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 10, 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 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

Acknowledgments

I would like to express a great appreciation to my thesis supervisor Amir H. Payberah for his patient and unconditional support I received throughout the thesis. A special thanks to MIT SCL for providing the internship opportunity and many valuable remarks on my project, mostly by my external supervisors Carlo Ratti, Fábio Duarte, and Paolo Santi but also all the amazing coworkers I was privileged to work alongside with.

Contents

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
CNTK = Microsoft Cognitive Toolkit
CV = Cross Validation
D.C. = District of Columbia
DDGF = Data-Driven Graph Filter
DTW = Dynamic Time Warping
EDA = Exploratory Data Analysis
ELU = Exponential Linear Unit
FNN = Feedforward Neural Network
GPA = Generalized Procrustes Analysis
GCNN = Graph Convolutional Neural Network
GRU = Gated Recurrent Units
GUI = Graphical User Interface
HCA = Hierarchical Cluster Analysis
HITS = Hyperlink-Induced Topic Search
IP = Integer Programming
LCHS = London Cycle Hire Scheme
LDA = Latent Dirichlet Allocation
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
NN = Neural Network
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 Function
VGP = Vanishing Gradient Problem

List of Figures

List of Tables

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[1].

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 police, however, impounded the bikes, claiming that unlocked bikes incited theft [2].

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 [3] [4]. Data from the USA Department of Transportation's 2017. National Household Travel Survey[1] 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

[1] https://nhts.ornl.gov/vehicle-trips

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[5] [6]. 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[7]. In addition, it also brings several personal benefits for both mental and physical health[8]. 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 cardio-respiratory intensity is associated with health benefits[9]. However, both pedestrians and cyclists are more exposed to accidents and injuries compared to a car or transit passengers[10]. 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.[11] 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. [12] However, this method does not take the full-fledged prediction into an account. More specifically, every day in the week is observed as property-wise indistinguishable 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.

1.1.1 Knowledge Gap

Currently, there are not many state of the art methods that use graph mining and machine learning in the are of optimizing bike sharing networks.

The one that are published recently (between 2017 and 2019) implement Recurrent Neural Network to predict station level demand[13] of New York's Citi Bike dataset with an RMSE calculated 2.7069 on average[14]. However, the global overview of the network is not considered, the method have not been used to medium-sized bike sharing networks but exclusively on the New York's dataset that had been already optimized using a specially tailored operations research algorithm devised by Cornell University [12].

Only one paper [15] that addressed CNN used GCNN-DDGF model for station-level hourly demand prediction in a large-scale bike-sharing network. Although this paper proposes quite a highly efficient method, it does not tackle a consideration to use the method as a component in a comprehensive framework for dynamic bike rebalancing. Also, the model is not applicable to a directed graphs and it cannot learn a sparse graph filter.

Boston area BSS is a medium-sized BSS with a different amount of data and topology that may incur a particular dynamics than those investigated by the other papers (heavily New York dataset), which might favor other methods than those who have performed well in related research. Also, we would like to have computationally low-demanding method that takes little time to output the results and use more data than any other paper as no study used more than one year worth of data.

One paper[16] that addressed using conv-LSTM was focusing only on dockless bikes,

weather is not taken into consideration and rebalancing was not discussed.

In conclusion, knowledge gap to be explored in this thesis include a multitude of improvement combined based on the research papers written during the last three years in the domain of bike sharing:

- focuse on middle-sized docked bike sharing system not before explored in detail
- using secondary dataset such as weather information in addition to the primary bike dataset
- combine mobility flows for both spatial and temporal patterns
- the method does not only present the predictions but also suggests how the bike rebalancing strategy should be utilized for the predicted future state
- rebalancing strategy is scalable from the whole network system to the municipality or neighbourhood level
- computational complexity is low and the utilized method does not require hours to calculate results
- accuracy is still high and comparable to the other approaches
- not only are the research results presented and discussed but they also provide an application for the Boston bike sharing company with an approximation of the number of bike truck and areas that will require a rebalancing process

1.1.2 Research Question

The improvements and additions on top of the shortcomings of proposed state of the art models described previously will provide answers to the research question of this thesis:

Is it possible to predict both spatial configuration and flow dynamics of the pairwise most unbalanced pairs of stations as an approximation to the discrepancy of the supply-demand network balance?

The research question will be answered and evaluated in Chapter 7.

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:

1.4.1 Hypothesis 1

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.

1.4.2 Hypothesis 2

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

1.4.3 Hypothesis 3

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.

1.4.4 Hypothesis 4

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

1.4.5 Main Hypothesis

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. Expected accuracy for the predicted dynamic flows should be around 90% and predicted spatial patterns around 70% when compared to the ground truth. Spatial pattern matching using a trained 2D CNN will perform reasonably well for visual item similarity, but fine-tuning might be needed and will be the possible bottleneck for months with low bike flows such as winter months.

1.5 Ethical Considerations

Due to the increasing pervasiveness of machine learning, it is crucial that there is a discussion about the safety, transparency and bias of machine learning systems. However, in the bike sharing systems the focus is mainly on re-balancing and although information such as bike id, gender and user type are available - it is not used to identify individuals. The bike sharing company, however, may be storing personal information from the user the moment he registers for the service but the data is never publicly disclosed. From the company's perspective, in case they are using a more attribute-wise detailed data, they should ensuring that a machine learning model does not leak its training data to an adversary that might able to intercept a large number of queries the attacker can see how the statistics of the output distribution and cross-reference it to recover the

original data. In addition to the data driven ethical considerations, it is possible to infer from the data whether a bike had been stolen and for that case, a company should have regulations to investigate what is the maximum time range before the matter should be investigated.

1.6 Sustainability

As argued in the introduction, using bike sharing services reduces carbon footprint and promotes healthy lifestyle. Optimization of environment friendly transportation will effectively attract new customers and users. Ultimately, goals of this thesis work towards the overarching goal of creating a more sustainable and smarter cities, and many of related positive effects in the are of bike sharing networks is in accordance with "The 2030 Agenda for Sustainable Development", adopted by all United Nations[2].

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.

[2]https://sustainabledevelopment.un.org/

1.8 Thesis Outline

The following thesis report is organized as follows:
This first chapter introduced a short overview of bike sharing systems in general and its current development in the realm of flow predictions and re-balancing strategies using graph mining and machine learning. Knowledge gap had been defines and research question stated that we will attempt and answer in the following chapters.

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 chapter starts of by presenting a case study of Boston area bike sharing network and using data exploration and visualization techniques to summarize the most important properties and spatio-temporal changes to the network. Moreover, most unbalanced pairs of links between stations will be defined as a means to approximate an important metric to be used in the machine learning chapters to simplify the network only to those stations that are good candidates for re-balancing strategy.

Fourth chapter shortly summarize system architecture and machine learning pipeline used. A list of technologies, libraries and softwares will be provided.

Fifth chapter makes an overview of all the candidate methods for predicting dynamic patterns. A chosen method will be used on the identified stations in the next chapter.

Sixth chapter describes how to identify spatial structures and uses the unbalanced networks defined in Chapter 3. This is done by creating adjacency matrices of the past network configurations. Afterwards, adjacency matrices are labeled and used as an input for training the convolutional neural network which decides how would the newly observed spatial structures be labeled.

Seventh chapter discusses how to apply the previously introduces method in case of Boston bike sharing system and describing the results. Here, our research question will be answered.

Eighth chapter closes the thesis with a conclusion and recommendations for future work based on the matter presented in this thesis.

2 Related Work

2.1 Spatiotemporal Patterns

In the paper written by Grant McKenzie (2018)[17], 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)[12] 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)[11] 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[18].

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)[19] 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)[20] 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)[21] and Austwick et al.(2013)[22] 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 Sakoe-Chiba band (1978).

2.6 Mobility Prediction using Random Forest

Yang et al. (2016)[23] 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 by Pan et al. (2019)[24] 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 Chen et al. (2017)[13], 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. Furthermore, data exploration is performed, correlation made between weather and number of rentals, and a baseline approaches defined. Baseline approaches include: Ordinary least-squares regression (OLS), random forest (RF) with 50 estimators, and feedforward neural network (FNN) with 4 layers of ReLU activation function. Evaluation was made using RMSE and MAE.

3 Data Exploration & Statistical Analysis

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

3.1 Data Preprocessing

To illustrate the methodology of this case study, 7 years of data from the Boston Blue-Bikes[4] 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":
  integer number with the unit measure in seconds, all trips longer than 24 hours (or 86400 seconds) were not taken into account as those trips are treated as faulty and not representative of the bike sharing system flow.

- "starttime":
  exact time of the bike check-out with the YYYY-MM-DD HH:MM:SS format representing the start of the bike trip.

- "stoptime"
  exact time of the bike check-in with the YYYY-MM-DD HH:MM:SS format representing the start of the bike trip.

- "start station id"
  integer number representing the unique bike station where the check-out of the

[3]https:// www.ime.usp.br/ kon/
[4]https://www.bluebikes.com/system-data

bike occured.

- "start station name"
  string representing start bike station name.

- "start station latitude"
  float number representing geographic latitude of the start station.

- "start station longitude"
  float number representing geographic longitude of the start station.

- "end station id"
  integer number representing the unique bike station where the check-in of the bike.

- "end station name"
  string representing start bike station name.

- "end station latitude"
  float number representing geographic latitude of the start station.

- "end station longitude"
  float number representing geographic longitude of the start station.

- "bikeid"
  integer number representing a unique identification of each bike vehicle.

- "usertype"
  string, either "Subscriber" or "Customer" where Subscriber are subscribed to use bikes for a longer period (monthly or annual) of time while the Customers only pay for a one-time (single or a day pass) usage.

- "birth year"
  integer number as a year of birth of the particular user, in case they provided one.

- "gender"
  Binary integer value, "0" for female and "1" for male, self reported by member.

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.

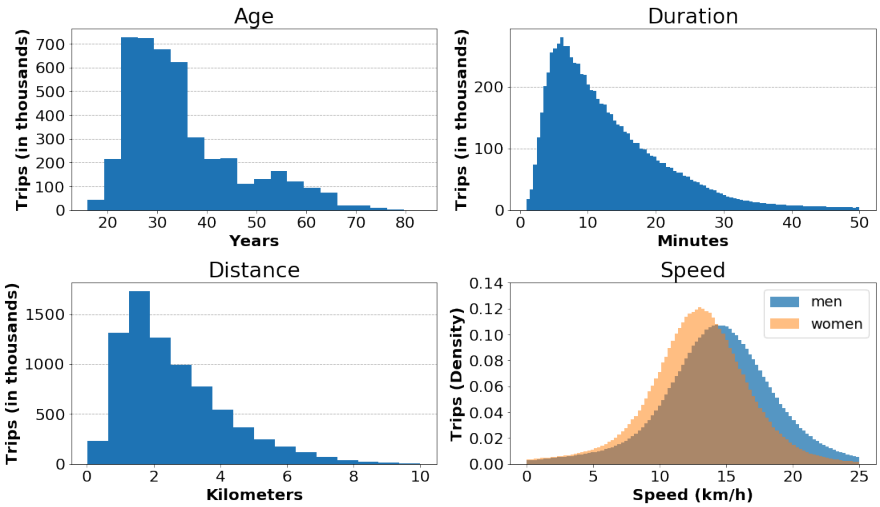

Figure 1: Descriptive Statistics for Boston Blue Bikes data

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.[25][26] However, it is still too soon to speculate if these will be a trend in the long run for the Boston area as well.

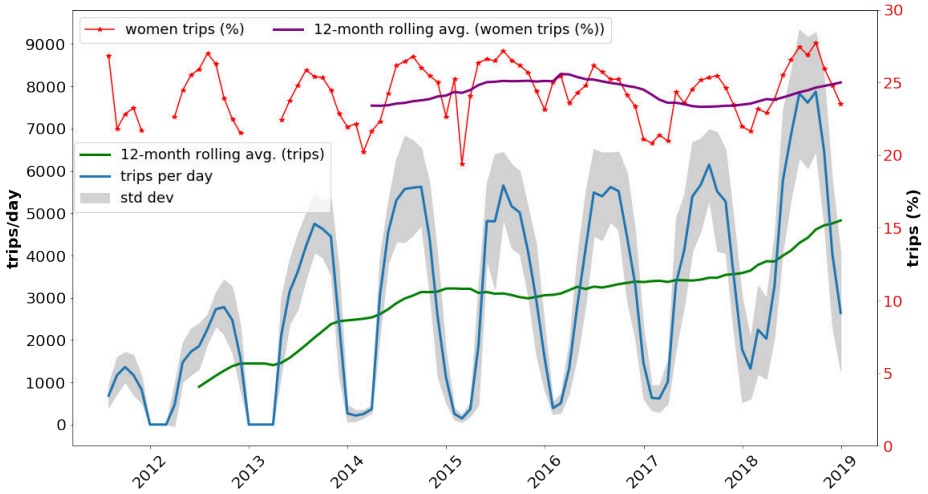

Figure 2: Evolution of trips from April 2013 to January 2019

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. Another option is to calculate distance based on the start and end point of longitude and latitude using haversine formula[27][28]. The formula is given below:

$$a = \sin^2(\frac{\Delta\varphi}{2} + \cos\varphi_1 * \cos\varphi_2 * \sin^2(\frac{\delta\lambda}{2}))$$

$$c = 2 * a \tan 2(\sqrt{a}, \sqrt{(1-a)})$$

$$d = R * c$$

where ϕ is latitude, λ is longitude, and R is Earth's radius.

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[29]. 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)[29]. 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.[21][30][19][31]

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 [32]; 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.

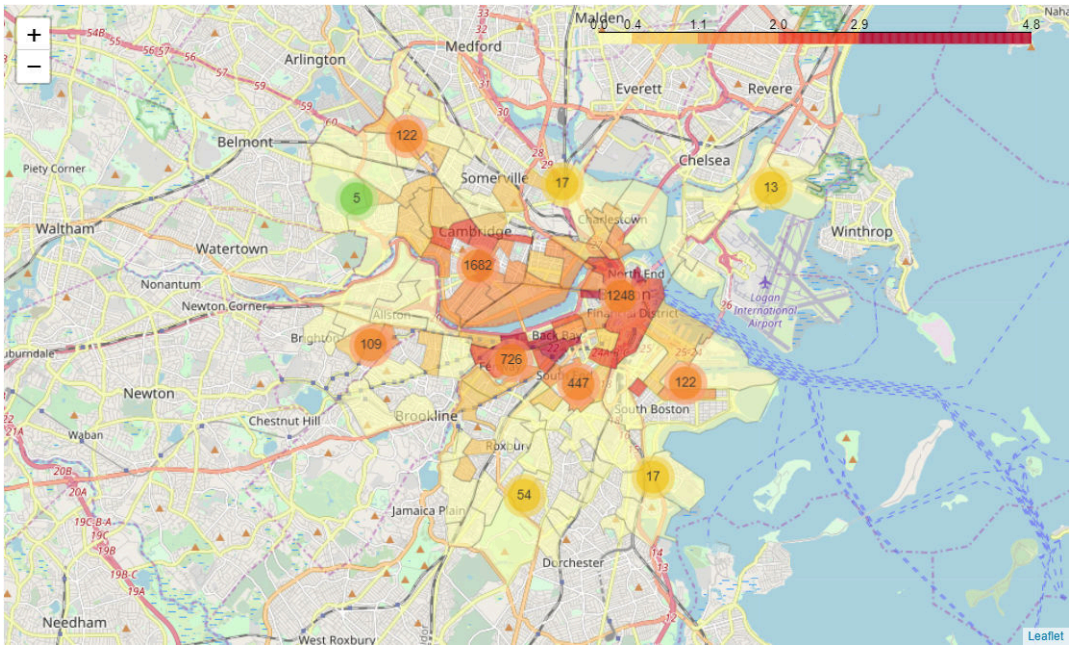

Figure 3: Morning trip Check-outs clustered by neighbourhoods for July 2018

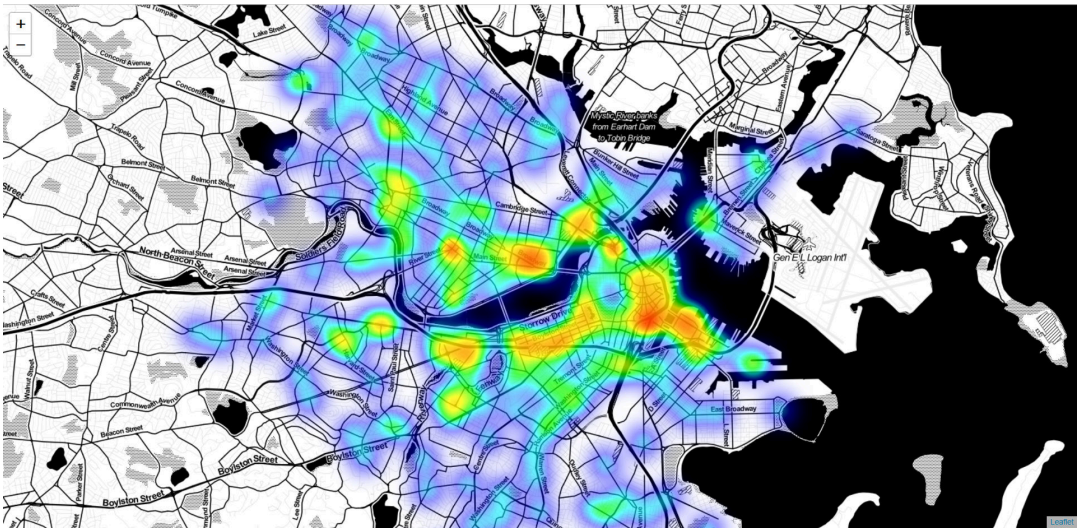

Figure 4: Morning trip Check-outs heat map for July 2018

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.

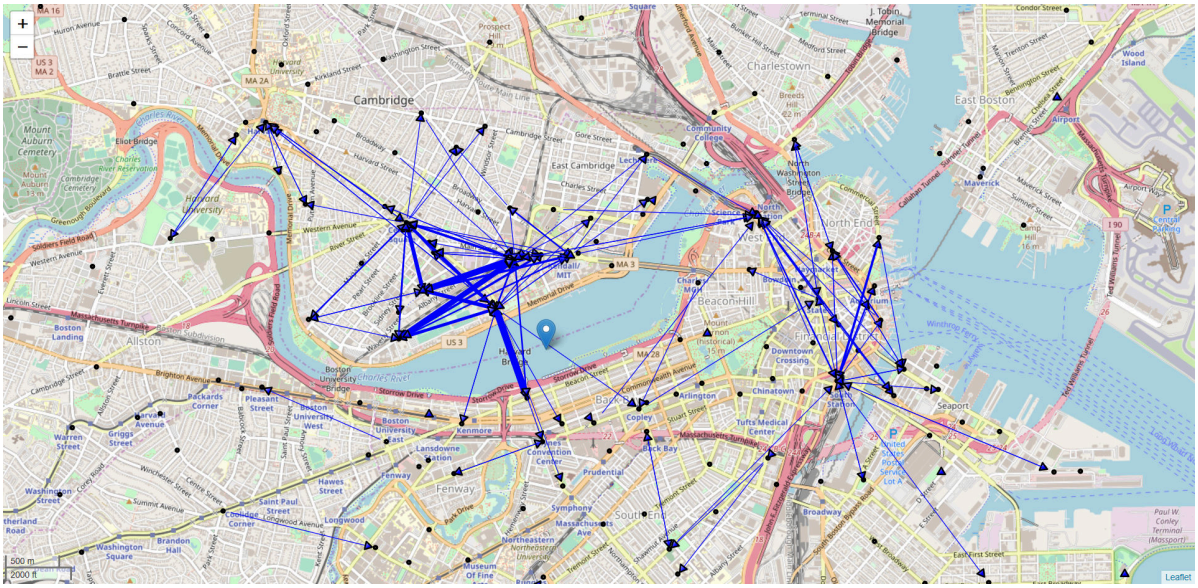

Figure 5: July 2018 Mobility Flows as a Directed Graph

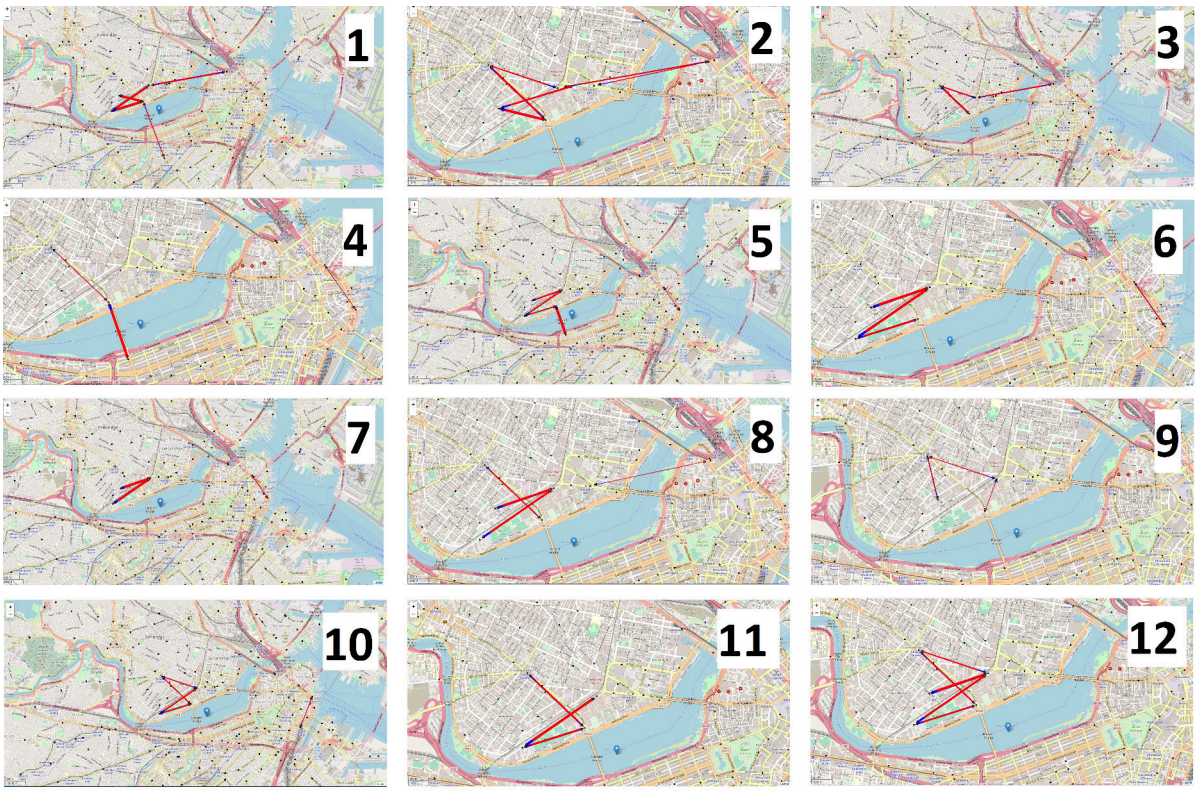

Figure 6: Most asymmetric or unbalanced links per month, 2018

```
for start id (odmatrix.row) do
    for end id (odmatrix.column) do
        start lat = stations.loc[start id,'lat'];
        start lat2 = stations.loc[end id,'lat'];
        start long = stations.loc[start id,'lon'];
        start long2 = stations.loc[end id,'lon'];
        end lat = stations.loc[end id,'lat'];
        end lat2 = stations.loc[start id,'lat'];
        end long = stations.loc[end id,'lon'];
        end long2 = stations.loc[start id,'lon'];
        num trips = odmatrix.loc[start id, end id];
        num trips2 = odmatrix.loc[end id, start id];
        if (abs(num trips - num trips2) > K) then
            draw arrow(start lat2, start long2, end lat2, end long2, num trips2);
            draw arrow2(start lat, start long, end lat, end long, num trips);
        end
    end
end
```

Algorithm 1: Most unbalanced links

In 1 there is a code snippet of the algorithm used to find the most unbalanced links in the network whose absolute pairwise flow difference is larger than a critical number K. Two for loops are going through the origin-destination matrix of stations where flows are stored as an integer number. Every combination of two stations is examined and for each one, coordinates are stored and flows are taken into account as number of trips. In the if statement, we check if the difference between the station pairs is larger than K and if it is, we are drawing the bi-directional arrow to mark the unbalanced link.

The motivation to use the most unbalanced pairs and link they form was because when comparing the unbalanced subgraphs to the data exploration heat map of rush hour check-ins/check-out as an indication of bike shortage or abundance, the spatial correlation was high. Not only that, but the more unbalanced links we observe, the more hidden problematic areas we uncover that was maybe not so visible with heat map. Also, with unbalanced pairs we know where we should add or take bike from when applying the relocation policy which is not possible to do with just observing the clusters of checked-out bikes. Moreover, having unbalanced pairs defined as a subgraph is a data structure that will be important when using as an input for CNN in Chapter 6.

4 System Architecture

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

Jupyter[5] - Web-based interactive computational environment for creating Jupyter notebook documents

TensorFlow[6] (TF) - Machine Learning framework for Python

Keras[7] - High-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano

Scikit-learn[8] - Machine learning library for the Python programming language

Seaborn[9] - Python data visualization library based on matplotlib

Short overview of the methods used in a pipeline is described here:

Firstly, raw data is obtained and prepared, cleaned and made suitable for performing time-series analysis. Secondary data such as weather is aggregated and the benchmark and base case is found in an ARIMA method that uses basic data properties like mean and average to setup the most rudimentary performance scores that are defined as the lowest one we should get with any other method to be compared with it. Simple recurrent neural network and deep recurrent neural network are used but ultimately long-short term memory is found to be the most advanced one to get good prediction scores.

Then, based on the data exploration, most unbalanced link between pairs of stations are defined and transformed into adjacency matrices. Each adjacency matrix represents top n unbalanced links in monthly granulation manner, also called snapshots.

Once again, we use data exploration for first 10 days of the month we want to predict the overall snapshot for. First 10 days are transformed into snapshot that we use as an input for convolutional neural network. We suppose that first 1/3 of the monthly snapshot flows tend to converge to a specific monthly shape that CNN will guess based on all the previous monthly snapshots it is trying to learn from. As an output, CNN gives as a guess label that best describes the partial 10 day snapshot we provided to it.

Within the predicted snapshot we have predicted top unbalanced links of station pairs.

[5]https://jupyter.org/
[6]https://www.tensorflow.org/
[7]https://keras.io/
[8]https://scikit-learn.org
[9]https://seaborn.pydata.org/

For those stations, LSTM RNN will be utilized to predict two-way flows between station pairs and calculate the difference. The difference gained is an indication of an unbalance and we can use the ground thruth if it is known to see how the overall prediction went. The predicted differences are very valuable to bike sharing companies as they can plan how many additional truck to dispatch and where, during the bike relocation process, added on top of their regular routine.

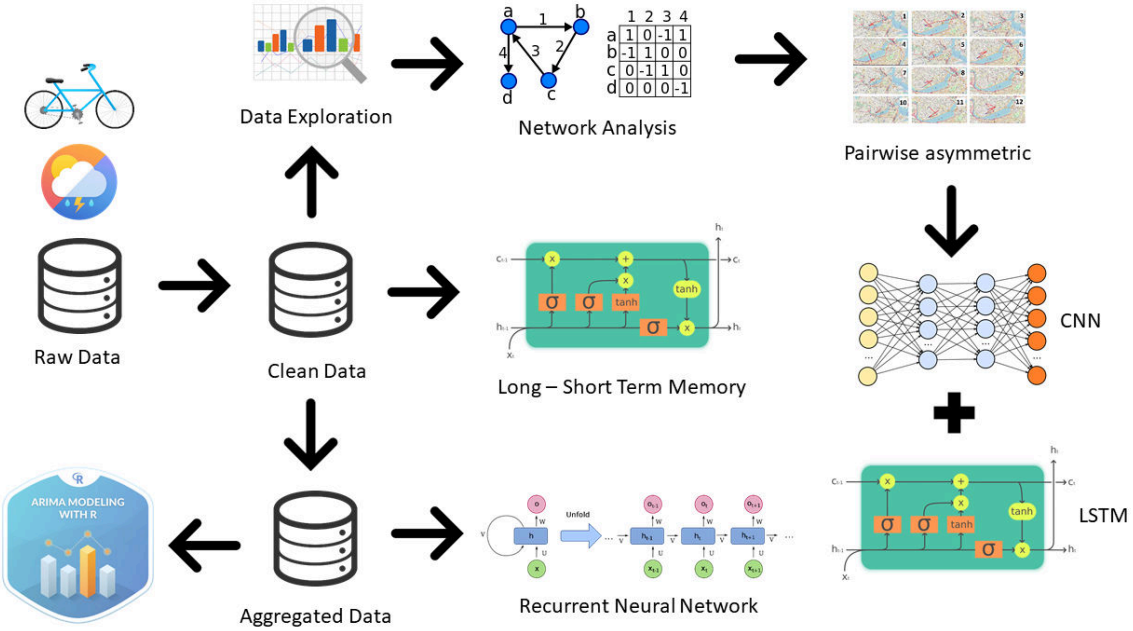

Figure 7: Proposed System Architecture & Pipeline

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"[10] website, but originally scraped from "Weather Underground"[11] 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. Also, some of the more extreme dates with bike usage having an anomaly value were replaced with the mean of that month. This is because an extreme weather occurred

[10] https://www.kaggle.com/
[11] https://www.wunderground.com/

(such as heavy snowfall) that is defined as a rare and impossible to predict event. Replacement with the monthly mean will result in a better performance of neural networks.

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."[33] 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 attributes (high and low) can be used as relevant factors. In Figure 8 we can see that low temperature is positively correlated with the usage of bikes, while in Figure 9 we can see no correlation with high humidity. Table 1 contains all the correlation coefficients between different attributes and bike usage.

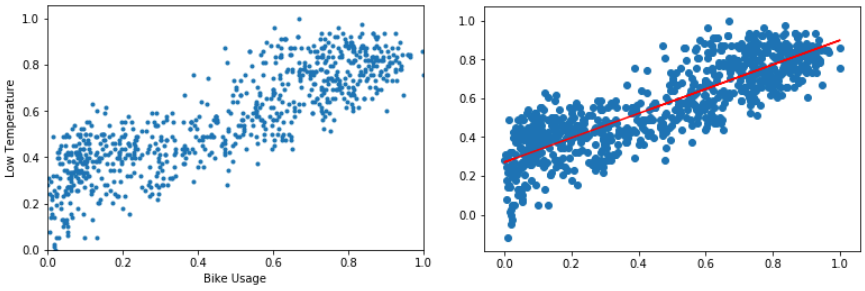

Figure 8: Correlation (0.7) between Low Temperature and Bike Usage

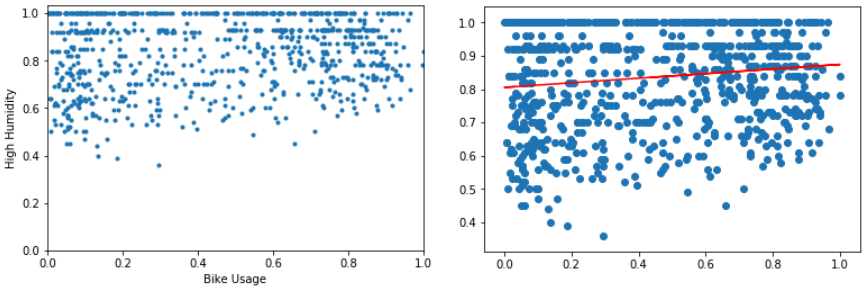

Figure 9: Correlation (0.02) between High Humidity and Bike Usage

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[34][35][36] 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[37][38] 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[39] 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[40]. In 10 and 11 we can observe and confirm these assumptions.

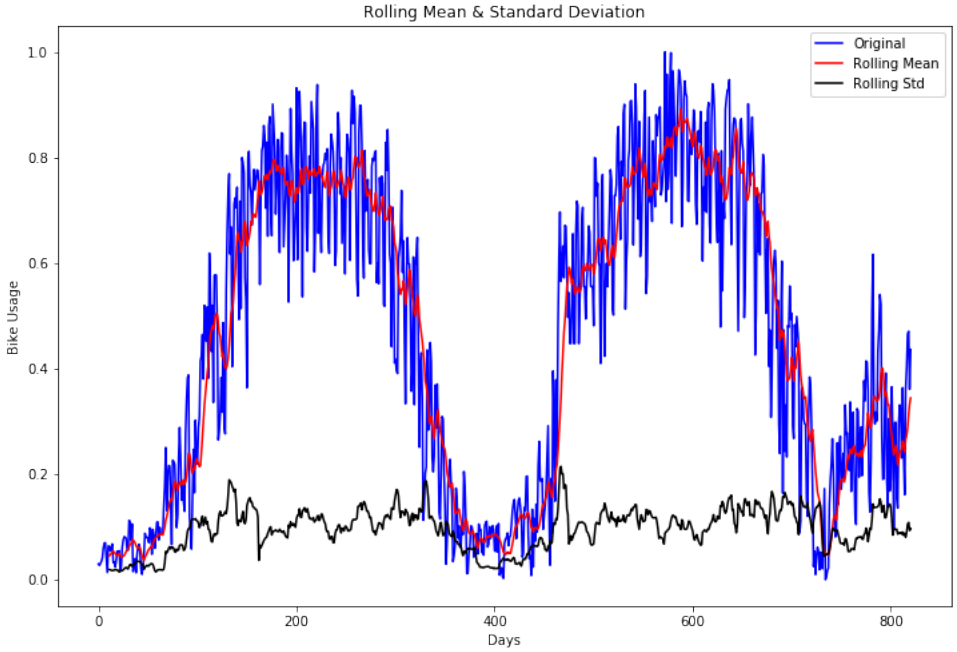

Figure 10: Rolling mean and standard deviation in ARIMA modelling

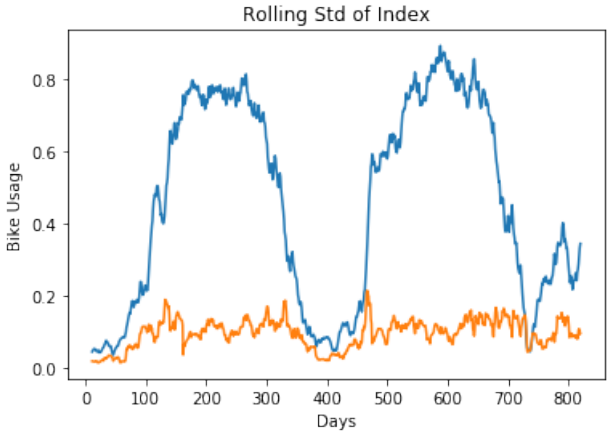

Figure 11: ARIMA model stationarity

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 12.
Auto-correlation function (ACF) explains how well the present values of the series are related to its past values. We can see 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 13 is the auto-correlation plot in case of the extreme case of choosing maximum number of lags (820). According to the literaure [41] 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"[12]. Again, as described in theory [42], we would expect the plot to show a strong relationship to the first lag and then suddenly trailing off of correlation afterwards.

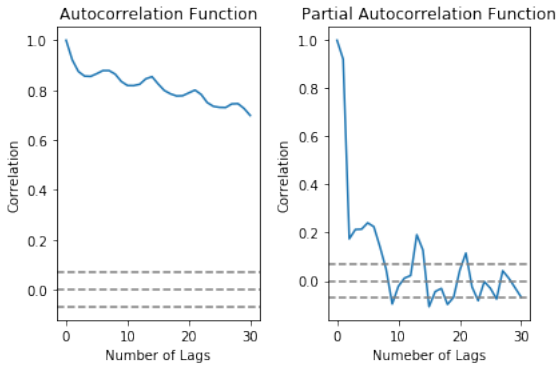

Figure 12: Autocorrelation functions for 30 lags

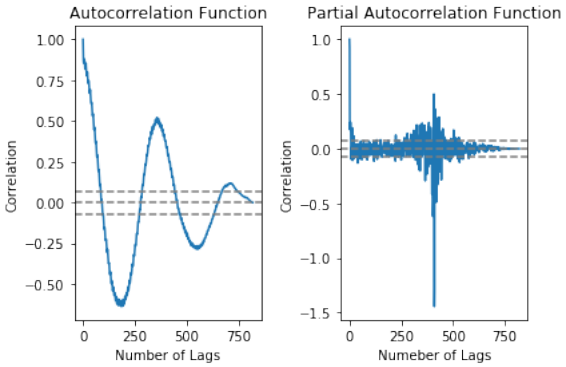

Figure 13: Autocorrelation functions for 820 lags

Finally, ARIMA model can be built based upon all the previous calculated necessities.

[12]https://machinelearningmastery.com/

Figure 14 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 [43]:

- 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.

In Tables 3 and 4, evaluationn metrics are summarized for ARIMA methods with two different settings of numbers of lags - (2,p,q) and (20,p,q).

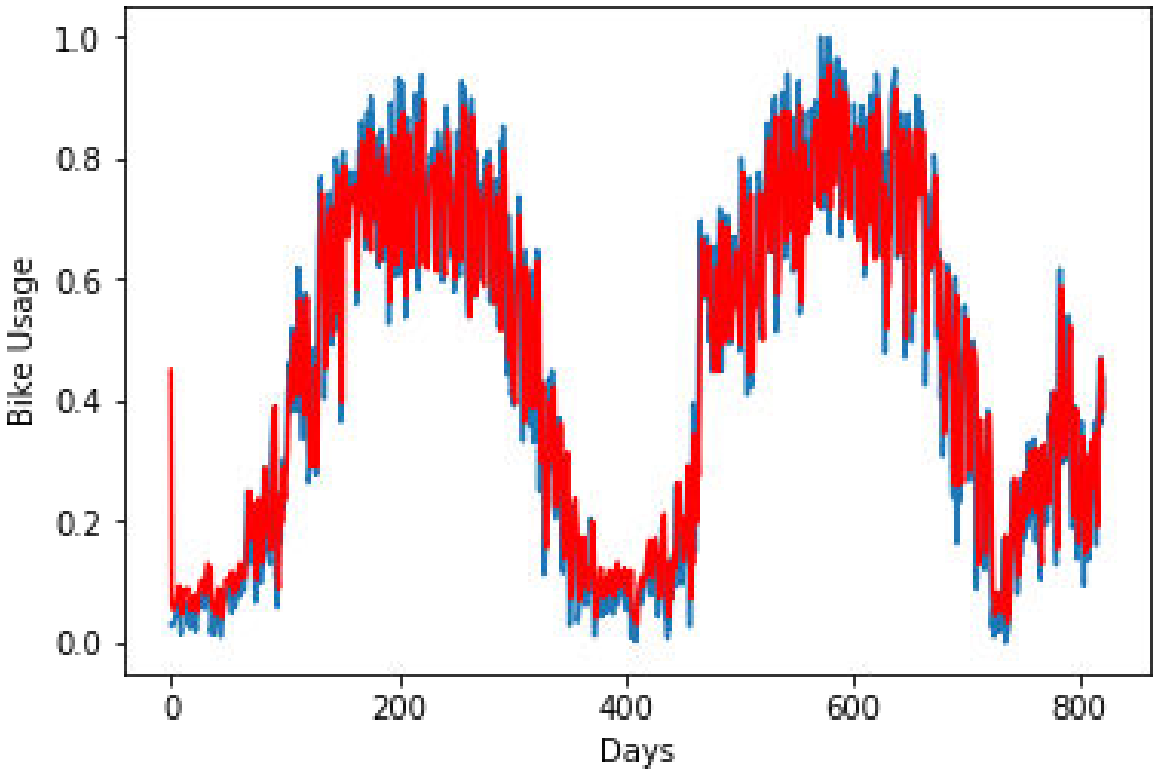

Figure 14: ARIMA Predicted Bike Flow (2,0,0)

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[44]. One big drawback of a simple RNN is that it has a vanishing gradient problem[45]. In Figure 15 we can see the simple RNN prediction of bike usage in blue and the ground truth in orange for the test data. In 16 training and validation loss functions are shown. Table 5 comprises of different simple RNN settings and results that are achieved when running them.

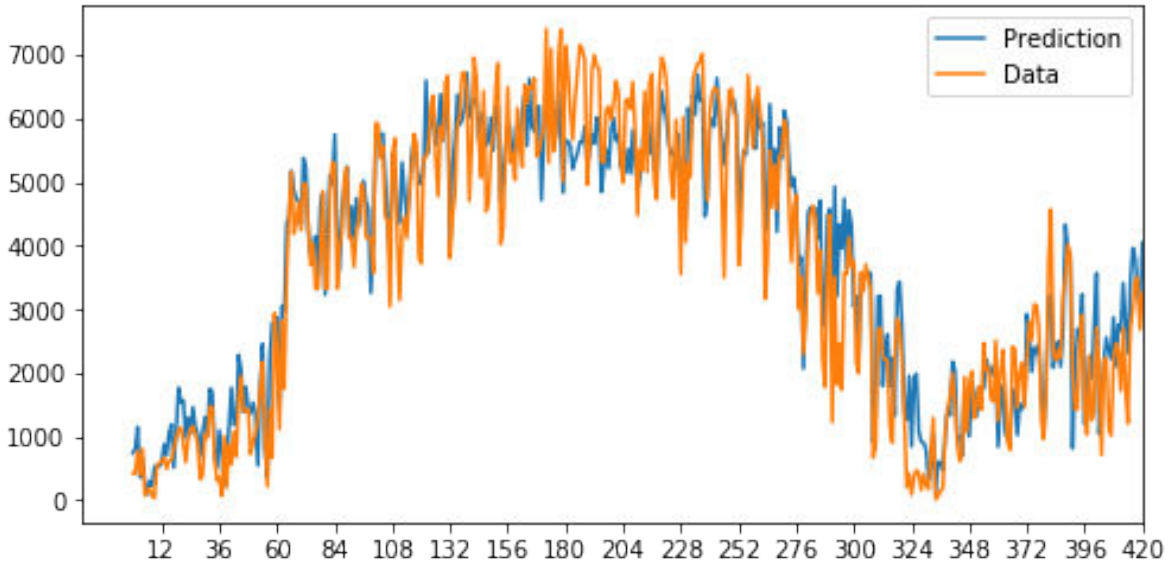

Figure 15: Predicted Bike Flow with simple RNN (x-axis = Days, y-axis = Bike usage)

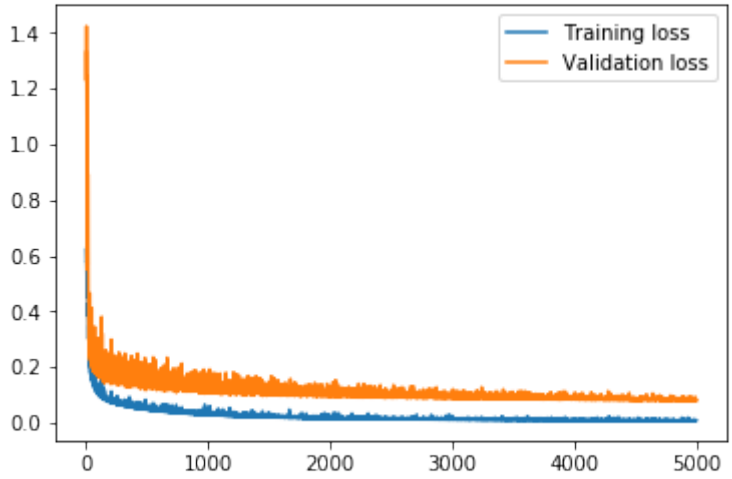

Figure 16: Simple RNN Loss Functions (x-axis = Epochs, y-axis = Loss value)

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. They benefit from the depth and outperform the conventional, shallow RNNs. [46] However, deep neural networks are often much harder to train than shallow neural networks[47]. 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 [48]. 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. In Figure 17 prediction with deep RNN can be seen, Figure 18 shows the prediction of the pre-defined hold-out data which is a form of cross validation to ensure that predictions score is really consistent throughout the data, and Figure 19 represents the two loss functions and their values as the looping through epochs is running. Finally, in Table 6, all the evaluation metrics and deep RNN settings are enlisted such as: activation functions, number of hidden nodes in each layer, number of epochs etc.

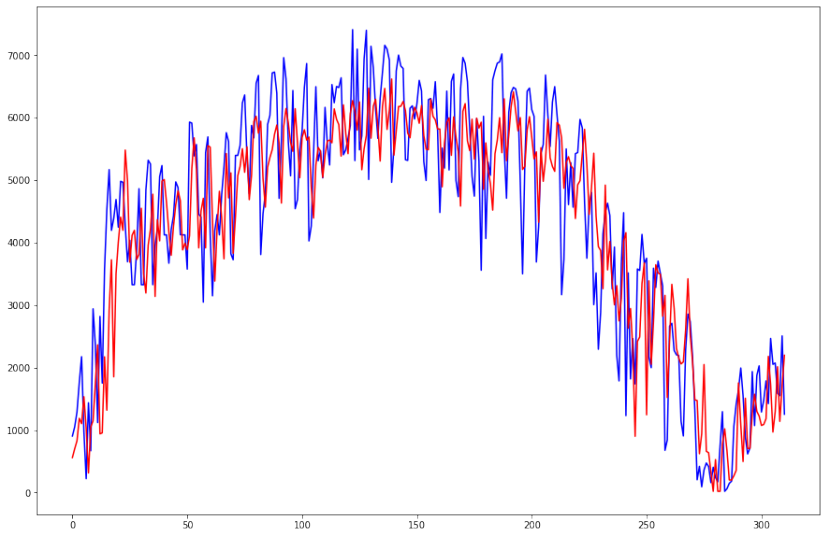

Figure 17: Deep RNN test dataset predicted bike usage

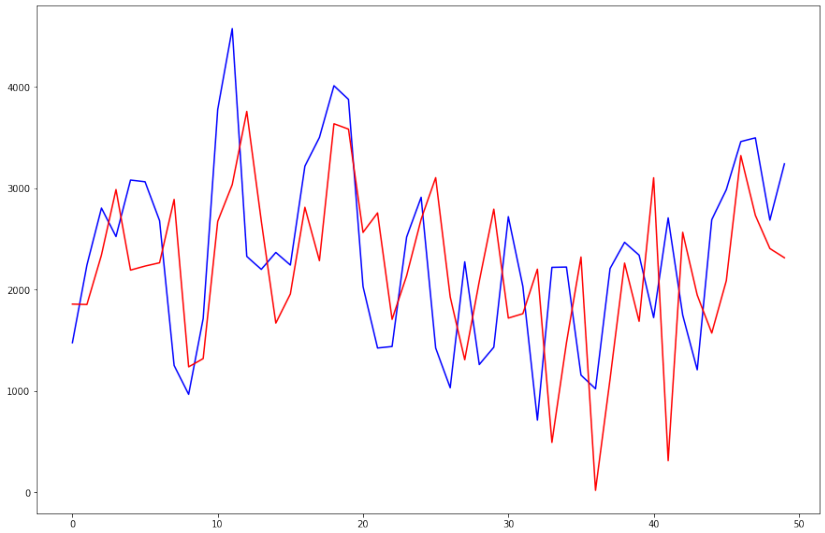

Figure 18: Deep RNN holdout dataset predicted bike usage with 65% accuracy

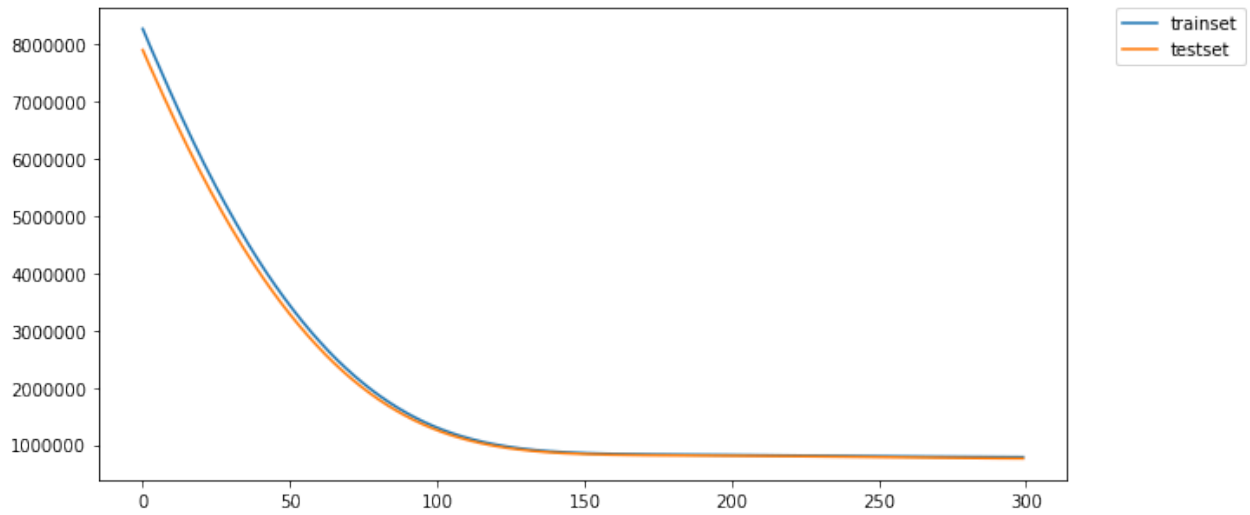

Figure 19: Deep RNN loss functions where x-axis = number of epochs, y-axis = loss value

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. 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 [49]. In Figure 20 prediction of scaled bike usage with LSTM can be observed, while in Table 7 performance metrics and LSTM settings are recorded.

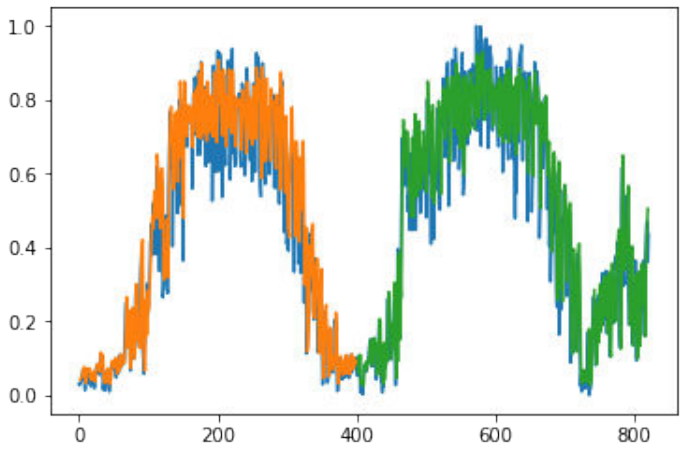

Figure 20: RNN LSTM predicted bike usage, orange = training set, green = test set

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	
Activation functions	tanh,relu			
Hidden Nodes	16,12			
Elapsed time	38.7 sec			
MAE	102.5375			
MSE	463712.94			
RMSE	680.9647			
Accuracy	0.88235			

As LSTM will be the method used for predicting dynamic patterns in this thesis due to its excellent performance, a cross validation of its results will be checked in order to make sure that prediction has reasonable values throughout the test dataset.

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 [50]. 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 21, graphs in the first row have a training set of size 500 and a sliding test set of size 107. From left to right scores obtained by using parameters seen in Iteration 1 of 7 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.

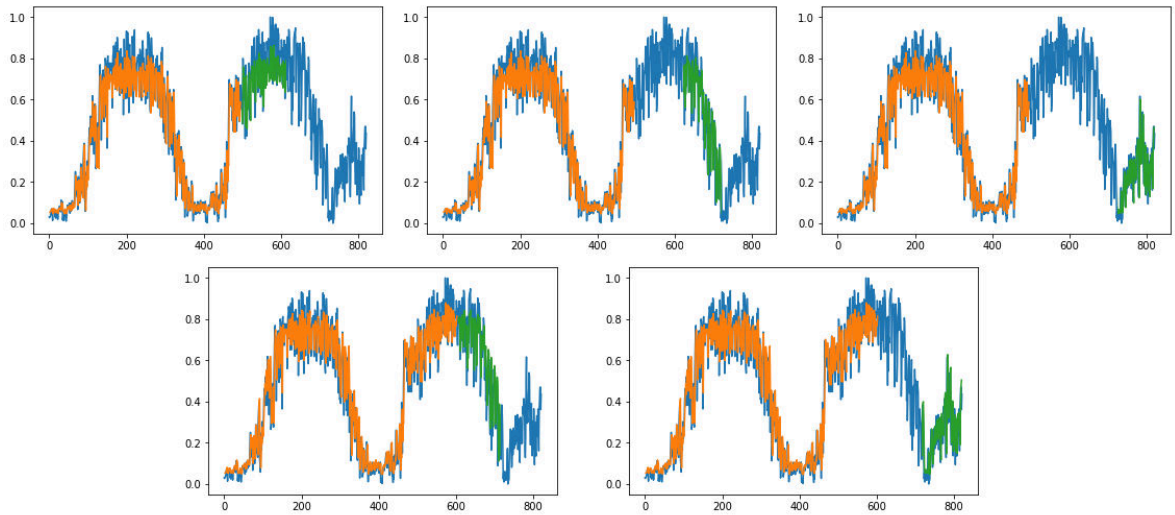

Figure 21: LSTM cross validation

Other evaluation metrics used in this chapter include Mean Absolute Error (MAE) and Root mean squared error (RMSE) which are two of the most common metrics used to measure accuracy for continuous variables. Mean Absolute Error measures the average magnitude of the errors in a set of predictions, without considering their direction, its the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight. Root mean squared error is a quadratic scoring rule that also measures the average magnitude of the error. Its the square root of the average of squared differences between prediction and actual observation. Both MAE and RMSE are negatively-oriented scores, which means lower values are better. Taking the square root of the average squared errors has some interesting implications for RMSE. Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. This means the RMSE should be more useful when large errors are particularly undesirable[51] .

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y_j}|$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

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 and mostly applied in image recognition and classification areas. Convolutional neural network architectures are usually built with the following layers: convolution layer, rectified linear units (ReLU) layer, pooling layer, fully connected layer and loss layer[52]. 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 in Table 8 and Table 9 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 22 we can observe most unbalanced stations and network edges for year 2017 and in Image24 we can see the same for year 2018.

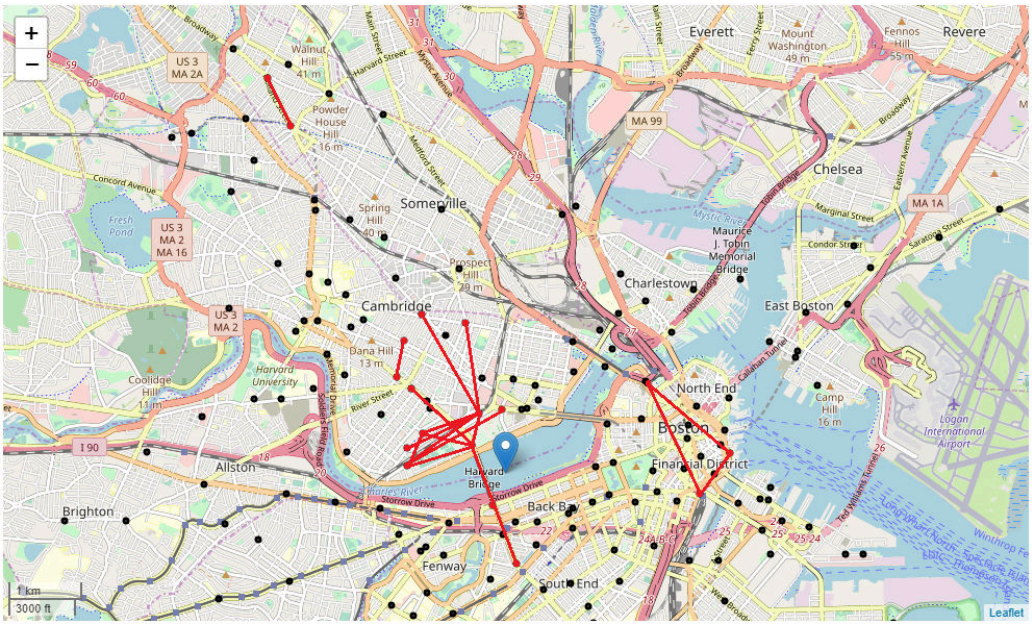

Figure 22: Aggregated unbalanced edges for year 2017

Additionally, we can perform Hyperlink-Induced Topic Search (HITS) algorithm which is a link analysis algorithm that rates hubs and authorities of a given network. For instance, if we take year 2018 as an example, in Figure 23 we can see the obtained hubs and authorities scores. In this particular case, node number 4 (Vassar bike station) is the best hub or the node that has the most bike check-ins from a variety of different stations. And node number 3 (Stata bike station) is the best authority or the node that has the most bike check-outs from a vast number of other stations.

```
({1: 0.08633351432455826,
  2: 0.345559462823765,
  3: 0.06167139846734633,
  5: 0.061717045899058555,
  4: 0.39027931173320785,
  7: 0.028118292319291003,
  6: 0.026320974432773113,
  8: -0.0,
  9: -0.0},
```

```
{1: 0.0783461996348582,
 2: 0.053509728840955416,
 3: 0.6127003762114566,
 5: 0.1845963740861543,
 4: 0.05128816508020135,
 7: 0.004887076079568587,
 6: 0.014672080066805546,
 8: 0.0,
 9: 0.0})
```

```
1: Central
2: Pacific
3: Stata
4: Vassar
5: Mass
6: Beacon
7: Nashua
8: Cross
9: Rowes
```

Figure 23: Hubs (left) and Authorities (middle) scores for 2018

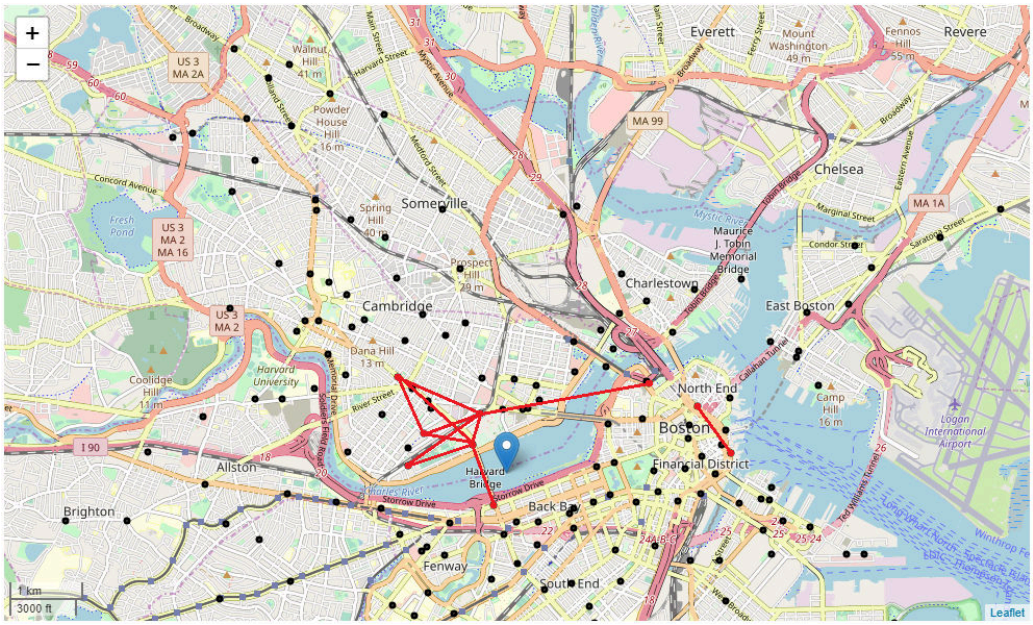

Figure 24: Aggregated unbalanced edges for year 2018

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 seen on Image25. 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 8 and Table 9 are translated into Table 11 where each station is represented as its distinct ID number (shown in Table10) 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 and hollow matrix.

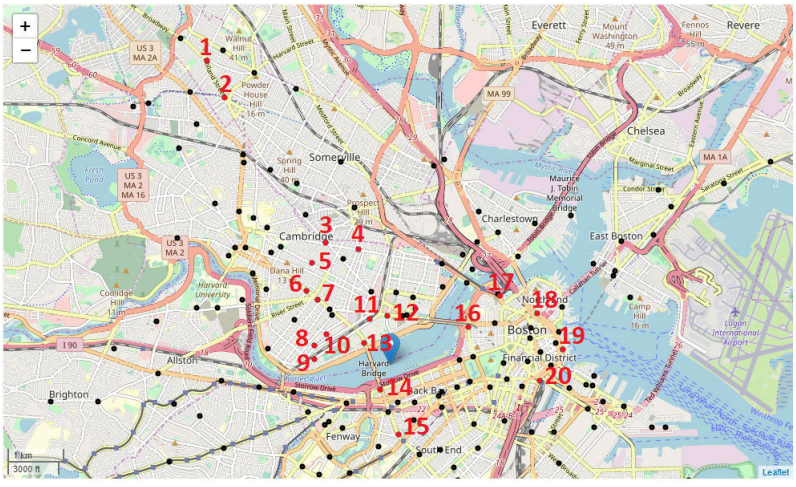

Figure 25: Appearances of all unbalanced stations during both 2017 and 2018

Table 10: Encoding the stations

Station name	**Identification Number**
Teele	1
Davis	2
Inman	3
Columbia	4
359 Broadway	5
Post Office at Central	6
Central	7
Sidney Research	8
Vassar	9
Pacific	10
Stata	11
Ames	12
Mass	13
Beacon	14
Boylston	15
Charles	16
Nashua	17
Cross	18
Rowes	19
South Station	20

Table 11: Encoded most unbalanced station pairs in 2017 and 2018

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 and also hollow meaning that we can observe only zero values on the main diagonal, thus avoiding node self-loops. 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.

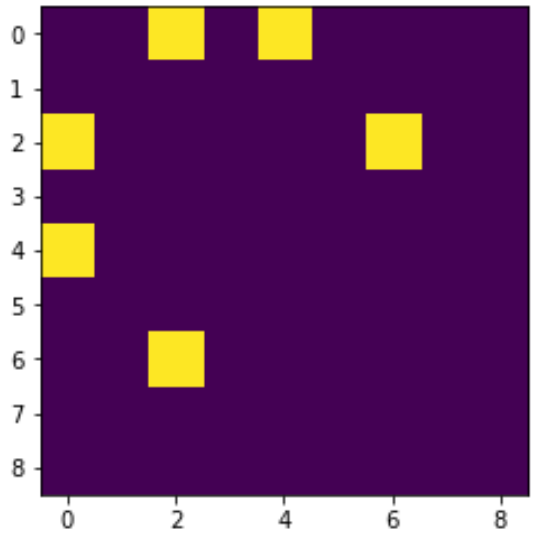

Figure 26: Adjacency matrix for March 2018 with isolated stations for that year

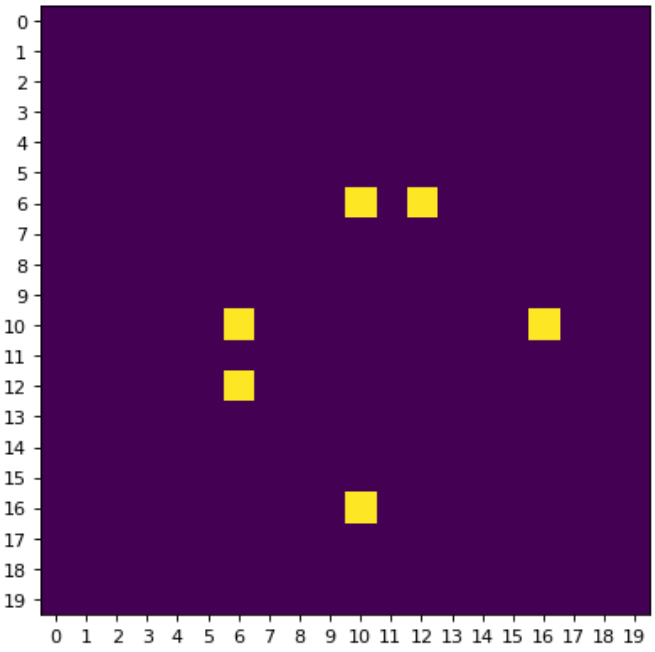

Figure 27: Adjacency matrix for March 2018 combined with stations from both years

For example, in Image 26 there is a matrix representing a particular month (March of 2018 in this case) 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.

When we want to observe a two year period of unbalanced snapshots, of course that means that we will have some new stations appearing in the unbalanced pairs. Thus, our matrices will grow in order to accommodate this new stations. In Image 27 we can see the same stations as in Image 26 but with an expanded matrix which means that encoding of the stations with their identification numbers will be slightly different, but the spatial configuration pattern will never lose its shape. The specific way the 20x20 has its space configured and defined based on the location of station spatially across the area can be seen in Figure 28. Three cities are found in dividing the matrix: Somerville, Cambridge, and Boston. Cambridge has certain concentration of observed unbalanced stations around Inman Square area, Central Square area, shore area of Cambridgeport, and Campus area of MIT. Boston has a Port Bay area just across the Harvard bridge, North Boston consisting of West End and central Boston area, South Boston between Waterfront and Financial district.

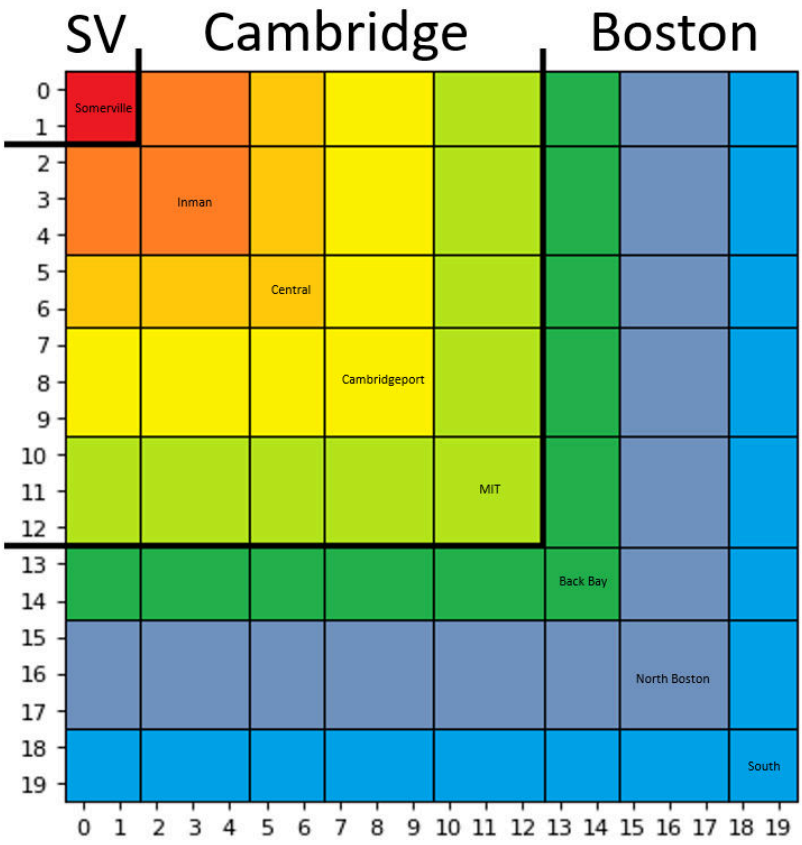

Figure 28: 20x20 unbalanced matrix configuration depending on spatial position of stations

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[53]. 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[54]. 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 27 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.

Table 12: Configuration of Deep CNN

Type	**Structure**
Input	24 x 20 x 20
Conv	filter 12 x 5 x 5, stride 2 x 2, ReLu
Pool	Max, 2 x 2
Dropout	p = 0.12
Conv	filter 24 x 5 x 5, stride 2 x 2, ReLu
Pool	Max, 2 x 2
Conv	filter 48 x 5 x 5, stride 2 x 2, ReLu
Pool	Max, 2 x 2
FC	120, ReLu
Dropout	p = 0.5
FC	84, ReLu
softmax	21

Convolutional layers consists of a specified amount convolutional filters applied to the image. These layers perform mathematical operations and an output feature map is given as a result. Typically, ReLU activation function is used to the output as it converges faster.

Pooling layers reduce the dimensionality of the feature map which decreases processing time. A commonly used pooling algorithm is max pooling, which extracts 2x2-pixel tiles, keeps their maximum value, and discards all other values.

Dense or fully connected layers perform classification on the features after convolutional and pooling layers have been used. In a dense layer, every node in the layer is connected to every node in the previous layer.

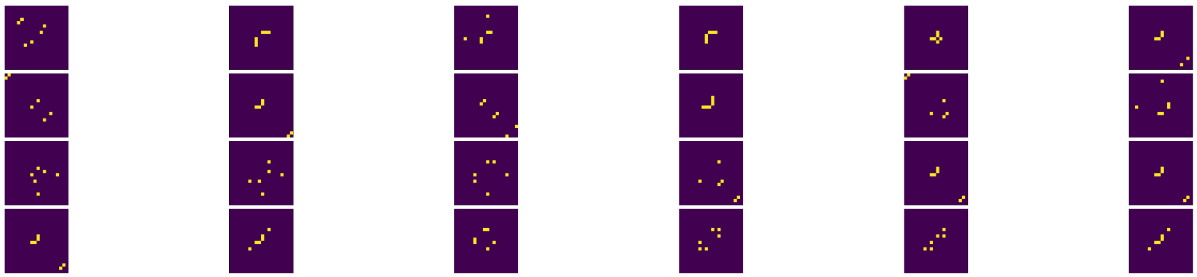

Figure 29: Training data

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 30 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.

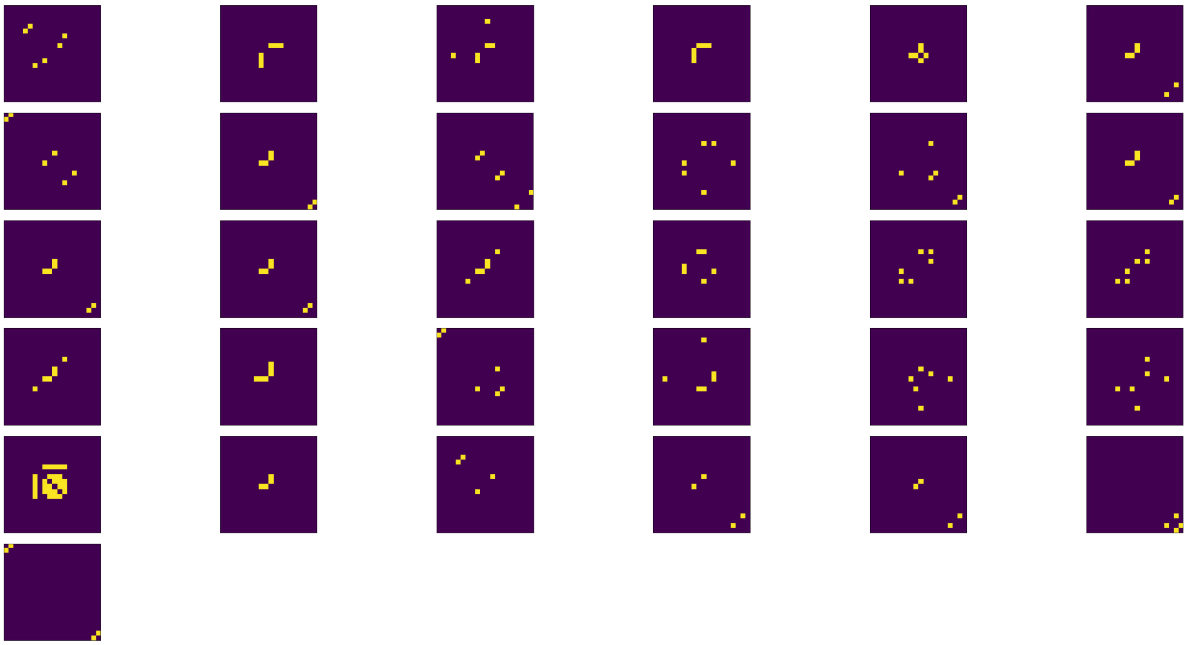

Figure 30: Test data

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[55]. 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.

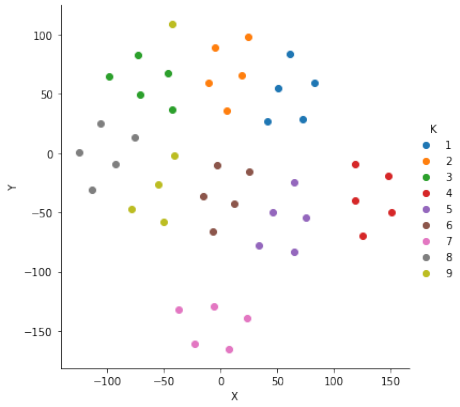

Figure 31: Stochastic Neighbor Embedding for isolated 2018 training set

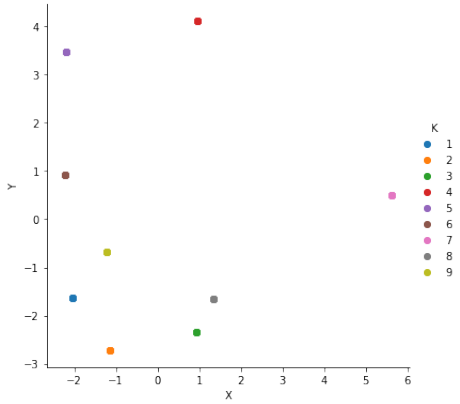

Figure 32: Principal Component Analysis for isolated 2018 training set

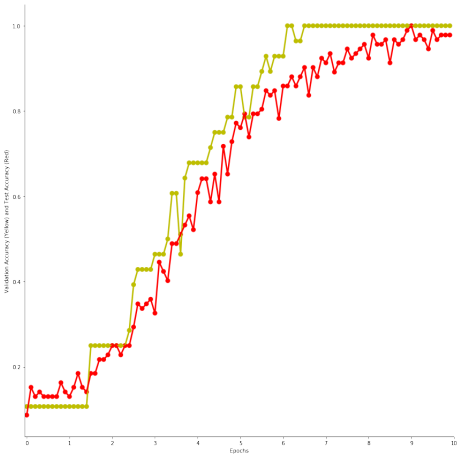

Figure 33: Validation (yellow) and test (red) accuracy

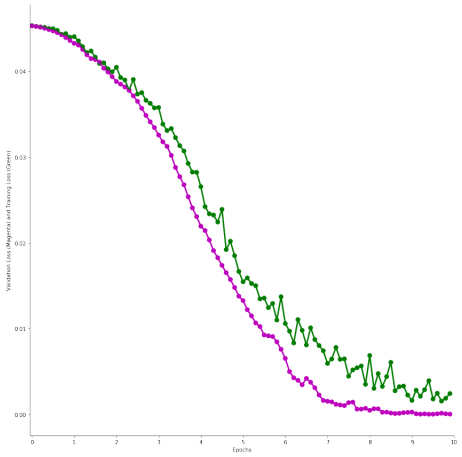

Figure 34: Validation (magenta) and test (green) loss

As an additional detail, before running the CNN, we can explore t-SNE and PCA algorithms.

T-Distributed Stochastic Neighbor Embedding is a non-linear technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets. This algorithm attempts to find patterns in the data by identifying observed clusters based on similarity of data points with multiple features. However, after this process, the input features are no longer identifiable, and you cannot make any inference based only on the output of t-SNE. Hence, it is mainly a data exploration and visualization technique[56]. Principal Component Analysis or PCA is a linear feature extraction technique. It combines input features in a specific way that it is possible drop the least important feature while still retaining the most valuable parts of all of the features. As an added benefit, each of the new features or components created after PCA are all independent of one another[57].

An addition to improving CNN decide which station might have greater importance before deciding to label the image is to add weights to each pixel in the image representing bike usage for that month. In that case, our training set of matrices gets pixels coloured based on the intensity of asymmetrical balance between certain stations.

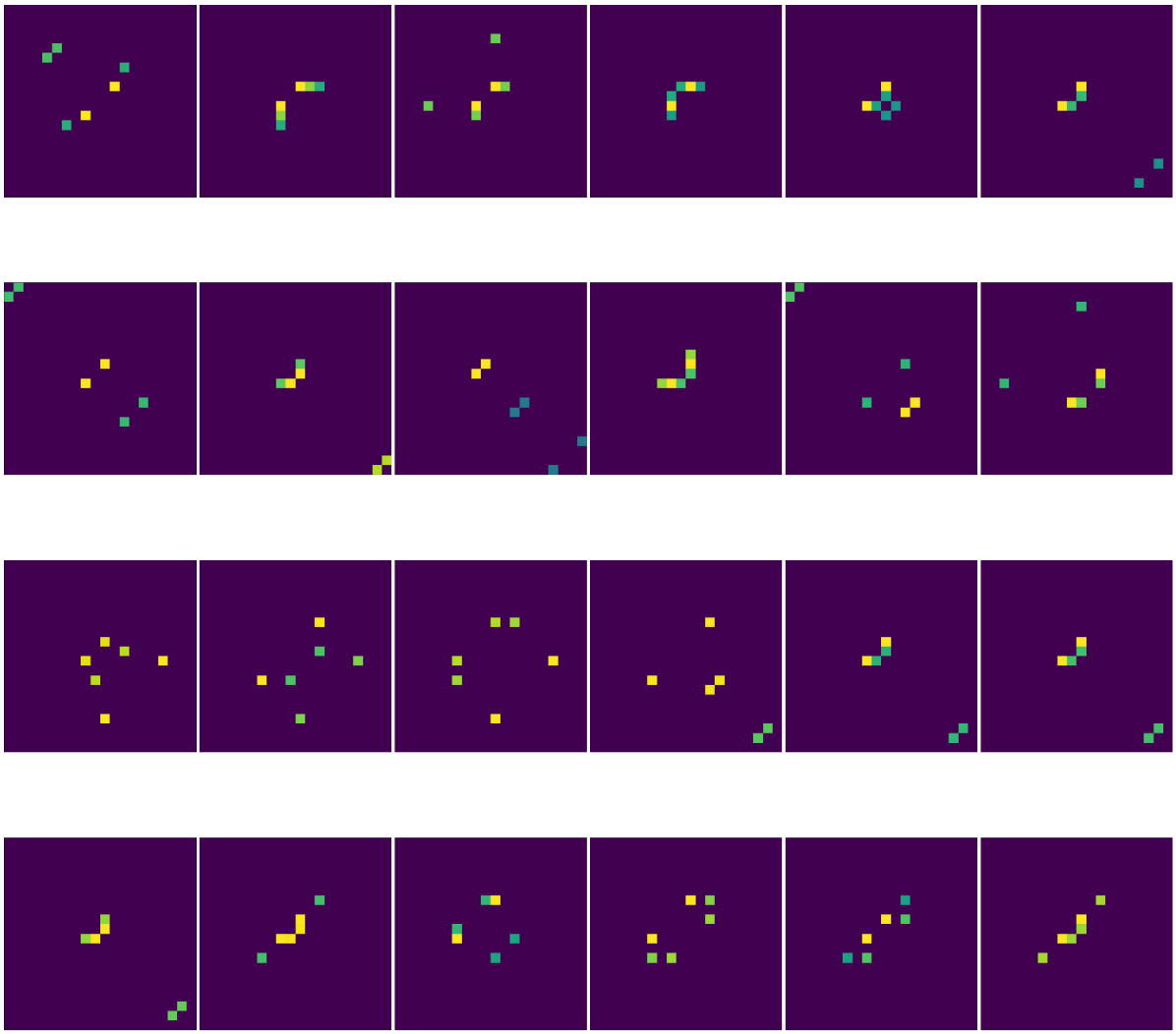

Figure 35: Weighted training set

7 Discussion

7.1 Results

The methods described in the previous chapters will be applied for the Boston use case in order to predict spatial structures and dynamic patterns. For this purpose, months of January, February, March and April of the year 2019 will be used as a validation test to discuss the performance of the thesis project.

First, January (69'872 trips in total) will be taken and split into two sets. Smaller (training) set will consist of the time period when 1/3 of bike trips were cycled, generally approximated by the first ten days of the month and for those first 10 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 (or a known station in the pair is represented as a loop pixel in the adjacency matrix). But, of course in an ideal case, new stations should be added and matrices expanded as those stations could start appearing in a top 3 list for the future months and years. This is also true in case top 3 lists would be expanded to a top n lists. Such thing is feasible but would require manual expansion and filling in of the matrices.

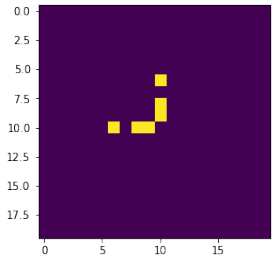

Figure 36: First 10 days of January 2019 unbalanced links: (10,11), (11,7), (11,9)

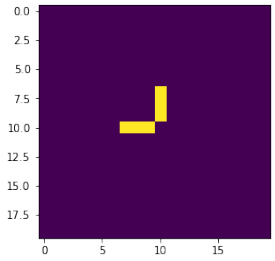

Figure 37: CNN prediction for the whole month of January 2019: (10,11), (11,8), (11,9)

CNN will try and classify the matrix of first week of 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 whole month of January as we assume that identified unbalanced flows

usually tend to converge to the predicted labels.

Next, for the stations that are found in the predicted matrix, RNN method is utilized as we want to find the predicted flow dynamic for each. To get the most unbalanced scores, we subtract the predicted flows from the unbalanced pairs. When using RNN to predict flows between station Pacific (10) and station Stata (11), we used time range from January 2016 to April 2019, and the training data consisted of 0.8945% of the total data to match that the test data fits with the whole year of 2019. which is 120 days for first 4 months. Setting used included: 400 epochs, 12 ReLU activation functions in the first hidden layer and 8 tanh activation functions in the second hidden layer. Prediction results achieved for the last 21 days (three weeks) of January were 78.38%. Cumulative sum of the bike usage during those two weeks was 241.05843 predicted, and the ground truth is 251. This is for the bike traveling from Pacific station (10) to Stata station (11).

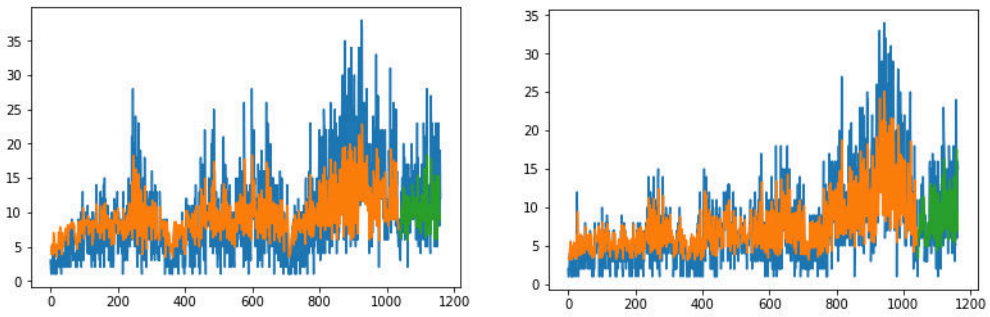

Figure 38: Pacific to Stata flow (left) and Stata to Pacific flow (right)

Now, the same thing needs to be done in the reverse flow direction, from station Stata (11) to station Pacific (10). For this flow are getting a prediction with 77.43% accuracy that the cumulative sum of bike usage between January 10 and January 31 will be 156.40091, while the ground truth is 162.

Finally, we can now observe what we predicted and see that we have a difference of flows between station 10 and 11 is 241.05843 - 156.40091 = 84,65752 bikes (rounded to 85), thus making the station Pacific more sparse with bikes. The difference for the ground truth was calculated by observing the real captured data from January 10 to January 31. The observed value for this difference is 68. The precision score for predicting this asymmetry is 80.0%.
Same procedure is repeated for other pairs of stations.
For stations Stata (11) and Vassar (9) the predicted values are: 219.41492 (186 is actual value) with 65.77% accuracy from station 9 to station 11, and 145.681 (127 as an actual value) 771.54% accuracy for direction from station 11 to station 9. The predicted difference is 73.73392 (rounded to 74) and compared to the real value of 61, the overall accuracy is 82.43%.

Regarding the station pair Stata (11) and Sidney (8) that CNN gave as a result instead of Stata (11) and Central (7), the difference is not neded to be predicted or calculated as it is almost zero, hence we got a false result for this link prediction.

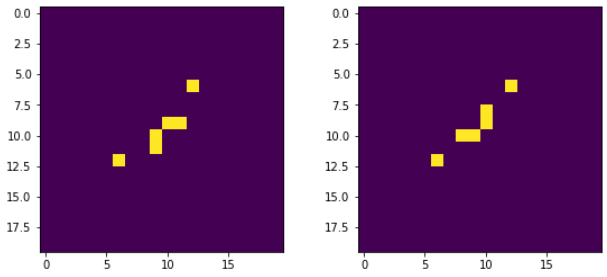

Figure 39: First 1/3 flows of February (left) and predicted rest of February (right)

Predicted unbalanced pairs of stations for February are (11,9),(10,11),(13,7). Using RNN on these pairs we get these flow differences:

(13,7) = predicted flow is 148.43597 and real flow is 146 (accuracy 98.35%)
(7,13) = predicted flow is 90.60720 and real flow is 76 (accuracy 83.87%)
Predicted difference is 57.82868, while the ground truth is 78. The overall difference accuracy is 74%.

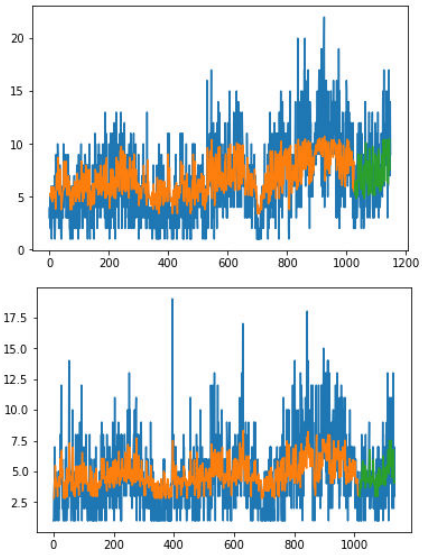

Figure 40: (13,7) Mass to Central (top) and (7,13) vice-versa (down)

(11,9) = predicted flow is 163.80017 and the real flow is 183 (accuracy 89.5%)
(9,11) = predicted flow is 226.40657 and the real flow is 222 (accuracy 98%)

Predicted difference is 62.6064, while the ground truth is 31. The overall difference accuracy is 50%. The low accuracy score we got is the consequence of falsely predicting this station pair when using CNN.

(10,11) = predicted flow is 223.32707 and the real flow is 268 (accuracy 83%)
(11,10) = predicted flow is 186.0166 and the real flow is 216 (accuracy 86%)
Predicted difference is 37.31047, while the ground truth is 49. The overall difference accuracy is 76%. However, it is important to mention that this CNN predicted pair is not found in the top 3 most unbalanced ground truth stations for the rest of February.

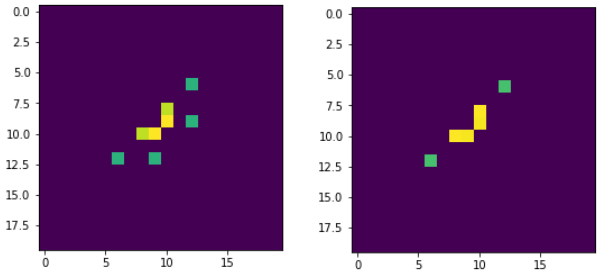

Figure 41: First 1/3 flows of March (left) and predicted rest of March (right)

Predicted unbalanced pairs of stations for March are (11,9),(10,11),(13,7). Using RNN on these pairs we get these flow differences:

(13,7) = predicted flow is 124.50949 and real flow is 125 (accuracy 99.6%)
(7,13) = predicted flow is 78.81064 and real flow is 73 (accuracy 92.6%)
Predicted difference is 45.69885, while the ground truth is 64. The overall difference accuracy is 67%. This most unbalanced pair exists as a ground truth in the top 3 most unbalanced stations for the rest of the March.

(10,11) = predicted flow is 202.1513 and real flow is 254 (accuracy 79.5%)
(11,10) = predicted flow is 183.42072 and real flow is 231 (accuracy 79.4%)
Predicted difference is 18.73058, while the ground truth is 23. The overall difference accuracy is81.47%. This unbalanced pair is not in top 3 ground truth most unbalanced stations for the rest of March.

Ground truth for the rest of the March is (13,7),(7,11),(17,11).

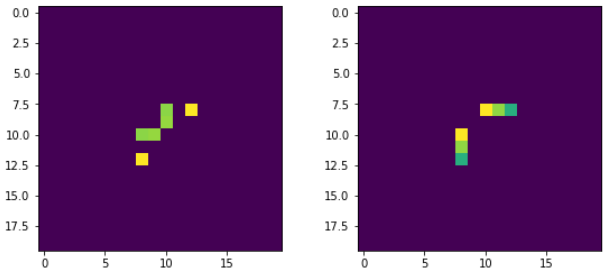

Figure 42: First 1/3 flows of April (left) and predicted rest of April (right)

Predicted unbalanced pairs of stations for April are (9,11),(9,10),(9,13). Using RNN on these pairs we get these flow differences:

(11,9) = predicted flow is 122.4496 and real flow is 131 (accuracy 93.4%)
(9,11) = predicted flow is 184.49792 and real flow is 180 (accuracy 97.5%)
Predicted difference is 62.04832, while the ground truth is 49. The overall difference accuracy is 79%. This most unbalanced pair exists as a ground truth in the top 3 most unbalanced stations for the rest of the April.

(9,13) = predicted flow is 146.04576 and real flow is 148 (accuracy 98.6%)
(13,9) = predicted flow is 168.44264 and real flow is 159 (accuracy 94.4%)
Predicted difference is 22.39688, while the ground truth is 11. The overall difference accuracy is 50%. This unbalanced pair is not in top 3 ground truth most unbalanced stations for the rest of April.

Ground truth for the rest of the March is (9,11),(7,13),(9,13).

7.2 Conclusion

Using the method described in this thesis, it is possible to use computationally lightweight combination of machine learning methods to grasp short-term approximation and prediction of spatial structures in the form of subgraphs and a more robust prediction of dynamic patterns and flows through the identified most unbalanced links. As a result, this method gives ability for bike sharing companies to make a month long prediction on how many trucks for relocating bikes will they need and in which streets or areas. The bottleneck of the project is its CNN segment due but still it offers the approach that haven't been yet considered, especially as predicting the exact future configuration can only be possible to achieve with moderately high accuracy. In a practical sense, if predicted unbalanced link between two stations for the rest of some arbitrary month is 60, that means that three bike rebalancing truck must be dispatched during peak days for those specific stations and with this method, even if one of the links is not predicted correctly, the approximate area is still highly accurate.
//TO-DO

7.3 Future Work

//TO-DO

References

[1] P. DeMaio, "Bike-sharing: History, Impacts, Models of Provision, and Future," *Public Transportation, 12 (4): 41-56*, October 1, 2009.

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

[3] D. Yanocha, "The bikesharing Planning Guide," *Institute for Transportation and Development Policy (ITDP)*, 2018.

[4] Z. Hong, A. Mittal, and H. S. Mahmassani, "Effect of Bicycle-Sharing on Public Transport Accessibility: Application to Chicago Divvy Bicycle-sharing System," *Transportation Research Board 95th Annual Meeting Transportation Research Board*, 2016.

[5] T. L. Susan Shaheen, "Reducing greenhouse emissions and fuel consumption: Sustainable approaches for surface transportation," *IATSS Res. 31,6-20*, 2007.

[6] Y. Zhang and Z. Mi, "Environmental benefits of bike sharing: A big data-based analysis," *Applied Energy, vol. 220, issue C, 296-301*, 2018.

[7] K. Saelensminde, "Cost-benefit analyses of walking and cycling track networks taking into account insecurity, health effects and external costs of motorized traffic," *Transportation Research Part A Policy and Practice 38(8):593-606*, October 2004.

[8] O. Pekka, S. Titze, A. Bauman, B. D. Geus, P. Krenn, B. Reger-Nash, and T. Kohlberger, "Health benefits of cycling: a systematic review." *Scand J Med Sci Sports.*, August 2011.

[9] R. J. Shephard, "Is active commuting the answer to population health?" *Sports Med.*, 2008.

[10] M. Peden, R. Scurfield, D. Sleet, D. Mohan, A. Hyder, E. Jarawan, and C. Mathers, "World report on road traffic injury prevention," *World Health Organization report, Geneva*, 2004.

[11] 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.

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

[13] P.-C. Chen, H.-Y. Hsieh, X. K. Sigalingging, Y.-R. Chen, and J.-S. Leu, "Prediction of station level demand in a bike sharing system using recurrent neural networks," *IEEE 85th Vehicular Technology Conference*, 2017.

[14] R. C. Zheng, "Predicting bike sharing demand using recurrent neural networks," *Procedia Computer Science 147:562-566*, 2019.

[15] L. Lin, Z. He, and S. Peeta, "Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach," *Transportation Research Part C: Emerging Technologies Volume 97*, December 2018.

[16] A. Yi, Z. Li, M. Gan, Y. Zhang, D. Yu, W. Chen, and Y. Ju, "A deep learning approach on short-term spatiotemporal distribution forecasting of dockless bike-sharing system," *Neural Computing and Applications*, April 2018.

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

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

[19] 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.

[20] 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.

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

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

[23] Z. Yang, J. Hu, Y. Shu, P. Cheng, J. Chen, and T. Moscibroda, "Mobility modeling and prediction in bike-sharing systems," *PLoS ONE*, MobiSys '16 Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services.

[24] Y. Pan, R. C. Zheng, J. Zhang, and X. Yao, "Predicting bike sharing demand using recurrent neural networks," *Procedia Computer Science 147:562-566*, January 2019.

[25] A. C. Lusk, P. G. Furth, P. Morency, L. F. Miranda-Moreno, W. C. Willett, and J. T. Dennerlein, "Risk of injury for bicycling on cycle tracks versus in the street," *BMJ Publishing Group Ltd*, 2011.

[26] J. Garrard, S. Handy, and J. Dill, "Women and cycling," *MIT Press*, 2012.

[27] K. von Lindenberg, "Comparative analysis of gps data," *The Berkeley Electronic Press*, 2013.

[28] C. C. Robusto, "The cosine-haversine formula," *The American Mathematical Monthly*, 1957.

[29] H. Coleman and K. Mizenko, "Pedestrian and bicyclist data analysis," *NHTSAs Office of Behavioral Safety Research*, March 2018.

[30] A. F. Imani, N. Eluru, A. M. El-Geneidy, M. Rabbatc, and U. Haq, "How land-use and urban form impact bicycle flows: evidence from the bicycle-sharing system (bixi) in montreal," *Journal of Transport Geography*, December 2014.

[31] X. Wang, G. Lindsey, J. E. Schoner, and A. Harrison, "Modeling bike share station activity: Effects of nearby businesses and jobs on trips to and from stations," *Journal of Urban Planning and Development*, March 2016.

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

[33] B. Ratner, "The Correlation Coefficient," *Journal of Targeting, Measurement and Analysis for Marketing*, May 18, 2009.

[34] G. P. Zhang, "Time series forecasting using a hybrid arima and neural network model," *Elsevier Neurocomputing*, 2001.

[35] U. Pritzsche, "Benchmarking of classical and machine-learning algorithms (with special emphasis on bagging and boosting approaches) for time series forecasting," *Ludwig Maximilian University of Munich*, 2015.

[36] A. A. Adebiyi, A. Adewumi, and C. Ayo, "Comparison of arima and artificial neural networks models for stock price prediction," *Journal of Applied Mathematics*, March 2014.

[37] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *Journal of the American Statistical Association*, 2011.

[38] R. Mushtaq, "Augmented dickey fuller test," *Social Science Research Network SSRN*, 2011.

[39] F. X. Diebold and L. Kilian, "Unit root tests are useful for selecting forecasting models," *Journal of Business and Economic Statistics*, 2000.

[40] E. Zivot and J. Wang, "Rolling analysis of time series," *Springer, New York, NY*, 2006.

[41] R. Adhikari and R. K. Agrawal, "An introductory study on time series modeling and forecasting," *LAP Lambert Academic Publishing, Germany*, 2013.

[42] P. J. Brockwell and R. A. Davis, "Time series: Theory and methods," *Springer, New York, NY*, 1991.

[43] E. P. George and G. M. J. , "Time series analysis: forecasting and control," *Wiley*, 1970.

[44] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, , and S. Valaee, "Recent advances in recurrent neural networks," *CoRR abs/1801.01078*, 2018.

[45] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA*, 2013.

[46] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," *ICLR*, 2014.

[47] S. S. Du, J. D. Lee, H. Li, L. Wang, and X. Zhai, "Gradient descent finds global minima of deep neural networks," *Proceedings of the 36 th International Conference on Machine Learning, Long Beach, California, PMLR 97*, 2019.

[48] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to Construct Deep Recurrent Neural Networks," *International Conference on Learning Representations*, 2014.

[49] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, 1997.

[50] C. Bergmeir and J. M. Benitez, "On the use of cross-validation for time series predictor evaluation," *Information Sciences*, 2012.

[51] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Inter-Research*, 2005.

[52] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[53] Y. Guo, J. Zhou, Y. Wu, and Z. Li, "Identifying the factors affecting bike-sharing usage and degree of satisfaction in ningbo, china," *PLoS ONE 12(9): e0185100*, 2017.

[54] R. Uehara and Y. Uno, "Efficient algorithms for the longest path problem," *Algorithms and Computation, 15th International Symposium, ISAAC 2004, Hong Kong, China*, 2004.

[55] W. Zhao, "Research on the deep learning of the small sample data based on transfer learning," *AIP Conference Proceedings*, 2017.

[56] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, 2008.

[57] A. Tharwat, "Principal component analysis - a tutorial," *IJAPR*, 2016.