations in Barcelona's 'Bicing' system, explaining results according to stations location and time of day. A recurring conclusion across analyses is that spatiotemporal system usage patterns are tied to, and reflect, city-specific characteristics. By focusing on single cities systems, these works seem to indicate that each city has a unique pattern, and that forecasting algorithms applied to each one may not be generalisable across the world.
OBrien et al.(2014)[8] and Austwick et al.(2013)[9] characterise systems at the city-level, comparing them in terms of system size (both by station count and geographic area), daily usage, and compactness; they build a hierarchy of cities that share similar characteristics and apply community detection algorithms to analyse similarities within systems.
In the paper examined, pairwise ground distances are computed between all locations recorded for a single station using the Haversine formula (Robusto 1957). Aggregate occupancy time series are calculated with Pearson correlation used for comparison week-day and weekend. Hierarchical clustering with an agglomerative strategy (bottomup approach) was used to identify which individual stations share similar behavioural traits across different cities. Selected metric to measure the similarity between station vectors was used and distance metric based on the dynamic time warping (DTW) algorithm (Berndt and Clifford 1994). Finally, they mention a technique for finding the optimal alignment of two temporal sequences and also a 1-h SakoeChiba band (1978).

## 2.6 Mobility Prediction using Random Forest

motivate their work by explaining that the primary issue for both users and operators is the uneven distribution of bicycles due to the demand and supply changing trends. This demands better bike re-balancing strategies which depend highly on bicycle modeling and prediction. Contribution of their work is two-fold: spatio-temporal bicycle mobility model based on historical data, and traffic prediction model mechanism per each station with sub-hour granularity. For the evaluation relative error of an obtained prediction is used. The paper focuses on the city Hangzhou in China with around 2800 stations and 103 million records in a time span of one year. Methods used include: spatio-temporal modeling, estimating the number and time of check-ins at different stations, and using random forest theory to predict check outs given time, weather, as well as real-time bike availability.

## 2.7 Mobility Prediction using Recurrent Neural Networks

paper tackles bike sharing demand and supply by implementing a real-time predicting method, community detection, and a 2-layer LSTM RNN model for Citi Bike System in New York and Jersey City. In addition to the bike data, meteorology data is used as a secondary dataset. Training set includes year 2017, while test set consists of first three months in 2018. In total, 800 stations are identified. Regarding the evaluation, RMSE had been used. Motivation for the usage of deep LSTM is because it can handle a large amount of data in a reasonable small amount of time. One of the suggestions is to use these predictions in order to distribute the number bikes specifically to each station.

## 2.8 Predicting Station Level Demand using Recurrent Neural Networks

Again, in NAMEOFTHEPAPER, bike shortage problem due to uneven bikes distribution is in the focus and efficient online balancing strategy is proposed as a solution. Unlike other papers where most researches are about predicting global rental demand or rental demand at cluster level, this paper considers station level demand prediction which could be more beneficial. Proposed architecture makes predictions for all stations at once. New York Citi Bike dataset is used with 8,081,216 individual trips. Regarding the methods, RNN is used on station level for both rental and return, loss function uses backpropagation through time (BPTT) and Vanishing Gradient Problem.
Data Exploration
Correlation: Weather and Rentals
BASELINE APPROACHES:
Ordinary Least-Squares Regression (OLS)
Random Forest (RF) - with 50 estimators
Feedforward Neural Network (FNN) - 4 layers with ReLU (Rectified Linear Unit) activation functions
EVALUATION:
Root Mean Squared Error (RMSE), Mean Absolute Error (MAE)

# 3 Data Exploration & Statistical Analysis

Section 3.1 gives an overview of how input data was pre-processed. Section 3.2 describes the technical set up required for running all these experiments and deploying the model to production. The following data and architecture in this Chapter 3 had been investigated and analyzed in collaboration with the MIT visiting researcher Fábio Kon at SCL.

## 3.1 Data Preprocessing

To illustrate the methodology of this case study, 7 years of data from the Boston Blue-Bikes bike-sharing system were used. Bike sharing data was collected from the Bluebikes website, the largest Boston bike-sharing provider. Boston is a relatively bike friendly city, having received a silver medal award from the League of American Bicyclists in 2017. From 2007 to 2014, the bycicle lane mileage in Boston went from 0.03 miles (0.048 kilometers) to 92 miles (148.06 kilometers), with a decrease in bicycle accidents around 14% per year. Boston's original bike-sharing system, Hubway, was launched in 2011 and it has been growing since then. In 2018, its name changed to BlueBikes and it now has over 1800 bicycles and 308 dock stations across Boston, Brookline, Cambridge, and Somerville. In the proposed analysis, nearly 8 million bike trips have investigated since the inception of the bike-sharing program.

Below is a list of bike sharing data attributed with information about how they were represented:

- "tripduration"
- "starttime"
- "stoptime"
- "start station id"
- "start station name"
- "start station latitude"
- "start station longitude"
- "end station id"
- "end station name"
- "end station latitude"
- "end station longitude"
- "bikeid"

- "usertype"
- "birth year"
- "gender"

## 3.2 Framework and Libraries

The tool implementing the proposed methodology is a distributed collection of open source Jupyter notebooks. Jupyter is a python module, and is available either preinstalled as an Anaconda module, or can be installed manually with pip. In the case of manual installation, the user will also need to install the modules pandas, numpy, and scipy. The Jupyter Notebook files, with extension ".ipynb", can be run either from Jupyter's GUI, or run from the command line inside the unzipped folder with Jupyter Notebooks. Running the second command on a windows machine may require adding the Python scripts directory to the PATH variable, located in the sub directory "Scripts" under the Python installation directory. All of the code had been written used Python programming language. Some of the additional libraries include: matplotlib, seaborn, ggplot, geoplotlib, folium, GeoPandas, scikit-learn.

## 3.3 Case Study

Initially obtained descriptive statistics for Boston Blue Bikes data helps us understand usage patterns extracted from the data between 2011 and 2018. In Figure 1, produced age, trip distance, duration, and speed histograms can be observed. Trip duration follows a log-normal distribution with a median of 10 minutes and with 75% of the trips taking under 16 minutes. On the other hand, the speed follows a Student's t-distribution, with men riding slightly faster than woman.

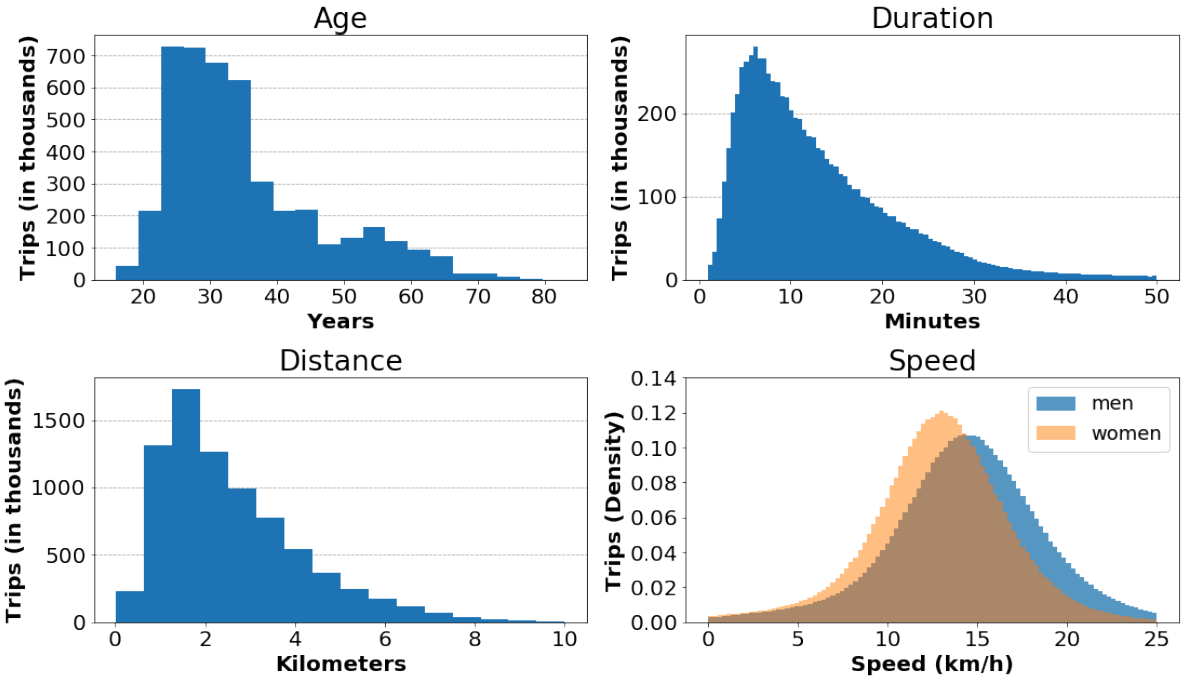


Figure 1: Descriptive Statistics for Boston Blue Bikes data

In Figure 2 we can see the evolution of the total number of trips per day for the entire bike sharing system. One can see both strong seasonal effects caused by the typical harsh winters in Boston, and the overall tendency for an increase in usage over the six years which is confirmed by the 12-month rolling average plotted. The men and women ratio shows not only that men use bike sharing more frequently but that the difference increases during the winter time. Finally, the figure also shows a slight increase in the proportion of female users in the past year. The cities of Boston, Cambridge, and Somerville have been improving the quality and extension of their cycling infrastructure. As women feel more comfortable and secure in the cycling tracks, the gap in usage for men decreases. However, it is still too soon to speculate if these will be a trend in the long run for the Boston area as well.

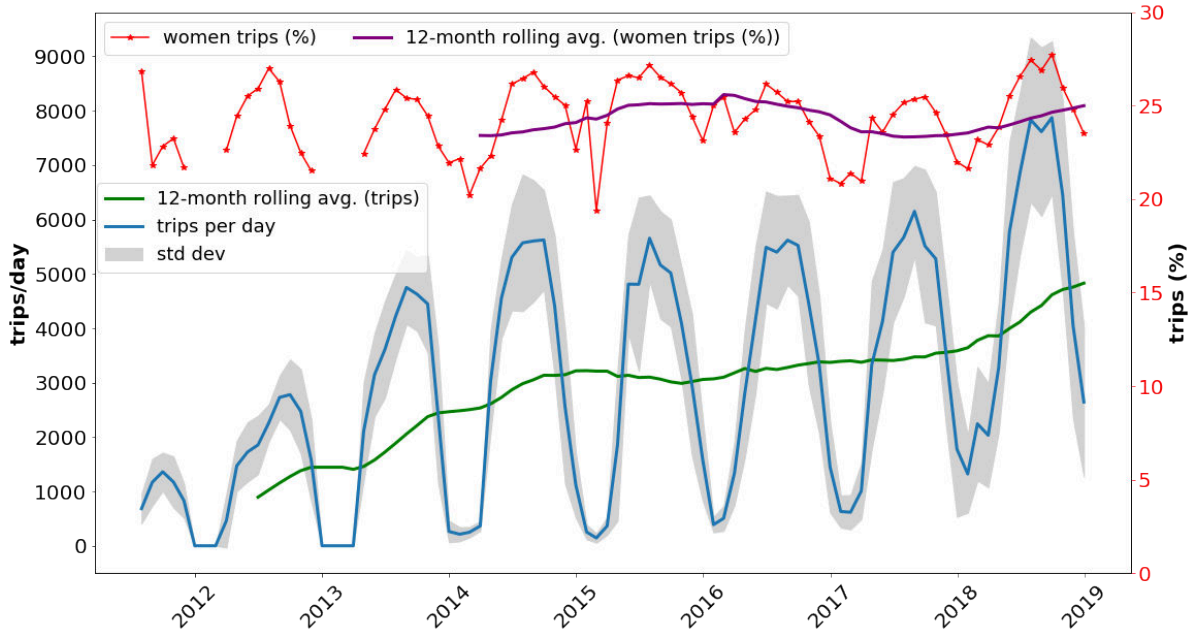


Figure 2: Evolution of trips from April 2013 to January 2019

For the analysis involving distances and speed, the road distance between two bicycle dock stations is estimated by using the GraphHopper API over OpenStreetMap map; in particular, the bike mode route planner is used, which provides bike-friendly routes. The bike routes suggested by the API are around 30% longer than the Euclidean distance, on average.

Using the calculated speed, it is possible to detect the evidence of rider reckless behaviour. The most common reason for cycling accidents and fatalities is to get hit by a car. Although car drivers are usually at fault for such accidents, according to the US Department of Transportation, from 2010 to 2015, the most common bicyclist action prior to fatal accidents was the cyclists failure to yield right-of-way (in 34.9% of cases). A city government, then, may wish to develop an educational campaign to decrease the number of cyclists that ride bike dangerously fast. Analyzing the dataset and selecting the trips whose average speed was over 20 km/h this analysis can be easily done. Given that the average speed of all trips is 13 km/h and that only 4.2% of the trips are above 20 km/h, we can consider that these fast trips have a large probability of being associated with cyclists riding dangerously fast. Profile of this speeders is as follows:

- 89% are men while only 11% are women
- 50% of the speeders are between 21 and 32 years old, and although speeders are present in all ages under 52 - the age range in which people have more tendency to drive dangerously fast is between 25 and 30

- The length of speedy trips is 20% longer than average and their duration is half that of an average of all trips
- A subscriber (usually, a resident) is 4.6 times more likely to be a speeder than just a customer (usually, a tourist)

## 3.4 Mobility Flows

Understanding where the major flows of cyclists are located within a city is the first step in providing urban planners with the knowledge required to draw a good mobility plan for urban cycling. Most previous work on BSS data analysis focuses on analyzing usage patterns of individual dock stations, without investigating the movements from one place to another, such as the origin-destination pairs of bike trips which can provide interesting insights on the punctual dynamics of the system.

Because stations are normally distributed unevenly across the city, investigating each individual station does not provide an overall picture of city mobility dynamics for the urban planner. In one of the studies, Zhou used a clustering alorithm to group together flows connecting dock stations in Chicago, identifying 378 relevant flows in the city for the year 2014 [10]; this is an interesting approach but showing so many flows to the user without any structure does not support policy making adequately. In addition, the computational complexity of the clustering algorithm might hinder the method's interactivity and fast usability.

For each trip, the location and time of origin - destination were used. Workdays present similar patterns among themselves but they differ greatly from weekends, so these classes can be treated separately. Within a single day, three different time periods are investigated: morning peak (from 7:00 to 10:00), lunch time (from 11:00 to 14:00) and afternoon peak (from 17:00 to 20:00) as their patterns differ significantly. Also, the average number of trips per hour in the dataset reduces significantly during the winter months.

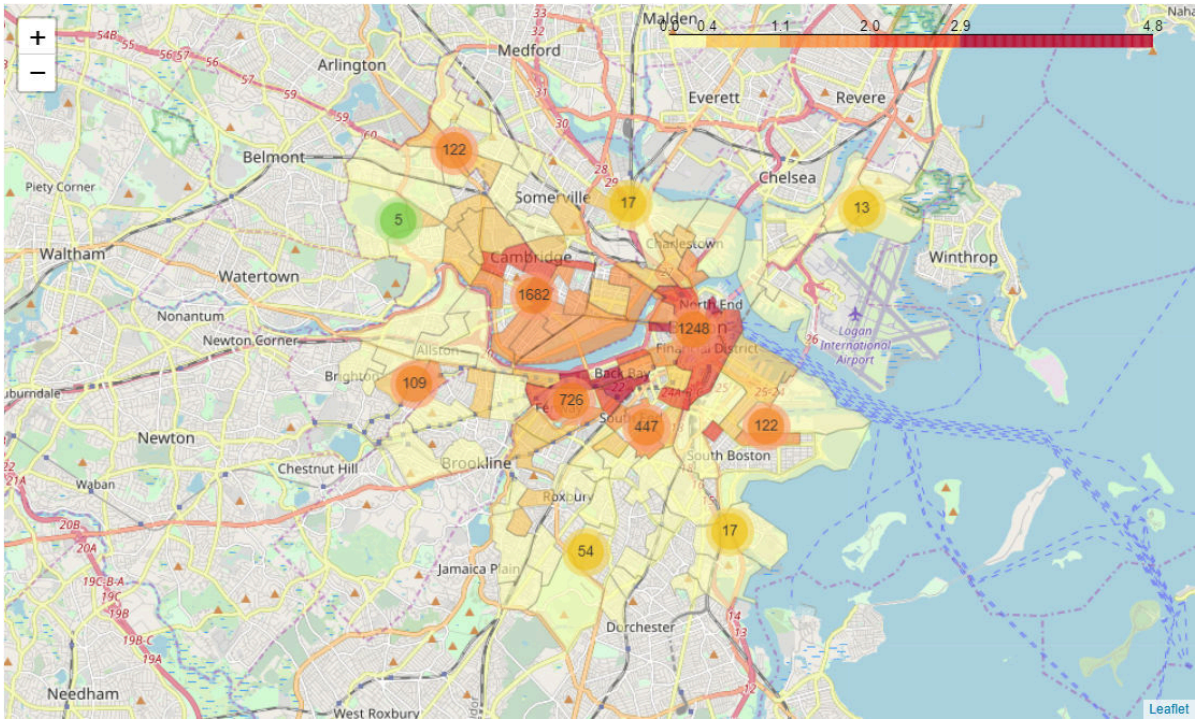


Figure 3: Morning trip Check-outs clustered by neighbourhoods for July 2018

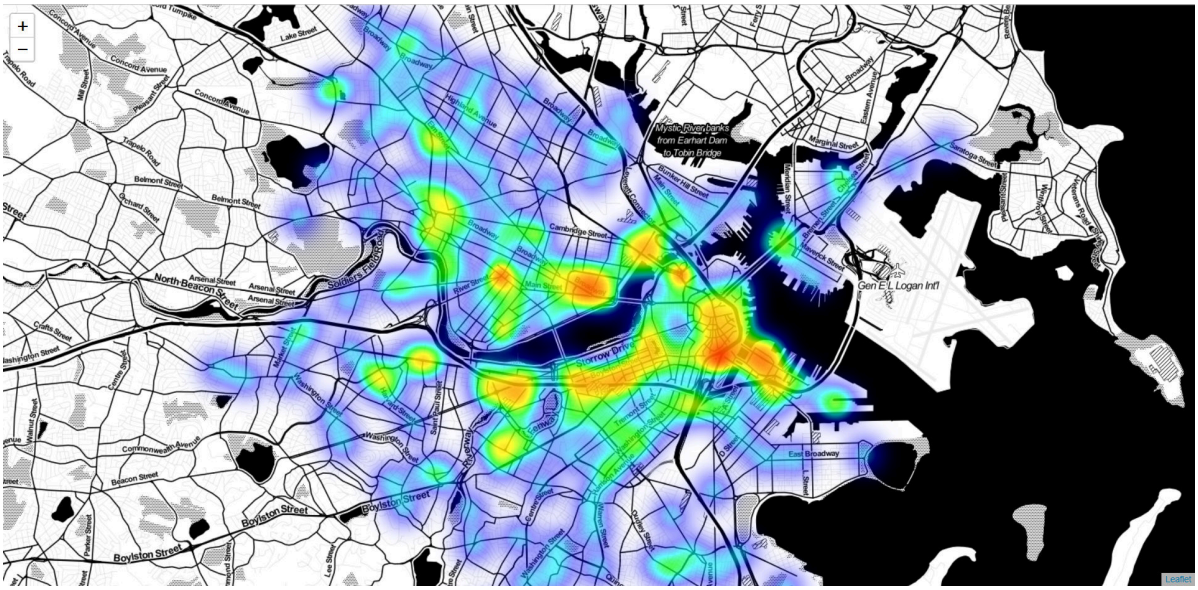


Figure 4: Morning trip Check-outs heat map for July 2018

Crucial visualization method is the one where mobility flows are represented as a directed graph. With the help of this method it is possible to define most pairwise asymmetric nodes which are of extreme importance in this thesis. Firstly, they contain the mobility flows most responsible for the unbalanced network which in turn, causes more frequent

need for bike relocation. Secondly, these nodes will be used as a sub-graph input for the CNN in Chapter 6 in order to gain insight of the future patterns that can be expected.

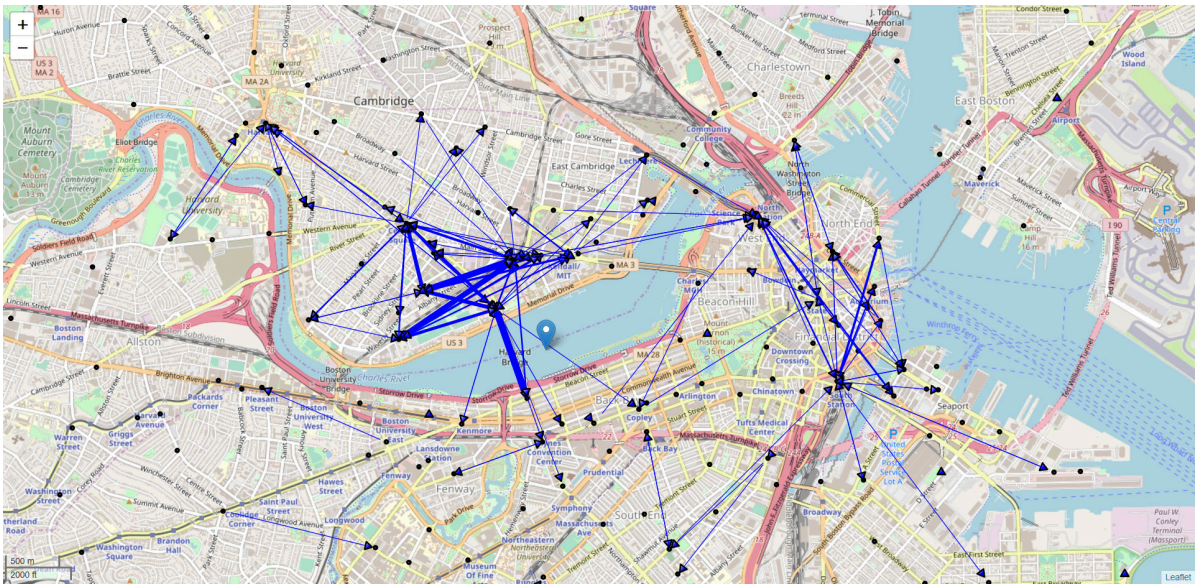


Figure 5: July 2018 Mobility Flows as a Directed Graph

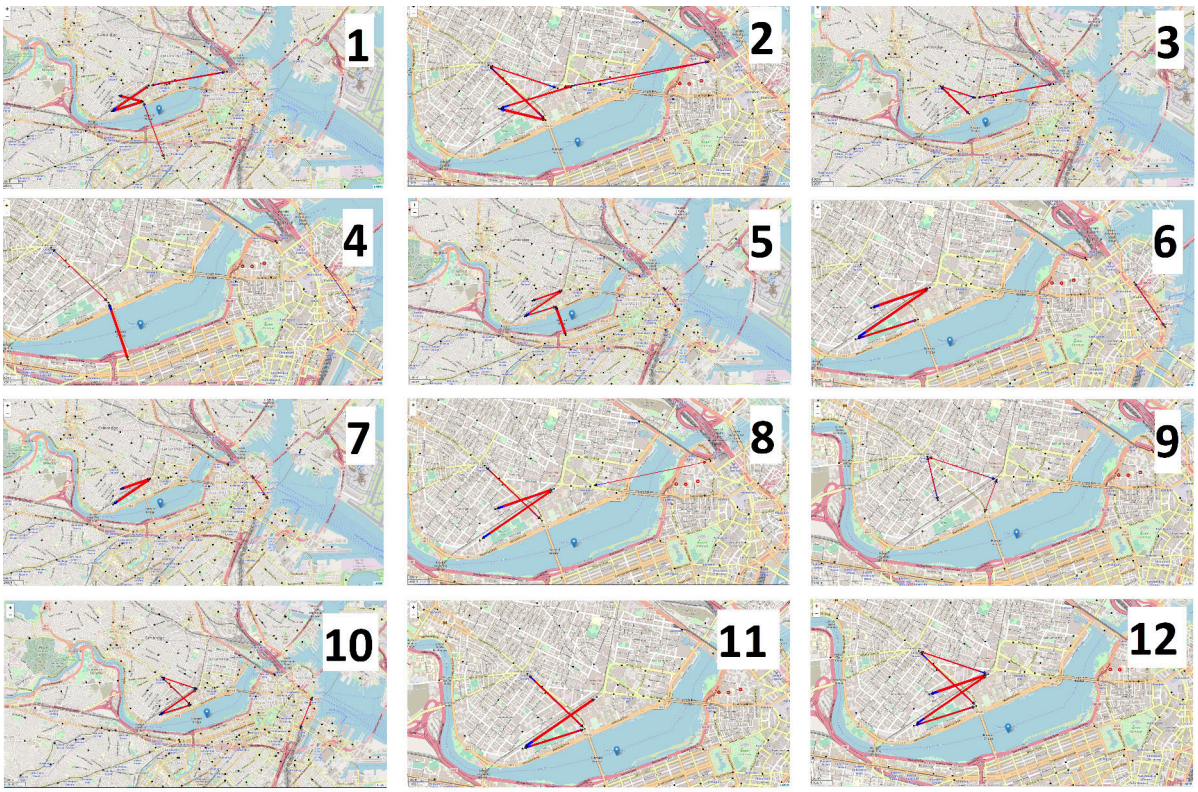


Figure 6: Most pairwise asymmetric

# 4 System Architecture

The following technologies were used to build the system for identifying spatial structures and predicting dynamic patterns of bike sharing networks:

TensorFlow (TF) - Machine Learning framework for Python

Keras -

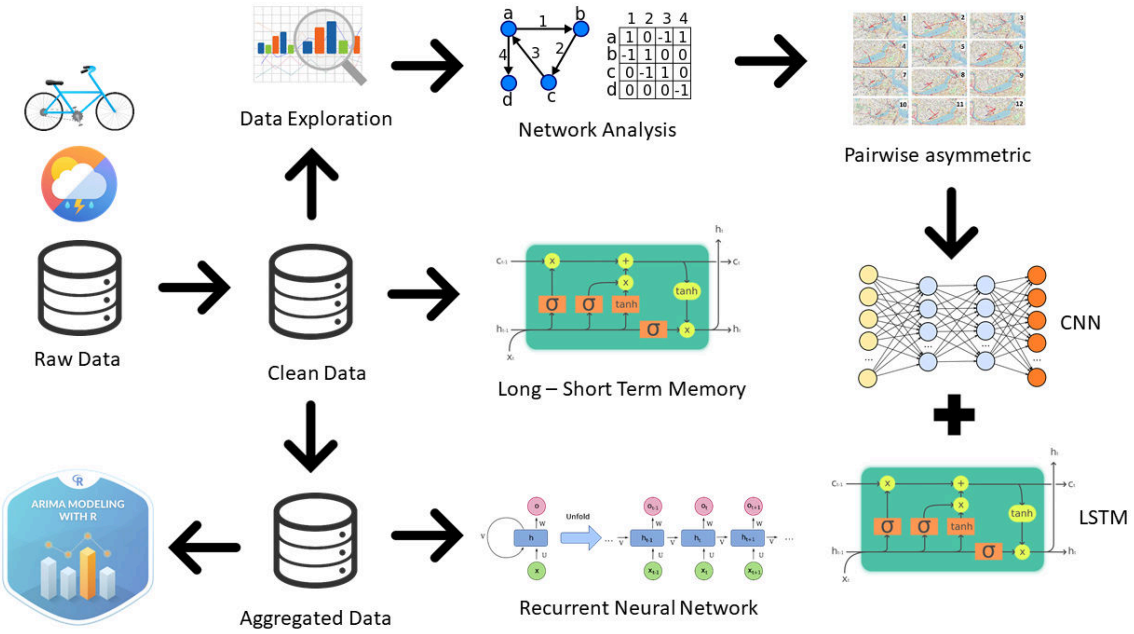


Figure 7: Proposed System Architecture & Pipeline

# 5 Predicting Dynamic Patterns with RNN

In this chapter a comparison between different predictive methods for time-series data will be examined. Primary dataset used is the aggregated number of bike check-outs for each day, and the secondary dataset contains weather details. Number of of bike check-outs, for reason of convenience, will be hereinafter referred to as **bike usage**. Best prediction method will be used for individual stations identified as the most unbalanced which are going to be obtained separately as an output of convolutional neural network in Chapter 6. The process of choosing the best prediction method will depend on the validation metrics retrieved and particular setting of parameters defined in the prediction model itself.

## 5.1 RNN Data Preparation

Primary dataset contains the Blue Bike data in the time-span between January of 2016 and March of 2018. In total, this accounts for 27 months of data worth, or 821 days. For the sake of simplification, only the number of bike check-outs (bike usage) is used and we will ignore other attributes. In addition, daily granularity is used which means that bike usage had been aggregated for each day independently based on the "starttime" or time of exact time of the bike check-out. This is denoted as variable "freq", while we can also use "freqscaled" which had been normalized by dividing each bike usage value of each day with the highest bike usage observed (that exact value is **7405**). This will create a range of values between zero and one, as for some neural networks it is sometimes easier to digest and process these normalized inputs.

Secondary dataset is a weather dataset previously acquired via "Kaggle"[1] website, but originally scraped from "Weather Underground"[2] platform. However, some of the original attributes were dropped out for the purpose of using it as an adequate input for recurrent neural network. For example, having an average temperature alongside high and low temperature seemed redundant, especially as it is so trivial to obtain average form the two latter mentioned extremes. Also, removing Event attribute is justified as we already have our information on snowfall and rainfall which is far more precise than just a boolean indicator of their presence. The produced weather dataset consists of: temperature (high and low), dew point (high and low), humidity (high and low), visibility (high and low), wind (high and average), high wind gust, snowfall, and precipitation. Now, having this dataset, we would need to examine which of this attributes are the ones that correlate with the bike usage frequency the most. Simply by performing linear regression between "freqscaled" representing the bike usage and each one of the attributes, it is possible to empirically decide which attributes are more suitable to be kept.

## 5.2 Linear Regression

For each pair between bike usage and one of the thirteen attributes, an isolated scatter plot will be produced showing all the points representing a relation between the two variables. In order to plot correctly, each of the attributes must be scaled by dividing their value with the maximum attribute value in existence. Then, a simple regression model will be applied and regression line can be observed on the plot. To evaluate which combination of bike usage and different attributes is the one with highest correlation (positive or negative), a coefficient of determination is defined and denoted as R squared. The value of R squared is typically taken as "the percent of variation in one variable explained by the other variable, or the percent of variation shared between the two variables."[11] As a rule of thumb that correlation coefficient value between 0.7 and 1.0 are representing the strong linear relationship, and that means that only temperature

[1] https://www.kaggle.com/
[2] https://www.wunderground.com/

attributes (high and low) can be used as relevant factors. In **Figure** we can see that low temperature is positively correlated with the usage of bikes, while in **Figure** we can see no correlation with high humidity. **Table** contains all the correlation coefficients between different attributes and bike usage.

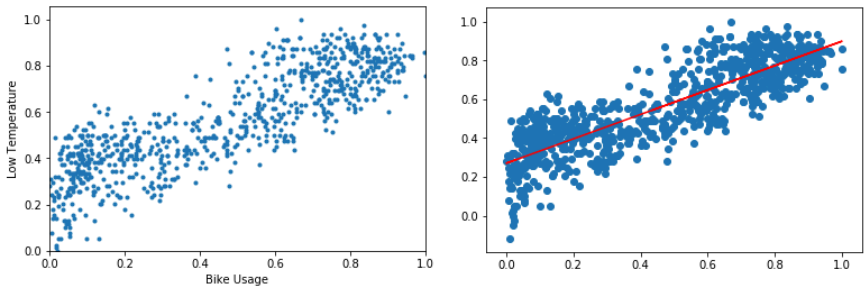


Figure 8: Correlation (0.7) between Low Temperature and Bike Usage

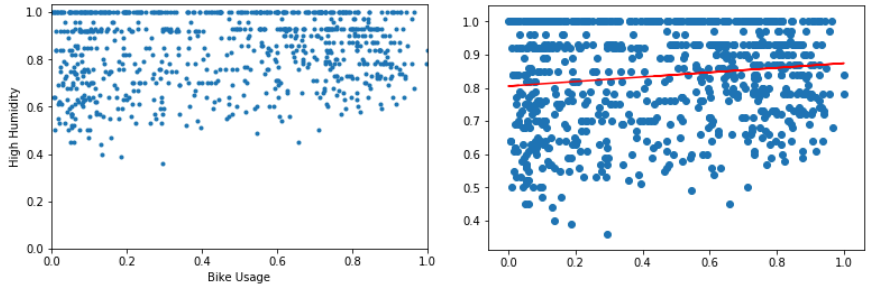


Figure 9: Correlation (0.02) between High Humidity and Bike Usage

Table 1: Correlation coefficient

| **Attribute** | $R^2$ |
|---|---|
| Low Temperature | 0.7278257733186797 |
| High Temperature | 0.6914113376022786 |
| Low Dew Point | 0.6208915092809677 |
| High Dew Point | 0.5772408684832167 |
| Low Humidity | 0.0014386835458154446 |
| High Humidity | 0.018873022035322595 |
| Low Visibility | 0.043489802661249355 |
| High Visibility | 0.000004993336488734 |
| Average Wind | 0.07990729177445965 |
| High Wind | 0.07481014953823728 |
| Wind Gust | 0.07667909130885997 |
| Snowfall | 0.05167595979579198 |
| Precipitation | 0.014230885353750944 |

## 5.3 ARIMA

In a nutshell, bike usage or bike flow data represents a time series, which is a sequence of scalars that depend on time t. The objective of prediction is to guess future values by observing the past ones. Auto-regressive integrated moving average is a generalization of an autoregressive moving average (ARMA), and it is composed of two distinct models which explains the behaviour of a series from two different perspectives: the autoregressive (AR) models and the moving average (MA) models. According to a number of sources **CITE** regarding univariate time series methods, when proposing new prediction methods, comparisons should be made against a naive and standard method such as an ARIMA model. This is to say that the models that should be considered as a novel one should outperform the ARIMA model by comparing the performance metrics.

First step in implementing ARIMA is to test stationarity with an augmented Dickey-Fuller (ADF) test **CITE** where we need to prove our reject our null-hypothesis:

- $H_0$ ... data is non-stationary
- $H_1$ ... data is stationary

A non-stationary time series show seasonal effects, trends, and other structures that depend on the time. Dickey-Fuller test in the case of data usage data produced a p-value of 0.371320. Because we got a p-value that is larger than 0.05, we proved the proposed null-hypothesis, which means our data has an unit root **CITE** and that the data can be used in ARIMA model once we verify rolling statistics.

Table 2: Augmented Dickey Fuller test

| **Test Statistic** | **p-value** | **Lags** | **Observations** |
|---|---|---|---|
| -1.818492 | 0.371320 | 20 | 800 |

Rolling statistics indicates that summary statistics like the mean and variance do change over time, providing a drift in the concepts a model may try to capture. **CITE** In **FIGURE** and **FIGURE** we can observe and confirm these assumptions.

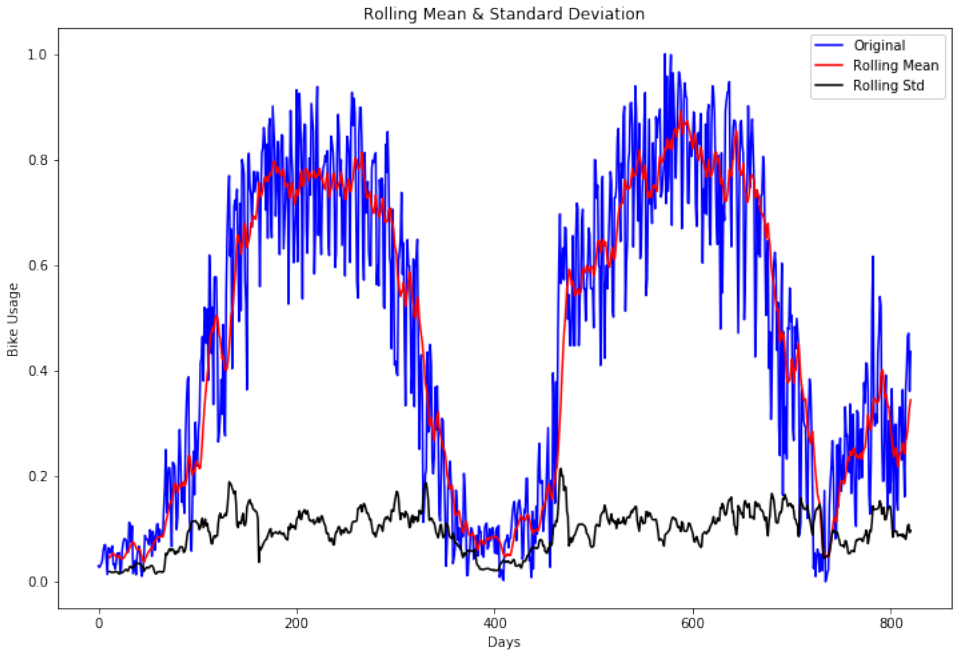


Figure 10

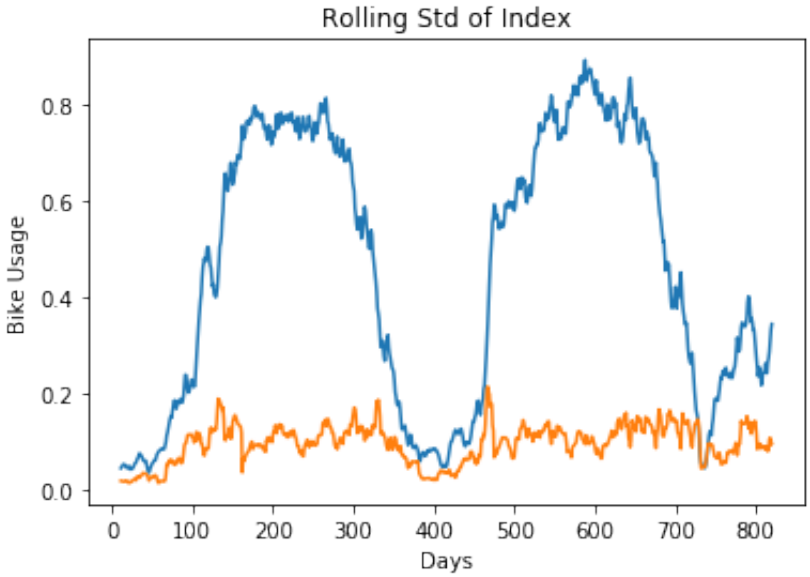


Figure 11

As a last step before building an ARIMA model, we are going to observe auto-correlation function and partial auto-correlation function which can be seen in **FIGURE**. Auto-correlation function (ACF) explains how well the present values of the series are related to its past values. We can sew that for 30 lags there is a strong correlation above the 0.7 value. Lags are defined as an observations with previous time steps and

the higher the lags, the further into the past we are trying to find correlation. Included in the **FIGURE** is the auto-correlation plot in case of the extreme case of choosing maximum number of lags (820). According to the literaure **CITE** this is exactly what we would expect: an ACF for the MA process to show a strong correlation with recent values up to the lag of k, then a sharp decline to low or no correlation.
Partial auto-correlation function (Partial-ACF) represents the correlation of residuals, hence being a summary of the relationship between an observation in a time series with observations at prior time steps with the relationships of intervening observations removed **CITE**. Again, as described in theory **CITE**, we would expect the plot to show a strong relationship to the first lag and then suddenly trailing off of correlation afterwards.

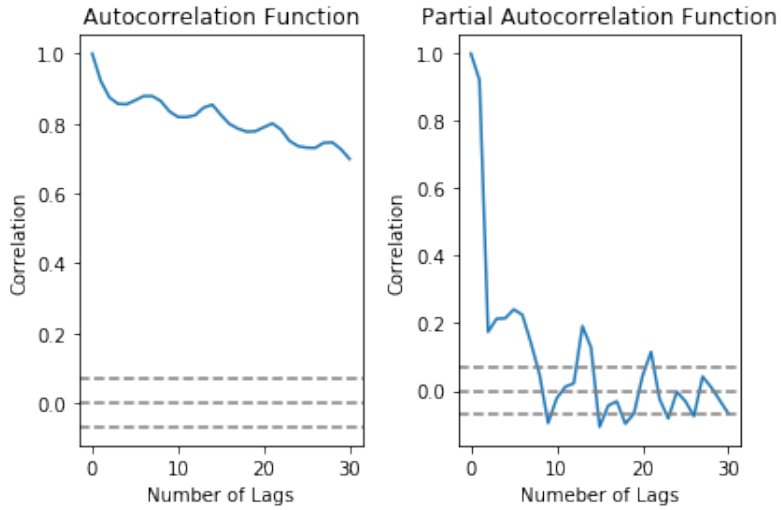


Figure 12

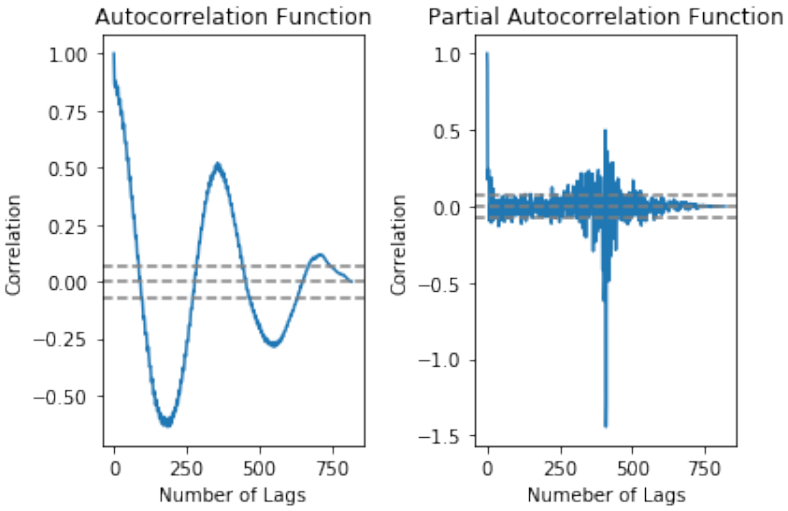


Figure 13

Finally, ARIMA model can be built based upon all the previous calculated necessities. **FIGURE** shows the predicted bike usage in red and also produces a number of error metrics. This metrics and performance results will be used as a benchmark for all the recurrent neural network predictions to be explored in this thesis.
The parameters of the ARIMA model (p,d,q) are defined as follows **CITE**:

- p: The number of lag observations included in the model, also called the lag order.
- d: The number of times that the raw observations are differenced, also called the degree of differencing.
- q: The size of the moving average window, also called the order of moving average.

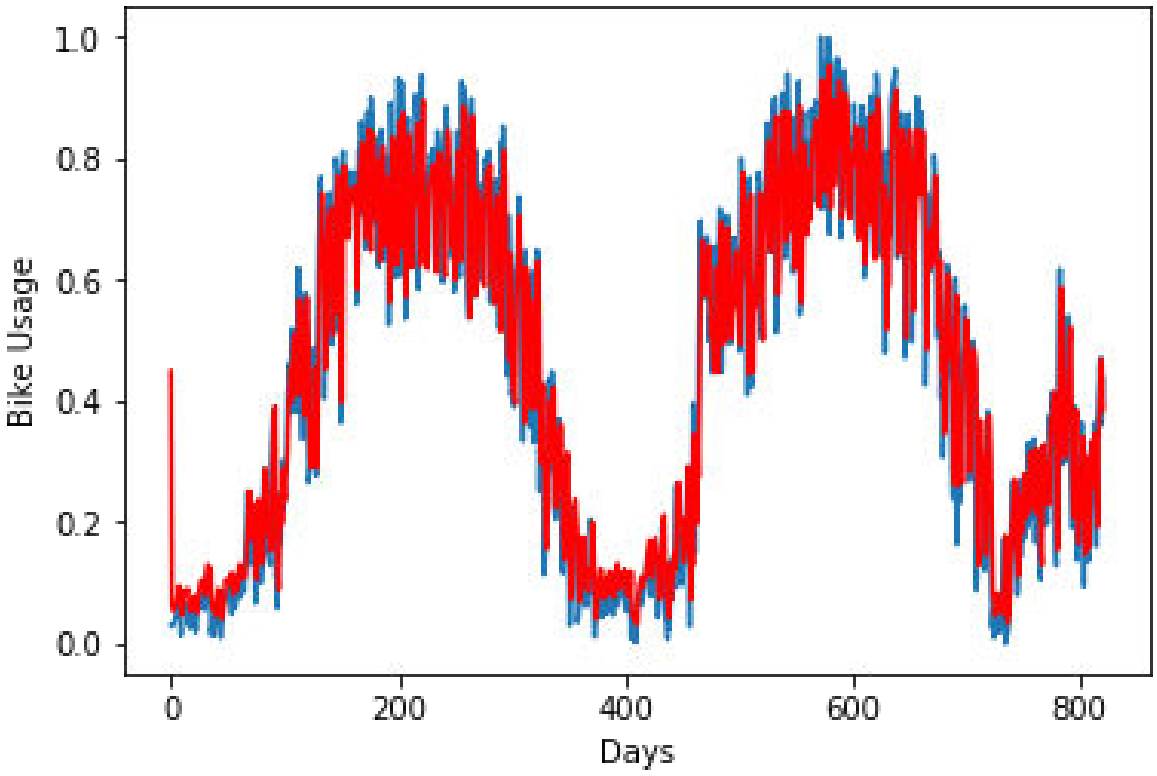


Figure 14: ARIMA Predicted Bike Flow (2,0,0)

Table 3: ARIMA (2,0,0) evaluation metrics

| **Metric** | **Score** |
|---|---|
| Elapsed time | 60 miliseconds |
| RSS scaled | 10.5776 |
| MAE scaled | 0.0856 |
| MAE | 632.1342 |
| MSE scaled | 0.0128 |
| MSE | 702543.2804 |
| RMSE scaled | 0.1135 |
| RMSE | 838.1785 |
| Accuracy | 0.8133 |

Table 4: ARIMA (20,0,0) evaluation metrics

| **Metric** | **Score** |
|---|---|
| Elapsed time | 161.28 seconds |
| RSS scaled | 7.6769 |
| MAE scaled | 0.07269 |
| MAE | 536.7394 |
| MSE scaled | 0.00935 |
| MSE | 510006.1863 |
| RMSE scaled | 0.09669 |
| RMSE | 714.1471 |
| Accuracy | 0.8415 |

## 5.4 Simple RNN

Recurrent neural network or feedback neural network expand on the major shortcomings of traditional neural networks. RNN are networks with loops, allowing information to persist and predicting the future by observing the past. In a sense RNN operates as a multiple feedforward neural networks. **CITE**. One big drawback of a simple RNN is that it has a vanishing gradient problem **CITE**.

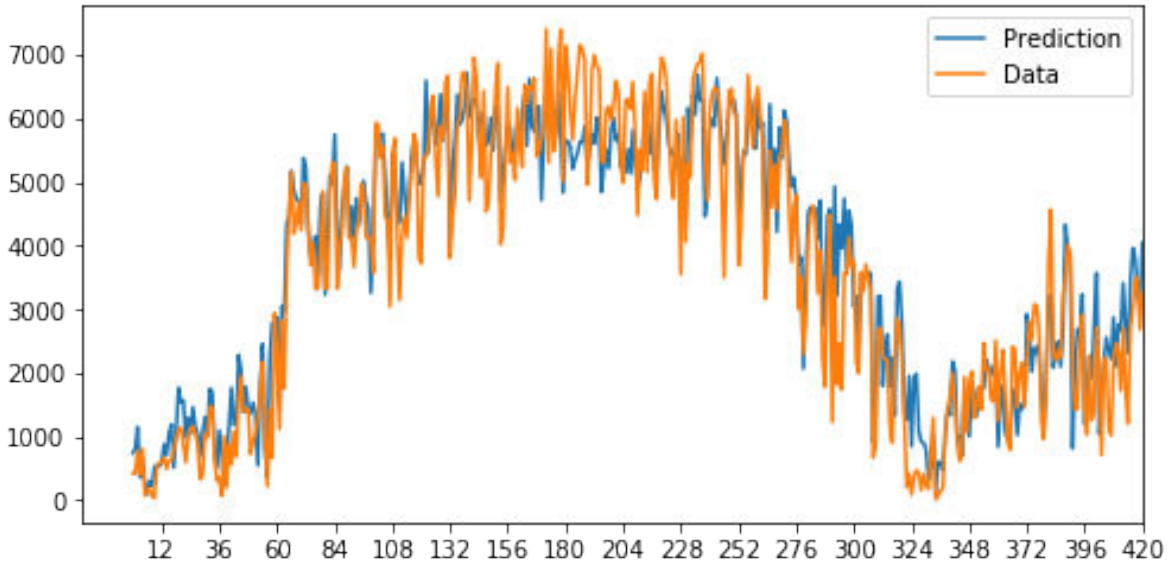


Figure 15: Predicted Bike Flow with simple RNN

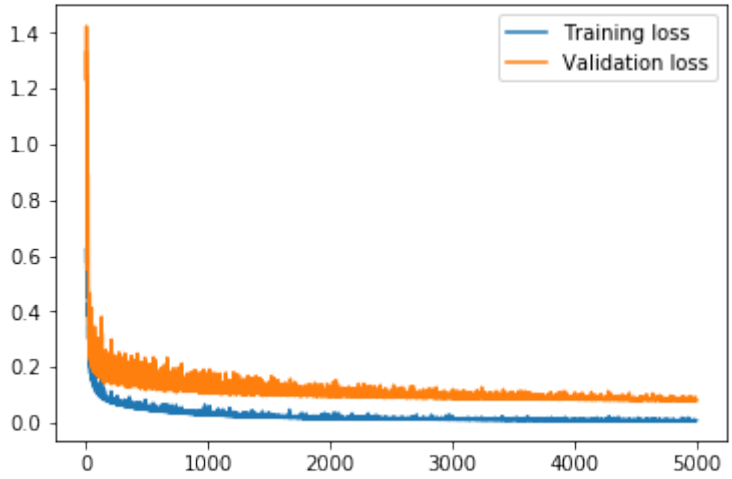


Figure 16: Simple RNN Loss Functions

Table 5: Simple RNN

| **Parameters & Scores** | **Iteration 1** | **Iteration 2** | **Iteration 3** | **Iteration 4** |
|---|---|---|---|---|
| Test Set | 321 | 321 | 321 | 321 |
| Epochs | 100 | 1'000 | 1'000 | 2'500 |
| Learning Rate | 0.001 | 0.001 | 0.001 | 0.001 |
| Hidden Nodes | 10 | 10 | 25 | 30 |
| Elapsed time | 0.4717 sec | 5.0156 sec | 5.5900 sec | 14.2091 sec |
| MAE | 1743.3761 | 1320.1722 | 1156.3751 | 783.8121 |
| MSE | 4269675.239 | 2612542.4243 | 1982988.6854 | 942435.7041 |
| RMSE | 2066.3192 | 1616.33611 | 1408.1863 | 970.7912 |
| Accuracy | 57.0039 | 67.4412 | 71.4808 | 80.6692 |

## 5.5 Deep RNN

In general, deep neural networks have multiple levels of hidden layers. However, deep neural networks are often much harder to train than shallow neural networks **CITE**. All of this is true for deep recurrent neural networks as well. Also, concept of depth in an RNN is not as clear as it is in feed-forward neural networks [12]. Here, a performance of deep RNN on the same bike sharing dataset will be analyzed in the same manner as it was done for the simple RNN.

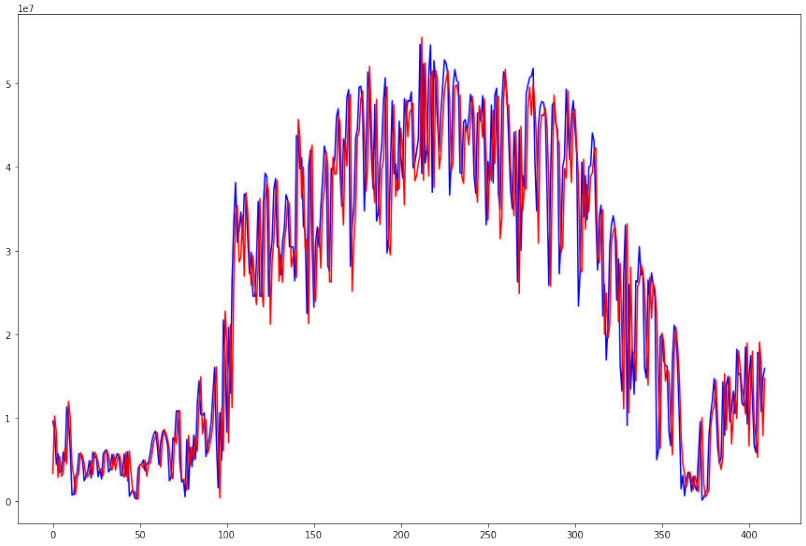


Figure 17: Deep RNN Predicted Bike Flow

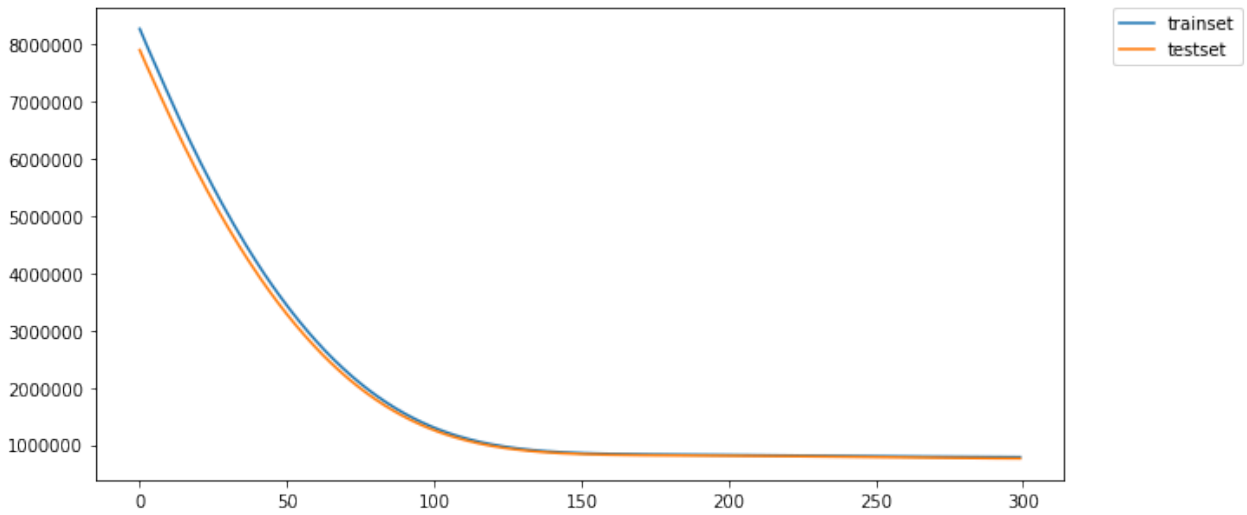


Figure 18: Deep RNN Loss Functions

Table 6: Deep RNN

| **Parameters & Scores** | **Iteration 1** | **Iteration 2** | **Iteration 3** | **Iteration 4** |
|---|---|---|---|---|
| Test Set | 321 | 321 | | |
| Epochs | 300 | 500 | | |
| Learning Rate | 0.001 | 0.001 | | |
| Activation functions | tanh, 2(relu) | 2(relu), tanh, relu | | |
| Hidden Nodes | 8+8+1 | 24+12+8+1 | | |
| Optimizer | adam | adam | | |
| Elapsed time | 160.87 sec | 259.6 sec | | |
| MAE | 696.4746 | | | |
| MSE | 626651.29 | | | |
| RMSE | 791.613 | | | |
| Accuracy | 83.8714 | 84.7167 | | |

## 5.6 RNN LSTM

Long Short-Term Memory is a specific type of recurrent neural network capable of learning long-term dependencies **CITE**. Not only does LSTM deal better with the vanishing gradient problem, but it is well suited for making predictions based on larger time series data **CITE**.

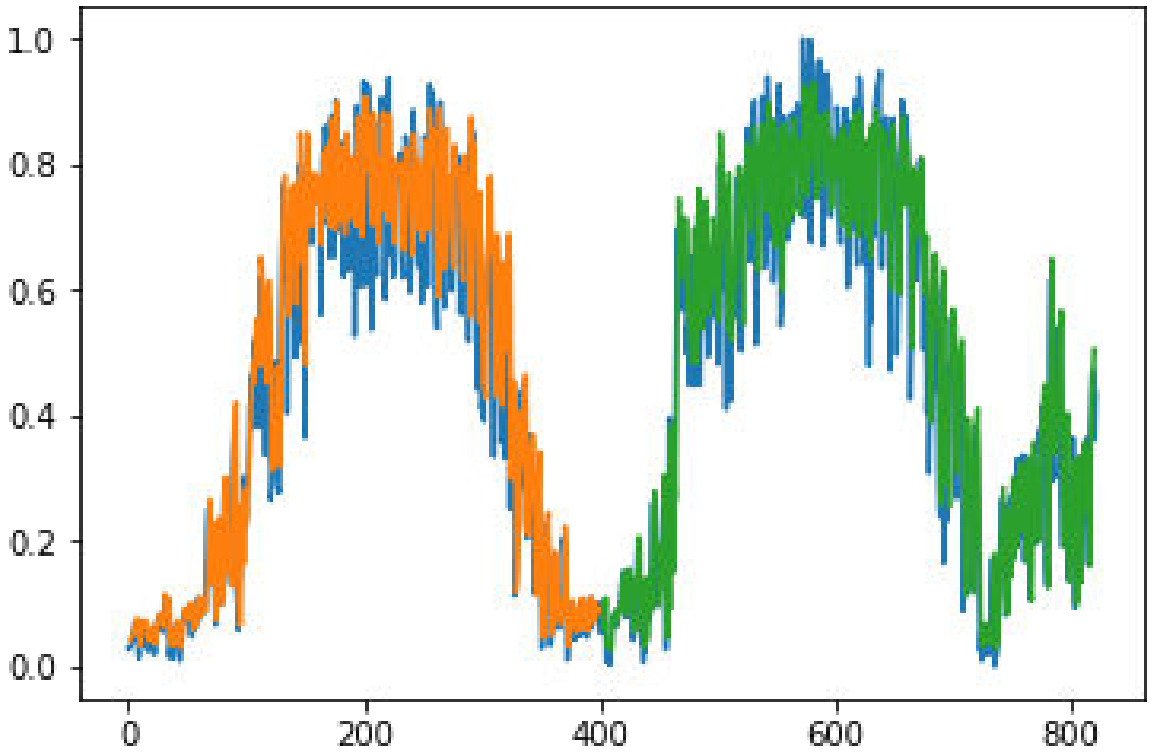


Figure 19: RNN LSTM Predicted Bike Flow

Table 7: RNN LSTM

| **Parameters & Scores** | **Iteration 1** | **Iteration 2** | **Iteration 3** | **Iteration 4** |
|---|---|---|---|---|
| Test Set | 321 | | | |
| Epochs | 300 | | | |
| Learning Rate | 0.001 | 0.001 | 0.001 | |
| Hidden Nodes | tanh,relu | | | |
| Elapsed time | | | | |
| MAE | 102.5375 | | | |
| MSE | 463712.94 | | | |
| RMSE | 680.9647 | | | |
| Accuracy | 0.88019 | | | |

As LSTM will be the method used for predicting dynamic patterns in this thesis due to its excellent performance, a cross validation of

## 5.7 RNN Validation Metrics

Instead of a widely used K-fold cross-validation metric, time series are specific and we would want to implement a series of test sets, each consisting of an equal number of

observations. The corresponding training set consists only of observations that occurred prior to the observation that forms the test set **CITE**. This is supposed to ensure that no future observations are used in constructing the forecast. Also, two different approaches are used together in order to get a fixed training set with moving test set across different training set spans. This is just to stronger ensure that the accuracies obtained previously are valid.
As shown in the **FIGURE**, graphs in the first row have a training set of size 500 and a sliding test set of size 107. From left to rihgt scores obtained by using parameters seen in Iteration 1 of **TABLE** are: 91.4%, 87.6% and 82%. Second row has a training set of size 607 and a sliding test of the same size as the sliding test set mentioned before. Score for the test sets of the two graphs in the second row are: 90.7% and 81% respectively. Now, it is easy to notice that as the sliding test window is moving further into the future, the prediction accuracy continues to drop down. But as long as we are trying to predict bike usage 4 months into the future, the LSTM guarantees to produce at least 90% prediction score.

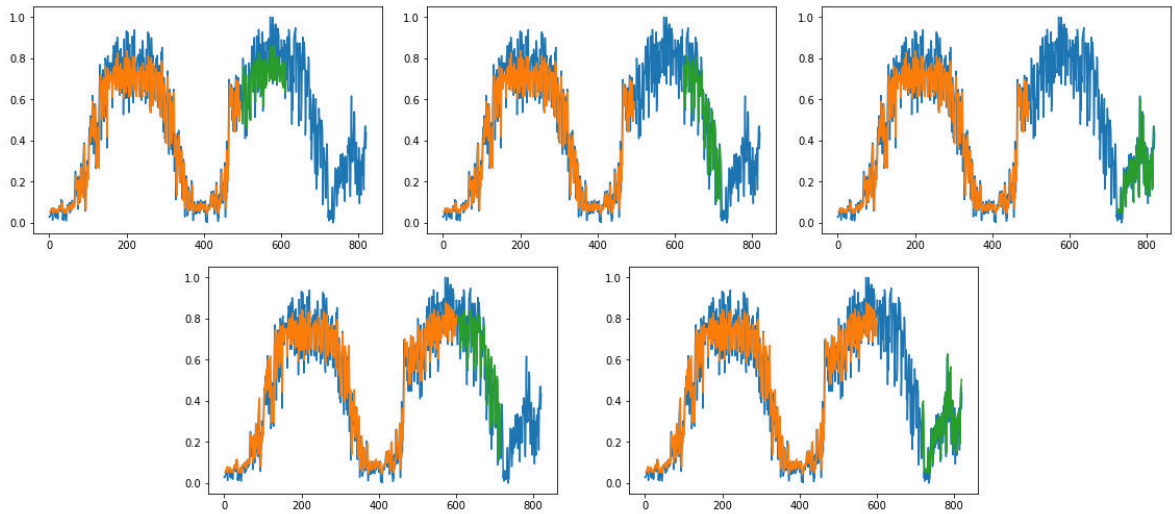

Figure 20: LSTM cross validation

# 6 Identifying Spatial Structures with CNN

In this section, pairwise most asymmetric nodes representing most unbalanced stations which had been previously retrieved in the Chapter 4 are transformed into adjacency matrices and prepared as an input for convolutional neural network. Before running the CNN model, training matrices need to be labeled and the ultimate goal of this method is to make sure that CNN correctly classifies test matrices to a correct label. In case of a false classification, we still want to maintain those errors to be spatially close to the expected stations thus making minimal mistakes.

CNN is a neural network, a regularized version of multilayered perceptrons **CITE** and mostly applied in image recognition and classification areas. Because bike flows and unbalanced stations in particular tend to form a specific graphs which changes over time, that was the motivation to try and figure out how could these observed spatial configuration be recognized and classified into an existing patter with the help of CNN.

## 6.1 Adjacency matrices

The first step is to take snapshots of the directed graph that is formed by the most unbalanced stations defined in Chapter 4. These snapshots have a certain time granularity, so we can observe different unbalanced graphs each month, week, day. Suppose that the granularity is monthly, which means that during one year worth of time we will have 12 unbalanced graphs. In the case discussed here, we will take years 2017 and 2018 into consideration amounting to a total of 24 distinct unbalanced graphs when considering monthly granularity.

Using the data exploration findings from chapter 4 we can observe this monthly snapshots of the top three most unbalanced pairs of stations:

Table 8: 2017 most unbalanced station pairs

| **Time** | **Gap1** | **Stations1** | **Gap2** | **Stations2** | **Gap3** | **Stations3** |
|---|---|---|---|---|---|---|
| Jan 2017 | 64 | Ames,Vassar | 46 | Broadway,Post | 41 | Central,Mass |
| Feb 2017 | 72 | Vassar,Stata | 60 | Vassar,Ames | 45 | Vassar,Mass |
| Mar 2017 | 68 | Vassar,Stata | 54 | Vassar,Ames | 53 | Stata,Cambridge |
| Apr 2017 | 121 | Vassar,Stata | 75 | Vassar,Pacific | 68 | Vassar,Ames |
| May 2017 | 187 | Stata,Vassar | 108 | Stata,Pacific | 98 | Nashua,South |
| Jun 2017 | 174 | Vassar,Stata | 118 | Stata,Pacific | 88 | Nashua,Rowes |
| Jul 2017 | 162 | Vassar,Stata | 113 | Davis,Teele | 110 | Mass,Boylston |
| Aug 2017 | 37 | Pacific,Stata | 33 | Rowes,South | 28 | Vassar,Stata |
| Sep 2017 | 75 | Pacific,Vassar | 31 | Beacon,Mass | 31 | Nashua,South |
| Oct 2017 | 65 | Vassar,Stata | 55 | Stata,Sidney | 47 | Stata,Pacific |
| Nov 2017 | 52 | Beacon,Mass | 38 | Davis,Teele | 34 | Mass,Vassar |
| Dec 2017 | 66 | Mass,Pacific | 52 | Mass,Stata | 44 | Stata,Inman |

Table 9: 2018 most unbalanced station pairs

| Time | Gap1 | Stations1 | Gap2 | Stations2 | Gap3 | Stations3 |
|---|---|---|---|---|---|---|
| Jan 2018 | 51 | Nashua,Stata | 49 | Stata,Vassar | 46 | Mass,Pacific |
| Feb 2018 | 77 | Central,Mass | 62 | Nashua,Stata | 56 | Mass,Pacific |
| Mar 2018 | 80 | Nashua,Stata | 71 | Central,Stata | 69 | Central,Mass |
| Apr 2018 | 96 | Mass,Central | 94 | Mass,Beacon | 72 | Rowes,Cross |
| May 2018 | 148 | Stata,Vassar | 99 | Rowes,Cross | 96 | Stata,Pacific |
| Jun 2018 | 161 | Stata,Vassar | 115 | Rowes,Cross | 115 | Stata,Pacific |
| Jul 2018 | 172 | Stata,Pacific | 145 | Stata,Vassar | 132 | Rowes,Cross |
| Aug 2018 | 198 | Stata,Vassar | 195 | Stata,Pacific | 140 | Central,Mass |
| Sep 2018 | 164 | Stata,Central | 109 | Central,Pacific | 97 | Stata,Mass |
| Oct 2018 | 151 | Central,Stata | 129 | Mass,Vassar | 123 | Central,Mass |
| Nov 2018 | 177 | Stata,Vassar | 130 | Mas,Vassar | 103 | Mass,Central |
| Dec 2018 | 125 | Stata,Vassar | 108 | Mass,Central | 106 | Stata,Pacific |

Now, it is necessary to convert this stations into some form of identification numbers. If we aggregate both years together and find the exact number of total distinct stations, it becomes clear that there are a total of 20 such identification numbers we would need to allocate to each one of the stations. Before the best ID placement strategy can be discussed, in the **IMAGE** we can observe most unbalanced stations and network edges for year 2017 and in **IMAGE** we can see the same for year 2018.

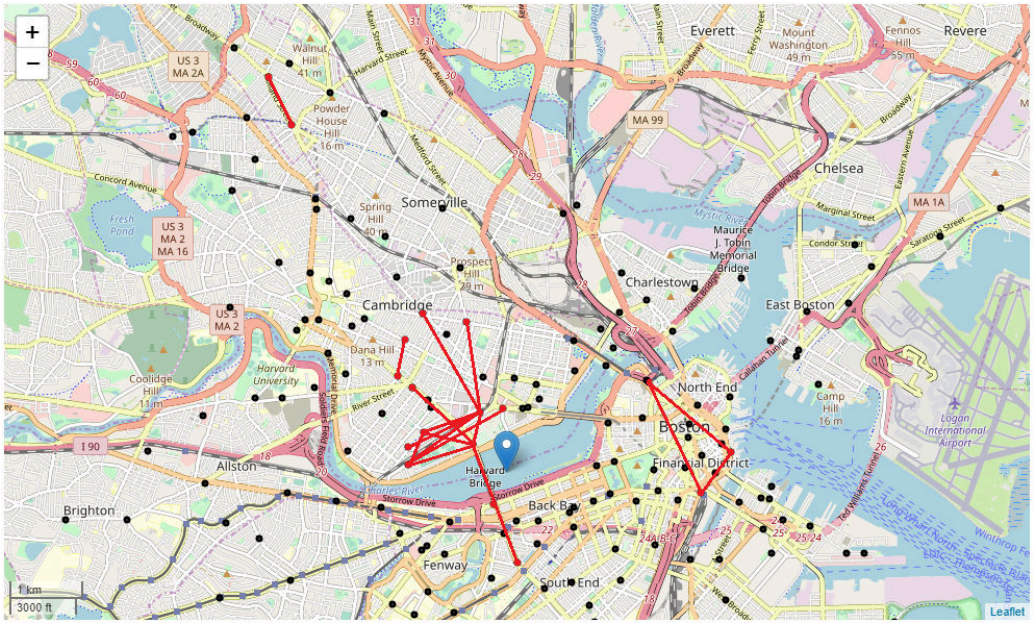


Figure 21: Aggregated unbalanced edges for year 2017

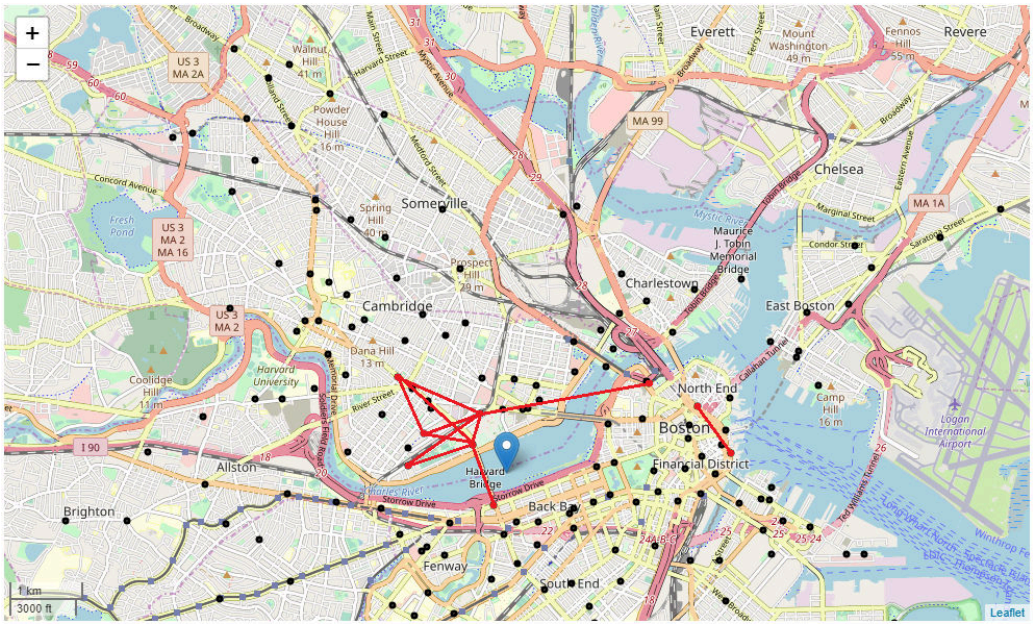


Figure 22: Aggregated unbalanced edges for year 2018

Allocation of the ID numbers is done in the manner that mimics the way these stations are spatially placed within the area of Somerville, Cambridge and Boston. Fore example stations Teele and Davis in Somerville got ID 1 and 2 because they are located in the far north-west part of the map. As we get closer to the Cambridge city center, other stations are given their unique ID. Last couple of numbers are assigned to the stations in Boston area. Now, **TABLE** and **TABLE** are translated into **TABLE** where each station is represented as its distinct ID number as this approach will make it possible to utilize the next step where each snapshot will be represented in a form of a square symmetric matrix.

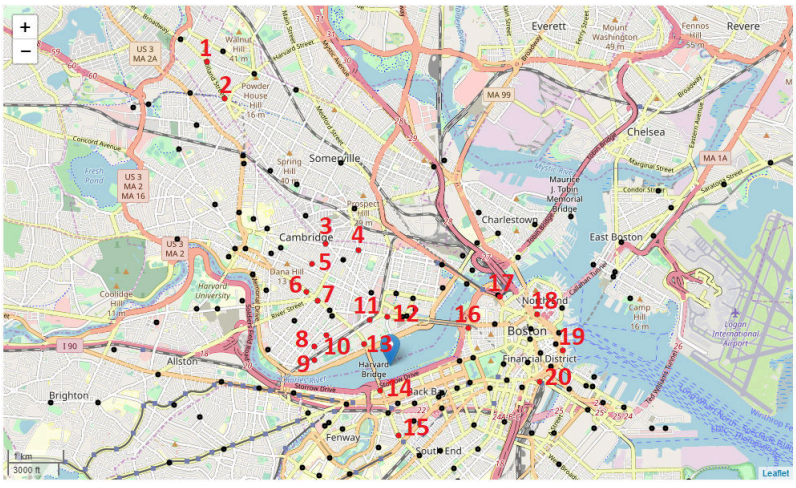


Figure 23: Appearances of all unbalanced stations during both 2017 and 2018

Table 10: 2017 most unbalanced station pairs

| **Time** | **Top1** | **Top2** | **Top3** |
|---|---|---|---|
| Jan 2017 | (9,12) | (5,6) | (7,13) |
| Feb 2017 | (9,11) | (9,12) | (9,13) |
| Mar 2017 | (9,11) | (9,12) | (11,4) |
| Apr 2017 | (9,11) | (9,10) | (9,12) |
| May 2017 | (11,9) | (11,10) | (11,12) |
| Jun 2017 | (9,11) | (11,10) | (11,12) |
| Jul 2017 | (9,11) | (1,2) | (13,15) |
| Aug 2017 | (10,11) | (19,20) | (9,11) |
| Sep 2017 | (9,10) | (13,14) | (17,20) |
| Oct 2017 | (9,11) | (11,8) | (11,10) |
| Nov 2017 | (13,14) | (1,2) | (13,9) |
| Dec 2017 | (13,10) | (13,11) | (11,3) |
| Jan 2018 | (17,11) | (11,9) | (13,10) |
| Feb 2018 | (7,13) | (17,11) | (13,10) |
| Mar 2018 | (17,11) | (7,11) | (7,13) |
| Apr 2018 | (13,7) | (13,14) | (18,19) |
| May 2018 | (11,9) | (18,19) | (11,10) |
| Jun 2018 | (11,9) | (18,19) | (11,10) |
| Jul 2018 | (11,10) | (11,9) | (18,19) |
| Aug 2018 | (11,9) | (11,10) | (7,13) |
| Sep 2018 | (11,7) | (7,10) | (11,13) |
| Oct 2018 | (7,11) | (13,9) | (7,13) |
| Nov 2018 | (11,9) | (13,9) | (13,7) |
| Dec 2018 | (11,9) | (13,7) | (11,10) |

In order to be able to use CNN, it is necessary to transform these unbalanced graphs into 24 adjacency matrices consisting of zeroes and ones, where "1" indicates that there is a directed edge from the station represented as row index "i" to the station denoted as column index "j" for that specific tuple. Of course, because we are dealing with the bi-directional unbalanced graphs, adjacency matrices will be symmetric. Now, these sparse matrices are simply pixelated images where "1" indicates pixel and it is something that can be send as an input to CNN. However, there is a crucial step in transforming graphs into matrices and, as explained before, that is to be aware that relative coordinates between the stations should match the allocation of row and column indexes inside the matrix. In other words, created matrices mimic the spatial representation of unbalanced stations in the real world.

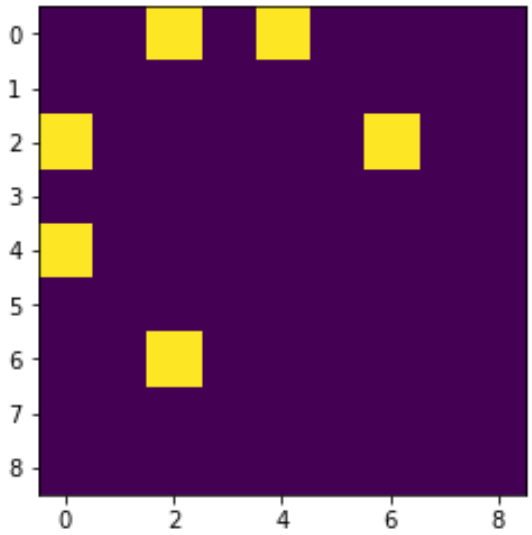


Figure 24: Adjacency matrix for March 2018 with isolated stations for that year

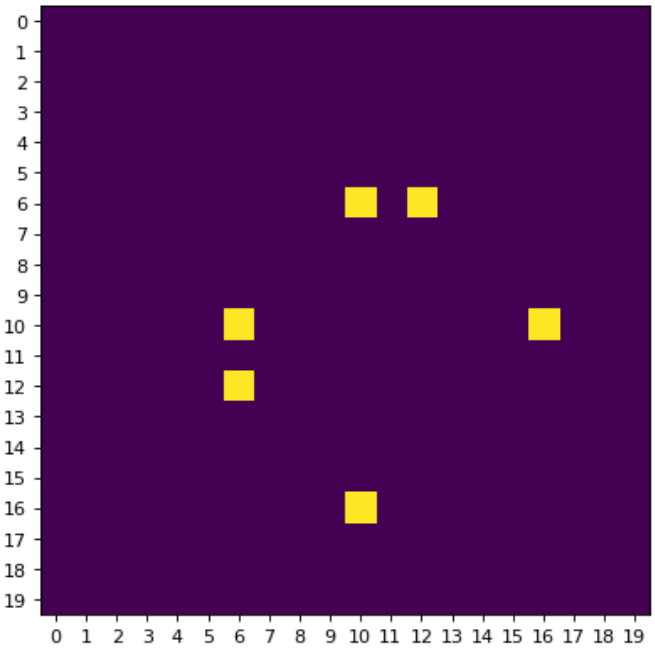


Figure 25: Adjacency matrix for March 2018 combined with stations from both years

For example, in **IMAGE** there is a matrix representing a particular month with the most unbalanced flows being between stations (0,2), (0,4) and (2,6). Stations 0,1,2,3,4 and 5 are located in the center of Cambridge, station 6 is Boston Bay area, 7 and 8 are in central Boston. This means that depending on the pattern of pixels in the matrix there are different flows connecting distinct neighbourhoods. Since CNN is a model that

learns to recognize these patterns and find them in pixelated images that have not been observed before, that is the reason why the method had been chosen for this purpose.

## 6.2 Motivation

This particular approach described was chosen because for the number of reasons. To start off, unbalanced bike flows between stations are a real problem and especially become more prominent as the bike sharing program grows bigger within the area over time **CITE**. However, it is not easy to predict exactly which unbalanced pairs are to be expected for future days but can be predicted by using what we currently observe as a test data for CNN. There are some trends that can be noticed like having one small graph in the winter and two sub-graph: a giant component in Cambridge and a smaller component either in Boston or Somerville. Weighted diameter of the giant component is smaller compared to the components outside Cambridge. Moreover, not only are the unbalanced pairs of nodes also a good approximation of the most used bike station in general (as described in the Chapter 4) but looking at unbalanced pairs can detect a less used bike stations that suddenly during a short period of time get congested with bike traffic and build their unbalanced ratio to a critical point. Also, CNN is needed as a step before RNN simply because it is not feasible to try and predict the most unbalanced pairs with that RNN method alone. It is not because it wouldn't be possible but because graphs, although seemingly simple, can require a vast computing power in order to predict bike flow dynamics for every single pair of nodes **CITE**. Having a number "n" of nodes, the total number of possible edges is $e = \frac{n*(n-1)}{2}$. Assuming 10 seconds which, on average, takes RNN to predict time-series, it would take 10*e seconds. Number of stations being 194 in 2018 means that it would take 187'210 seconds (52 hours). After that, we would still need to calculate flow differences and order them in the descending order. As we would like to make a short-term prediction for the following next few days, the method where we would use exclusively RNN is absolutely not appropriate from the computational complexity point of view.

## 6.3 Convolutional neural network

Training data that is being forwarded to CNN consists of 24 pixelated images of size 20x20. Each image represents the top 3 most unbalanced pairs of stations for that month. Yellow pixels, as seen in **FIGURE** is an indication of an edge between the station i and j, where i and j are identification numbers of those two stations having an unbalanced link. Training data will be used for CNN to observe and learn from all the past configurations and try to guess which of the pre-existing shapes would match the best any of the test data-sets we will additionally provide. It is important to notice that a small number of months in the training set have duplicates because some months have had the same top 3 unbalanced station pairs. This duplicated are a valuable indication for CNN that these configuration may have a greater importance to repeat again in the future. Each of the training images is manually labelled with a certain number which

corresponds to its unique spatial configuration. Of course, duplicates will be assigned an identical label.

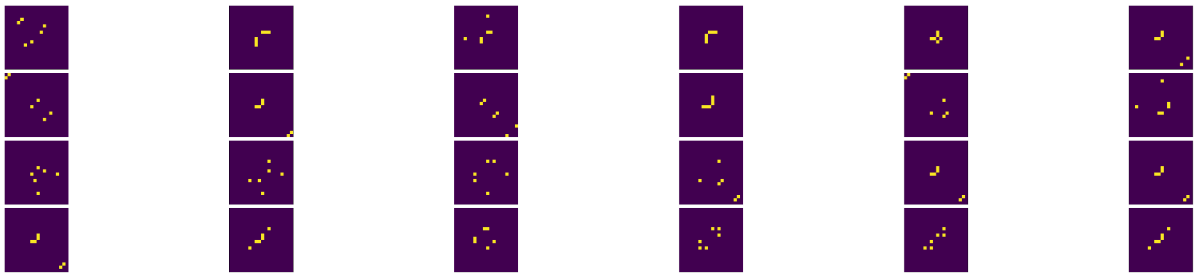

Figure 26: Training data

For the test set, as we want to test how CNN performs, a combination of various spatial configurations will be used. Just as a base case, every label from the training set will be used in the test set as well, because we want those labels to be predicted with a 100% accuracy as they are known. In addition, new shapes which have never been observed before will be put into the test data-set. as seen in **FIGURE** first four rows consist of permuted training images. Last two rows have a combination of images with only two pairs of stations that resemble an existing pattern, two pairs of stations that are found in a new relationship, and image with more than 3 pairs of stations that both have older edges but also the new ones.

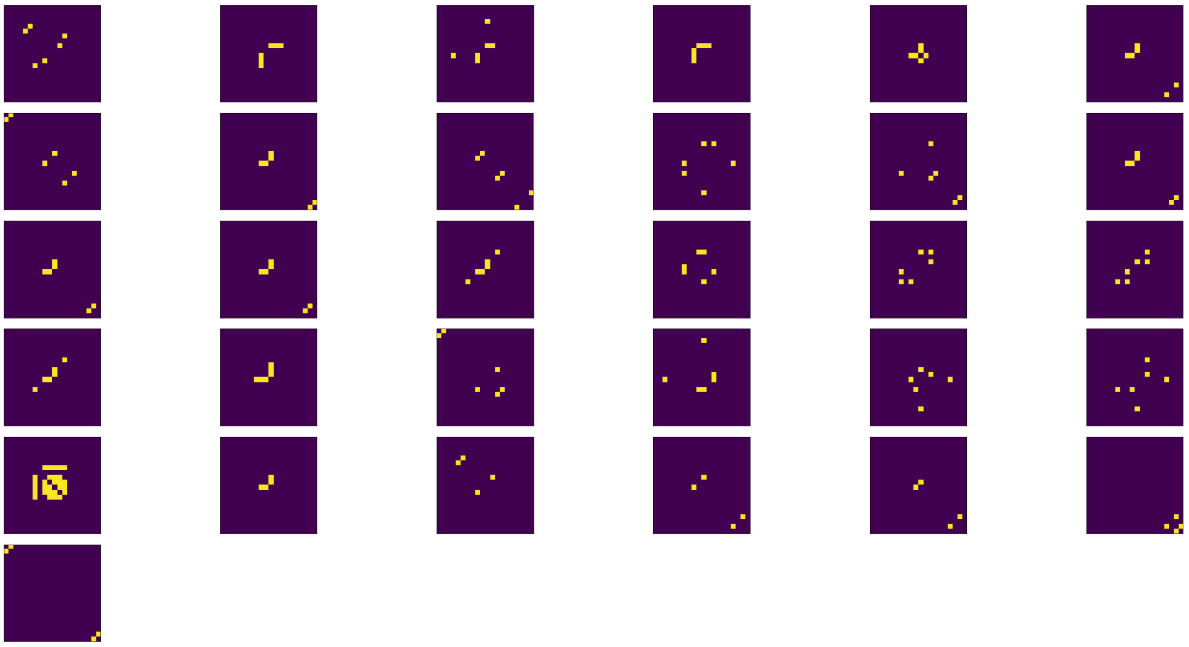

Figure 27: test data

Before running CNN, we define a ground truth with labels for the new shapes in the test set that we would probably want to see predicted correctly. Some of the new shapes

resemble a multiple older shapes, that is why once we see how CNN performs, false predictions will be inspected closely to see if there are somewhat precise even though our ground truth label was not matched. Also, the training data-set for the monthly granularity is not sufficiently big enough for producing high enough accuracy but that can be fixed by expanding it simply by duplicating 24 images 5 times resulting in a 120 images total. It is not advised to duplicate more than that as there could be a danger of overfitting **CITE**. CNN used for this purpose had 3 convolutional layers, 2 drop-out layers and 2 fully connected layers. Each convolutional layers had 12, 24 and 48 hidden layers within. Optimizer used was adam with the learning rate 0.001, activation function utilized was rectified linear unit, and the number of epoch was set to 100.
As a result, precision produced was precision is 93.5483 % by predicting correct label for 29 out of 31 images. Counting only new shapes, precision 0.714 % or 5 out of 7. It is important to highlight that false predicted labels were for those shapes which had completely new edges and extreme combinations. And they are false only because they did not match tot he ground truth which was approximated manually. In a sense, predicted labels were still good.

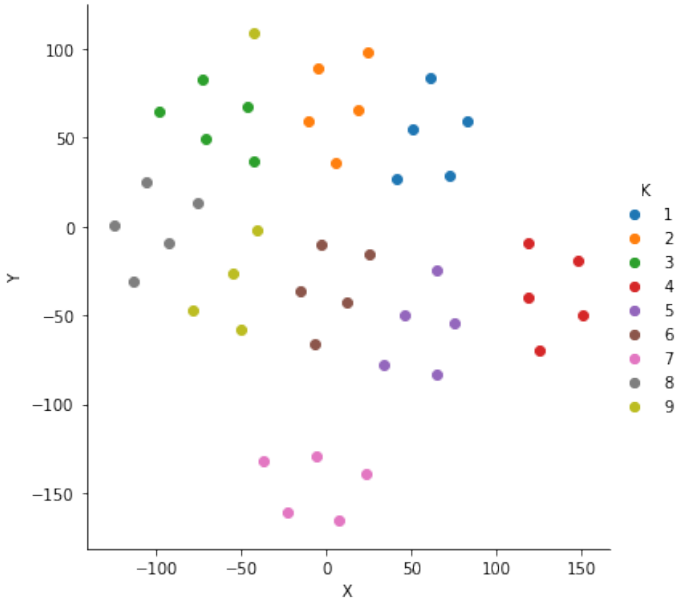


Figure 28: Stochastic Neighbor Embedding for isolated 2018 training set

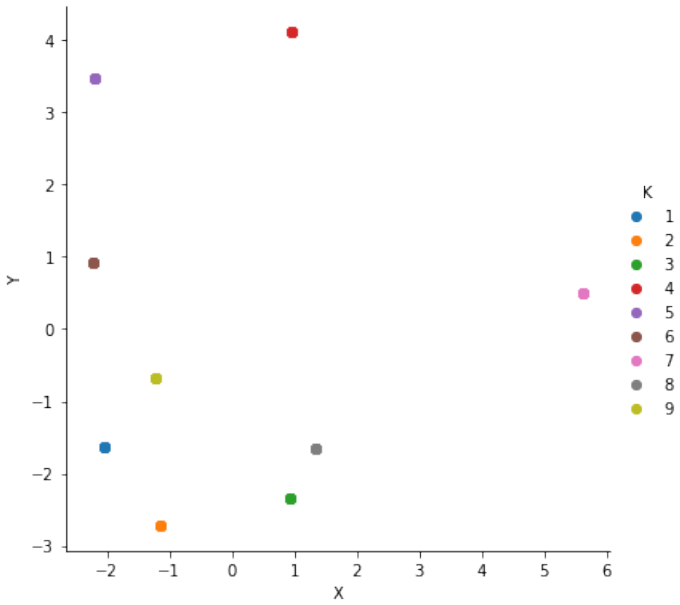


Figure 29: Principal Component Analysis for isolated 2018 training set

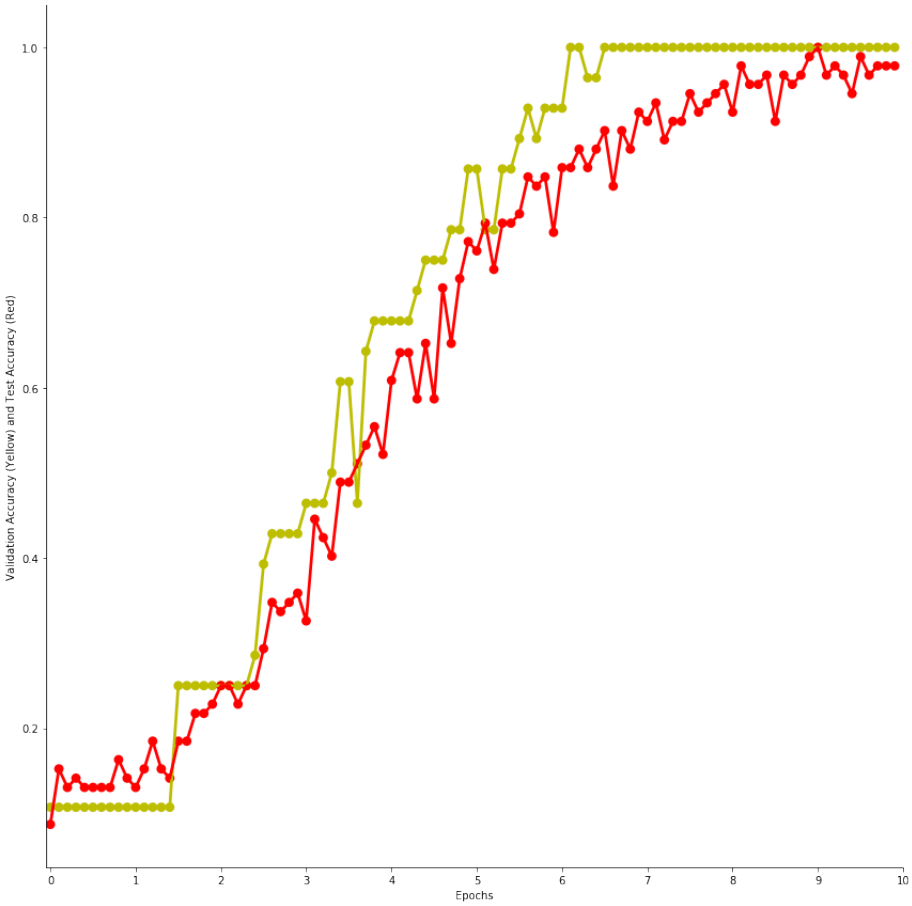
Validation Accuracy (Yellow) and Test Accuracy (Red)
1.0
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
9
10
Epochs

Figure 30

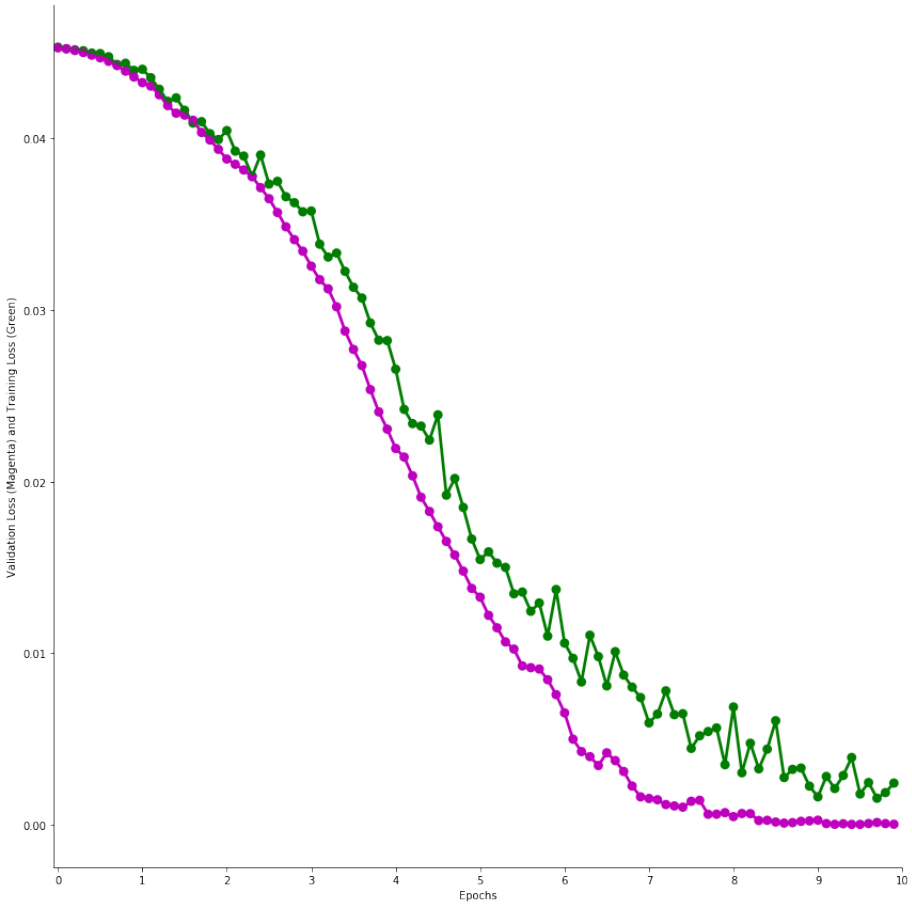


Figure 31

# 7 Discussion

//TO-DO
In this part the CNN and RNN described in the chapters before will be applied again using months of January, February, March and April of the year 2019 as a test set.

First, January will be taken and split into two sets. Larger (test) set will consist of around first 20 days of the month and for those first 20 days, a top 3 most unbalanced pairs of stations will be produced and transformed into adjacency matrix. In case we encounter a new station never observed before, that pair is ignored. But, of course in an ideal case, new station should be added and matrices expanded as those stations could start appearing in a top3 list in the future months and years. However, that is out of scope for the datasets of 2 year span used for this thesis and will not be performed.

CNN will try and classify the matrix of first 20 days in January to the one of the existing labels from the previous 2 years. This predicted label will be used as a mobility flow

model for the last 10 days of the month of January as we assume that together with the last 10 days, months usually tend to converge to the predicted labels.

Next, for the stations that are found in the predicted matrix, RNN method us utilized as we want to find the flow dynamic for each. To get the most unbalanced scores, we subtract the predicted flows from the unbalanced pairs.

To validate results, unbalanced scores and spatial configuration are copared to the ground truth of the January and last 10 days of Januray.

Same thing is repeated for other months.

### 7.1 Results

metrics, performances, validations...
//TO-DO

# 8 Conclusion

### 8.1 Future Work

//TO-DO

# References


[1] L. L. M. H. Schimmelpennink, "The Birth of Bike Share." October 1, 2012.

[2] G. McKenzie, "Docked vs. Dockless Bike-sharing: Contrasting Spatiotemporal Patterns," *10th International Conference on Geographic Information Science*, 2018.

[3] D. Freund, S. G. Henderson, E. O'Mahony, and D. B. Shmoys, "Analytics and Bikes: Riding Tandem with Motivate to Improve Mobility," *Interfaces*, 2019.

[4] R. Beecham, J. Wood, and A. Bowerman, "Studying commuting behaviours using collaborative visual analytics," *Computers, Environment and Urban Systems Volume 47, Pages 5-15*, September 2014.

[5] D. O'Sullivan and D. Unwin, *Geographic Information Analysis*, 2002.

[6] A. Sarkar, N. Lathia, and C. Mascolo, "Comparing Cities Cycling Patterns Using Online Shared Bicycle Maps," *Transportation, Volume 42, Issue 4, pp 541559*, April 2015.

[7] J. Froehlich, J. Neumann, and N. Oliver, "Sensing and Predicting the Pulse of the City through Shared Bicycling," *Proceedings of the 21st international jont conference on Artifical intelligence*, 2009.

[8] O. OBrien, J. Cheshire, and M. Batty, "Mining bicycle sharing data for generating insights into sustainable transport systems," *Journal of Transport Geography 34*, 2014.

[9] M. Z. Austwick, O. OBrien, E. Strano, and M. Viana, "The structure of spatial networks and communities in bicycle sharing systems," *PLoS ONE*, 2013.

[10] X. Zhou, "Understanding Spatiotemporal Patterns of Biking Behavior by Analyzing Massive Bike Sharing Data in Chicago," *PLoS ONE*, October 7, 2015.

[11] B. Ratner, "The Correlation Coefficient, journal =."

[12] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to Construct Deep Recurrent Neural Networks, journal =."