Student thesis series INES nr 634

Station-level demand prediction in bike-sharing systems through machine learning and deep learning methods

Nikolaos Staikos

2023
Department of
Physical Geography and Ecosystem Science
Lund University
Sölvegatan 12
S-223 62 Lund
Sweden

Nikolaos Staikos (2023).

Station-level demand prediction in bike-sharing systems through machine learning and deep learning methods

Master degree thesis, 30 credits in *Geomatics*

Department of Physical Geography and Ecosystem Science, Lund University

Level: Master of Science (MSc)

Course duration: *January* 2023 until *June* 2023

Station-level demand prediction in bike-sharing systems through machine learning and deep learning methods

---

Nikolaos Staikos

Master thesis, 30 credits, in *Geomatics*

Supervisor:

Pengxiang Zhao

Dep. of Physical Geography and Ecosystem Science, Lund University

Co supervisor:

Ali Mansourian

Dep. of Physical Geography and Ecosystem Science, Lund University

Exam committee:

Zheng Duan

Rachid Oucheikh

Dep. of Physical Geography and Ecosystem Science, Lund University

Abstract

Nowadays, public bicycle-sharing systems have been introduced in several big cities. Even though it is an undeniably sustainable and effective means of transport, monitoring the demand is of paramount importance to facilitate station planning and enhance the effectiveness of the local bike-sharing networks. Therefore, this thesis focuses on predicting the station-level demand of a docked public bicycle-sharing system by applying deep learning (DL) models. The predictions made by the DL models, specifically the Spatial Regression Graph Convolutional Neural Network (SRGCNN) and the SRGCNN-Geographically Weighted, are compared with classical machine learning (ML) methods, including Multiple Linear Regression, Multilayer Perceptron (MLP) Regressor, Support Vector Machine (SVM), and Random Forest (RF) Regressor. The number of trips between the stations is utilized to facilitate the station demand prediction. The two DL models are implemented and evaluated. Initially, station demand is measured based on the number of start trips for each station. Exploratory variables based on the urban built environment are calculated as influencing factors for the prediction task. The DL and ML models utilize these calculated features to predict the station demand. The overall results indicate that in terms of RMSE, MAE, and R-squared, the DL models and RF Regressor exhibit better performance in predicting station-level demand than the rest of the ML models. This thesis contributes to monitoring and decision-making processes regarding public bicycle-sharing station-level planning by examining the different influencing factors based on the urban built environment and implementing Machine learning and deep learning methods to predict the station-level demand. Future research endeavors could improve the models' performance results and examine different factors affecting the PBS station-level planning.

Acknowledgments

I sincerely thank Pengxiang Zhao and Ali Mansourian for their guidance and supervision throughout my thesis. Their expertise and support were critical in completing this research endeavor. Without their mentorship, this accomplishment would not have been possible.

I also wish to convey my appreciation to my colleagues in the Physical Geography and Geomatics department. Our interactions during meetings offered a platform for enriching discussions, constructive feedback, and mutual encouragement. The collaborative spirit and camaraderie we fostered were invaluable to this academic journey.

Furthermore, I am profoundly thankful to the contributors behind scipy, sklearn, and torch. Their dedication to developing these libraries gave me the essential resources and tools to evaluate my datasets using various models and architectures comprehensively. Their contributions have played an indispensable role in advancing machine learning and data analysis.

I want to express my heartfelt gratitude to all those who contributed to my growth and progress throughout this thesis. Your support, insights, and guidance have been immeasurable, and I genuinely appreciate the knowledge and motivation I gained from each of you.

Table of Contents

List of Tables

List of Figures

Table 1. List of abbreviations

Abbreviations	
ML	Machine Learning
DL	Deep Learning
MLP	Multilayer Perceptron
RF	Random Forest
GCNN	Graph Convolutional Neural Network
SRGCNN	Spatial Regression Graph Convolutional Neural Network
SRGCNN-GW	Spatial Regression Graph Convolutional Neural Network – Geographically Weighted
PBS	Public Bicycle Sharing
CNN	Convolutional Neural Network
OSM	Open Street Map

1. Introduction

A city comprises different subsystems, such as industries, universities, and transportation, facilitating residents' everyday life and development (Zong et al., 2019). The transportation subsystem ensures citizens' safety, convenience, and efficient daily commute from one place to another (Xiao et al., 2020). Nonetheless, the rapid growth of urbanization in numerous big cities has led to several problems, such as pollution, noise, parking, accidents, and congestion (Xiao et al., 2019). Therefore, the citizens' travel needs are hard to fulfill, and their travel efficacy could be higher (Xiao et al., 2020).

Alternative ways of shared mobility (bike, e-bike, e-scooter, and car sharing) have been introduced in several cities around the globe (Si et al., 2019). Specifically, Public Bicycle-sharing systems (PBS) as a more sustainable transportation solution aim to reduce carbon emissions further and provide citizens with independence and a more convenient, flexible, and sustainable transportation system (Chen et al., 2021). PBS systems consist of Station-based/Docked and Dockless Public Bicycle-sharing systems (SD-PBS and DL-PBS, respectively).

PBS programs and their users have grown significantly in recent years. According to Xiao et al. (2020), 256 cities in China had implemented their PBS programs with more than 50 million users by July 2018. The business model, the network design, the proper selection of operation areas, and predicting the picking up and returning bike demand are contributing factors to a successful and sustainable PBS system (Zhang et al., 2015). Despite the advantages, several problems have occurred after their application. To mention a few, unintelligent and random placement of bicycle stations and aimless and inaccurate bicycle deployment make it hard to deal with road management and traffic safety (Chen et al., 2020; Ma et al., 2016). In addition, problematic bicycle distribution leads to the depletion of bicycle resources. Hence, appropriate bicycle station arrangements are vital to enhancing the usage efficiency of PBS and bicycle availability, allocation, and usage (Chen et al., 2021).

The PBS demand prediction can be implemented in three ways: cluster, area, and station-based demand prediction (Xiao et al., 2020). In cluster-based demand prediction, stations are separated into clusters with multiple clustering algorithms. Stations in the same clusters have similar demand characteristics, and those in different clusters have distinct demand characteristics. Contrastingly, area-based demand prediction aims at calculating the bike station demand in a particular area based on map segmentation. Studies show that deciding the study area size is hard since bike-sharing stations are unevenly distributed. Lastly, station-based demand prediction is applied at a smaller scale, which is vital to improving the selection of operation areas and user convenience.

This research utilizes the novel deep learning method Graph Convolutional Neural Network (GCNN) to accurately predict the PBS station-based demand, optimizing the existing bicycle station layout. GCNN, as a type of neural network working with graphs, can predict nodes,

edges, and network-based tasks. There have already been several studies in micro-mobility patterns utilizing machine learning techniques. However, the use of the GCNN method has been relatively limited in previous research endeavors; therefore, the findings of this study can lay the foundation for comparing and utilizing previous studies in future endeavors. Additionally, implementing the proposed GCNN method will facilitate public transport capacity mitigation and micro-mobility monitoring by city planners, policymakers, and service providers (Li et al., 2021).

1.1 Research gaps

Most studies have applied GCNN in various domains, such as traffic forecasting, taxi demand, and business practices, but in micro-mobility demand prediction they are still rare, especially when it comes to station planning. Moreover, influencing factors on demand prediction have been examined but not in conjunction with GCNN modeling. Therefore, this study aims to bridge the gap by exploring different variables that can impact long-term PBS demand prediction, focusing on the built environment. Additionally, graphs are constructed, and two GCNN models are developed to investigate the influencing factors of PBS demand and assist station planning.

1.2 Master thesis aim

This thesis aims to study the application of GCNN for predicting the station-based demand to assist station planning. Two GCNN models are applied to graphs to predict the bike-sharing station demand. Graphs are constructed based on the number of trips. The GCNN models and the different graph structures are scrutinized. This thesis has the following research objectives:

- Outline the existing influence factors related to bike-sharing demand.
- Define and implement two Graph Convolutional Neural networks to predict station-based demand.
- Compare the prediction performance of two GCNNs with traditional machine learning methods.

2. Background

2.1 Graph Convolutional Neural Networks

The evolution of smart cities and intelligent transportation systems led to integration of big data with machine learning into the micro-mobility system analysis (Chen, Chen, et al., 2020). Graph neural networks (GNN), used for nonlinear machine learning models, have recently drawn much attention. Graph structures, which can emulate the spatial and temporal relationships with the temporal and spatial convolutional network, can facilitate the station-level prediction and achieve better accuracy (Luo et al.,2021). The training of the model can be based on the graphs. Researchers tried to make the model less complex by introducing convolution into the GNN.

Graph Convolutional Neural Networks (GCNN) have been recently used in various applications, especially within the geospatial domain. That is because most of the data in the real world are non-Euclidean, are described by nodes with various characteristics, and therefore have underlying graph structures. The difference between GCNN and regular CNN lies in the data structure; CNN operates in Euclidean data structures, while GCNN operates in unordered nodes, connected in various ways and having different properties. In addition, graph CNN performs better when it comes to computational efficiency.

GCNNs have been applied in multi-scale traffic forecasting, which is the basis of urban traffic control and guidance. The traffic flow variables (namely speed, volume, and density) indicate the traffic condition status and contribute to future traffic predictions. Although statistical methods such as linear regression perform well on short-interval forecasts, they could be more effective in long-term predictions. That is because traffic flow is complex and uncertain. Machine learning methods, such as Support Vector Machine (SVM) and Neural Networks (NN), have challenged classic statistical models by providing higher prediction accuracy (Yu et al., 2017). Nonetheless, traffic networks are dense. Therefore, extracting spatial and temporal features jointly from the input is challenging.

Yu et al. (2017) introduced a different approach for extracting traffic flow's temporal dynamics and spatial dependencies. The traffic network was modeled as a graph to use spatial information. Furthermore, they apply multiple Spatio-temporal graph convolutional blocks, a combination of convolutional graph layers, and convolutional sequence layers to model spatial and temporal dependencies.

GCNNs have also been used to understand places' properties in geographic contexts. These properties depend on the characteristics of the place itself and its neighboring places. Hence, predicting these properties can be challenging as the relationship between the different places can be difficult to identify, and the neighborhoods can be described in various ways. Zhu et al. (2020) utilized GCNNs to model places as graphs. Nodes represented places, node features the place properties, and edges represented the place connections. Specifically, observed place characteristics and connections were used to predict unobserved place properties in Beijing.

When it comes to business practices, selecting store sites can be a challenging task. Identifying potential locations for future stores is critical for attracting new customers and increasing a company's profits. The classic frameworks used for that purpose give a false impression of the local properties, which are multi-dimensional and unstructured. A recent study suggests a convolutional graph network (GCN) for practical site identification tasks (Lan T. et al., 2022). The case study is implemented in Singapore. The datasets included land use information and public transport networks. By constructing a geospatial GCN based on this novel dataset, the level of attractiveness of various store sites within the neighborhood was predicted.

Another recent study examines taxi demand forecasting using a graph convolutional neural network (Xu et al., 2022). Two main challenges were tackled: virtual station discovery and modeling of the taxi demand forecasting. A two-stage clustering method was used to deal with

the former challenge. This algorithm considered the geospatial properties of taxi data and the transport properties of the demand between stations and revealed the number of stations. A non-negative fusion model (NGFM) was utilized for the last challenge, consisting of a non-negative matrix factorization and graph convolutional neural networks. The two-stage clustering method determined the number of stations, considering the taxi demand's geographic and load properties, and identified the potential virtual stations accurately and efficiently. Moreover, the NGFM extracted the node properties and estimated the associated adjacency matrix. This approach proved to achieve taxi demand predictions accurately.

Another study examined passenger flow prediction in urban rail transit (URT) management, which aims to reduce congestion. Real-time passenger flow forecasting can be challenging due to the URT network's development, topology, and spatiotemporal correlation (Wang et al., 2021). Previous research on this topic shows that two challenges are still unsolved: the correlation between stations and the identification of changing trends. Therefore, this study applies a multi-graph differential convolutional network (MDGCN) to extract the correlations between stations in heterogeneous space for URT passenger flow forecasting and capture the spatiotemporal correlations.

Overall, GCNNs have already been used in several applications for traffic forecasting, predicting unobserved place properties, potential locations for future stores, taxi demand prediction, and passenger flow prediction (Wang et al., 2021; Xu et al., 2022; Zhu et al., 2020). However, they have yet to be used for micro-mobility demand prediction exclusively, which could reveal the patterns and users' preferences over micro-mobility solutions and help policymakers and urban planners to design more sustainable and environmentally friendly cities.

2.2 Factors influencing Public Bike-sharing demand

There are various factors contributing to the PBS demand prediction. Several studies examined the impact of weather conditions, such as air temperature and precipitation, air quality, demographics, bike-sharing distribution of stations, built environment, transportation infrastructure, bike availability, and the promotion of free rides (Bao et al., 2018; Shen et al., 2018; Zhang et al., 2017). Regarding the weather-related influence, Shen et al. (2018) showed that although the climate is equatorial and, therefore, the weather would be relatively hot (around 30 degrees Celsius), temperature fluctuations during the day significantly impact the usage of bikes in Singapore. Specifically, the number of bike trips would rise during the late afternoon when the temperature would drop. In addition, the study showed that heavy rain could significantly reduce overall bike usage, especially during peak hours compared to off-peak hours, since the number of bike trips had doubled after the rain.

When it comes to the influence of the built environment, Shen et al. (2018) divided land use into four categories (namely public residential, private residential, commercial, and industrial areas). They collected points of interest (POI) from the Google Place API. The usage rate of

bikes in public residential areas was negative and relatively low due to the oversupply of public bikes in highly accessible areas. Contrastingly, the usage of public bikes in private residential areas was positive but insignificant because the supply of bikes is less than its actual demand. Then, the PBS usage rate was high in commercial areas and low in industrial areas. Transportation infrastructure can also have an impact on the PBS demand. Factors such as the availability of longer cycling routes, accessibility to bike stations, the number of available bikes, and road network connectivity can positively affect PBS usage. Also, it has been observed that promoting free bike rides leads to more bike trips (Shen et al., 2018).

Although many studies examined the influence of the built environment, land use, demographic, infrastructure, and weather on the PBS demand, the diversity of each station's characteristics needs to be addressed. Bao et al. (2018) focused on classifying the PBS stations into different categories by using POI data around each station and further exploring the factors contributing to the PBS demand in New York City based on the different station characteristics. According to the model specification results, employment, the number of college enrollments, population, and median household income significantly impact bike-share ridership. Regarding infrastructure, the higher number of bike racks, bike lane length, and density of bike lanes can generate more bike trips. Also, precipitation and snowfall influenced the PBS demand negatively, while good weather conditions and warm temperatures would lead to more trips.

Another study examined the built environment's impact on PBS demand at bike stations. According to Zhang et al. (2017), we can determine that there is a higher demand for bike stations situated in central urban areas with the highest population density. In addition, bike station capacity leads to a positive influence on the PBS station demand on weekends and holidays. This shows the users' preference for stations with larger capacity since the likelihood of finding an available bike is high. On the contrary, if the bike stations are relatively close (within 300 meters), the bike station demand shows a significant drop.

Moreover, PBS demand increases with more bike lanes within the 1000 meters buffer area of the stations and the more considerable length of branch roads, which accounts for users' preference to choose bike-friendly roads and roads that are more accessible to parks, commercial, and residential areas. Land use diversity has also shown a positive impact on the PBS demand. Lastly, this study pointed out the existence of spatial autocorrelation between close stations. This means that the PBS station demand is positively correlated with the neighboring PBS station demand (Faghih-Imani & Eluru, 2016).

2.3 Demand prediction on Docked Public Bike-sharing Systems

Urban transport has experienced a remarkable evolution in recent years, especially after the advent of micro-mobility services (Zhang & Song, 2022). These services include shared bikes, e-scooters, and e-bikes, facilitating residents' daily commutes in urban centers. They can also help mitigate traffic congestion and reduce fuel consumption and carbon footprint, and they benefit human well-being (Zhang et al., 2015; Zhu & Diao, 2019).

Zhang and Song (2022) developed a framework to observe the network evolution of the docked bike-sharing system in New York City. They aimed to detect the periodic patterns of the system using a Gaussian Mixture Model clustering approach and uncover the growth trajectories of the bike stations via exponential and logistic models. Their findings showed that the PBS demand has a seasonal pattern, with high demand from July to November. Also, most of the stations have experienced growth since the program was launched, especially in the Bay area of Manhattan. These findings can help operators maintain an efficient PBS program and offer sufficient bikes to meet the bike trip demands.

Luo et al. (2021) combined graph neural networks (GNN) and long short-term memory (LSTM) to demonstrate the temporal and spatial dependencies of the docked bike-sharing program of a city in Zhejiang Province and predict the pickup/return demands of 186 stations. In addition, more data storage was required since the number of shared bikes and links in a transportation network increases fast, so using an efficient demand forecast model was vital. Hence, they used local spectral graph convolution (LSGC) to reduce the computational power and enhance the availability of real-time demand prediction. The results indicate that spatial dependency should be considered in predicting the demand for a docked PBS system.

Graph Neural Network (GNN) has extensively been utilized in graph or network analysis since it is an efficient deep learning model (Scarselli et al., 2009). Specifically, Graph Convolutional Neural Network (GCNN) operates in unordered nodes, which are connected in various ways, and have different properties, which have been applied to bike-sharing studies. Chai et al. (2018) suggested the prediction accuracy of the bike flow at the station level by utilizing deep learning methods. They saw the bike-sharing system as a graph and proposed a multi-graph convolutional neural network for the bike flow prediction at the station level. Specifically, they combined multiple graphs and then applied the convolutional layers to predict station-level future bike flow in New York and Chicago. The results showed that this multi-graph model performed better than other models. The prediction error declined by approximately 25% and 15% for NY and Chicago, respectively.

Xiao et al. (2021) used multi-view data and developed a spatiotemporal graph convolutional network to predict the public bike sharing (PBS) demand of 186 stations in Wenling, China. Specifically, they highlighted the superior performance of the GCNN method in terms of prediction accuracy and computational efficiency compared to the recurrent neural network (RRN) based models. By applying the spatiotemporal graph convolutional network (STGCN) model, it is denoted that PBS demand is greatly linked to the demand of the surrounding stations and the importance of spatial and temporal dependencies in predicting it.

3.Methodology

3.1 Study area and data

3.1.1 Data collection

The study area for this thesis is Zurich (Figure 1), the largest city in Switzerland, located in the north-central part of the country, with a population of 434,335. The PBS trip dataset is obtained from open-accessible APIs of service provider companies in Switzerland between May 27 and July 7, 2022. Each record corresponds to a trip, including the following attributes: id, start/end time, start/end longitude, start/end latitude, start/end station id, type, distance, and duration. Since this thesis examines only the docked PBS system, users can pick up a bike from one station, and they must return the bike to the same or another station. The PBS station dataset is obtained from Swagger API (Table 2). This dataset contains all the active PBS stations throughout Zurich (Figure 1), including attributes such as id, latitude, and longitude.

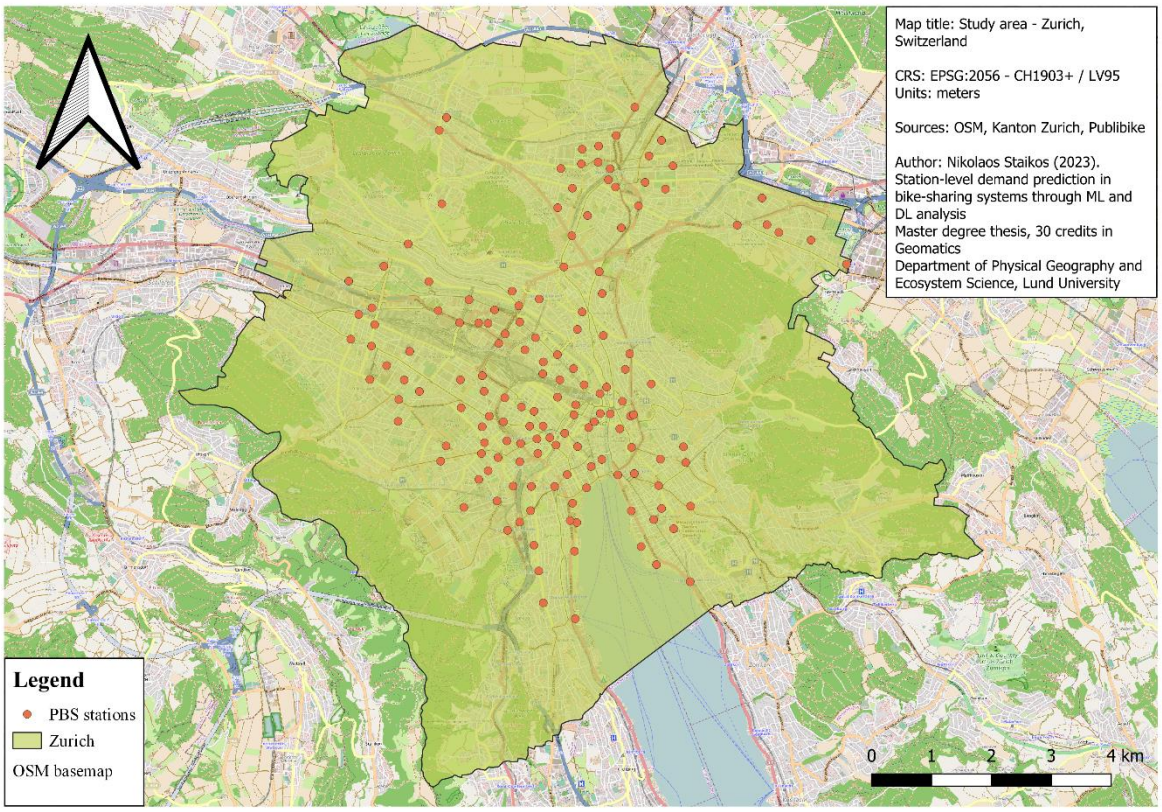


Figure 1. Map showing the study area: city of Zurich in Switzerland. The base map is obtained from Open Street Map. Also, the public bicycle sharing stations can be seen on the map, which are obtained from Publibike (Swagger API).

Table 2. List of datasets used for the analysis, their categories, and the sources.

Dataset	Category	Source
Land use	Land use classes	Open Street Map
POI	Points of interest	
Population density	Statistics	Kanton Zurich
Employment density	Statistics	
Municipality borders	Administrative regions	
Bike sharing stations	Transport	Swagger, Publibike API
Bike sharing trips	Transport	Lund University

The table in Appendic D shows the 16 variables used to train the model at a later stage and examine their impact on the PBS demand prediction and assist station planning. These variables are extracted from datasets downloaded from official geoportals such as Open Street Maps (OSM), Swagger API, the Kanton Zurich website, and Lund University.

3.1.2 Data processing

The datasets are processed with QGIS and Python. The bike-sharing stations dataset is called from Swagger API and saved into a CSV file format to be imported to QGIS for analysis. The stations kept for this thesis are only the ones within the city of Zurich. Then, 300-meter buffer zones are created around each station to examine further which variables fall into the zones and, subsequently, impact the PBS station-level demand (Jaber et al., 2022; Snehanshu et al., 2020).

Open Street Map datasets are used to examine the built environment characteristics around the bike-sharing stations. Industrial, commercial, and residential regions are extracted from the land use dataset. Moreover, the Points of Interest (POIs) dataset included numerous classes such as cafeterias, hospitals, bars, pharmacies, and more. To further differentiate the influencing factors of the PBS demand, the POIs are divided into ten variables: education, tourism, healthcare, entertainment, sports, accommodation, errands, dining, public transport stops, and bike lanes. In addition, the road dataset includes all the road networks within Zurich. Cycle paths are extracted from the road datasets by excluding the road classes inappropriate for cycling, such as highways.

Furthermore, population density and employment density show the population and the employees per 100mx100m, respectively. The public transport stops variable contains all the stops regarding the means of public transport such as buses, trains, and trams.

The following data analysis stage is to calculate the variables shown in Table 3. All the variables are overlayed with the 300-meter buffer zones of the bike-sharing stations to determine the influencing factors on the PBS demand around each station. The sum of each variable within the buffer zones is computed. For example, for the population density, the

number of people living within each buffer zone is calculated. The same process is applied for employment density, public transport stops, and POIs. For the bike lane density, the total length of the bike lanes within each buffer zone is computed. Moreover, the Euclidean distance is estimated from each bike-sharing station to the closest public transport stop.

A different method is applied for the land use variable. Land Use Mix is used in many urban and spatial planning applications, indicating the diversity of land use types within a study area. For instance, land use mixture can boost the promotion of walkability in urban areas (Mavoa et al., 2018) and is supposed to be energy-efficient (Zhang & Zhao, 2017). As mentioned above, residential, commercial, and industrial land use types are extracted from the land use dataset. Therefore, the land use mixture is estimated for these three land use types for each buffer zone, i.e., each bike-sharing station zone. The entropy index is the most used metric to quantify land use mixture (Song et al., 2013):

$$\text{Entropy} = -\frac{\sum_{j=1}^{k} P^j \ln P^j}{\ln k}$$

(1) Landuse Entropy Index

Where:

- P^j is the ratio of each land use type j in the buffer zone, and
- k is the total number of land use types.

The ratio P^j is computed in terms of the area of each buffer zone.

3.2 Overall Research Framework

Data collection is the first step of the analysis (Figure 1). The PBS data collected for this research are obtained from a micro-mobility services operator, as done by Li et al. (2021). This study will utilize docked bike-sharing trip data. The PBS station data are collected from Swagger API, including a unique ID for each station, latitude, longitude, and state of each station. Furthermore, more datasets, such as population density, employment density, POIs, and land use, are obtained from Open Street Maps API and Kanton Zurich's open data website. These datasets are used as the influencing factors on the PBS demand to assist station planning.

After collecting the data, preprocessing and filtering are implemented. The start of each trip is used to calculate the station-level demand. The PBS station data are used to show all the PBS stations in Zurich and thereafter create 300-meter buffer zones to calculate the influencing factors of each PBS station-level demand. Additional datasets are processed to calculate the influencing factors. A final geojson file is created, which contains the station ID, station-level demand, and the sixteen influencing factors of each PBS station.

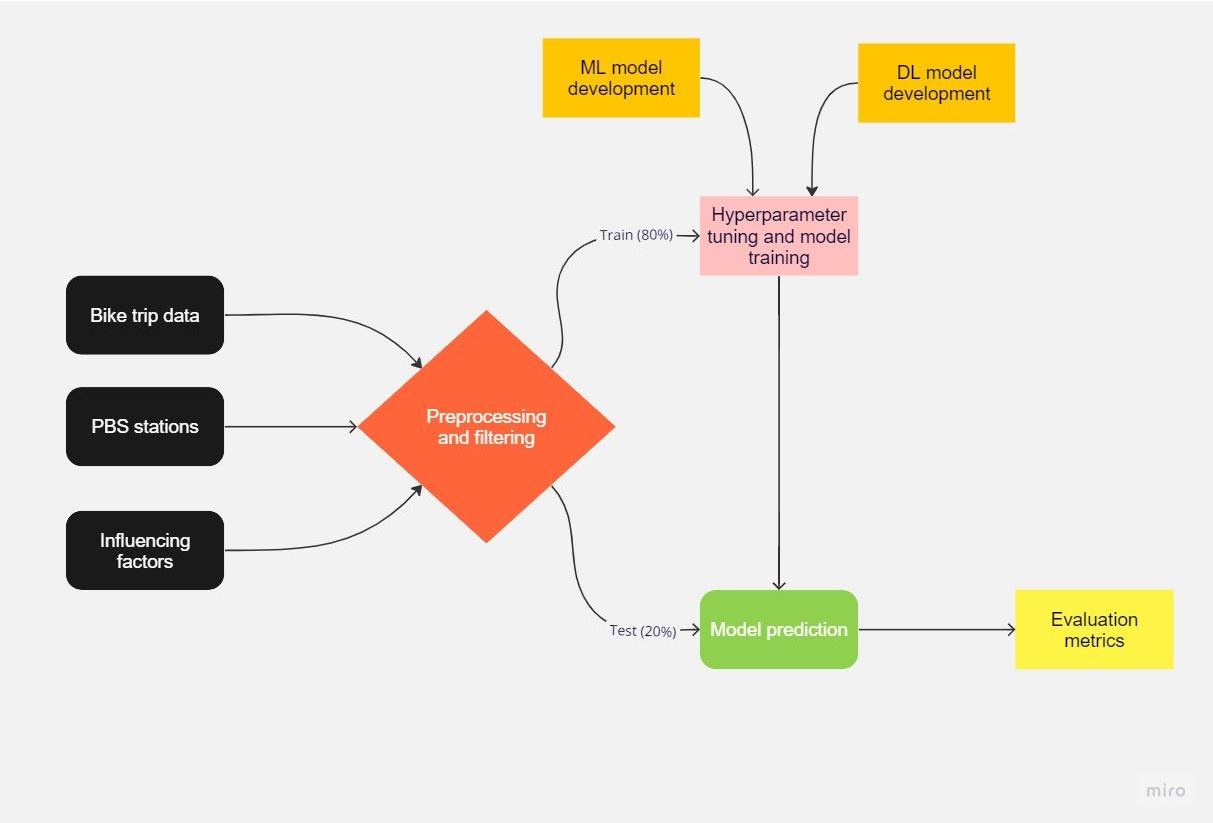


Figure 2. Step by step overview of the research framework. Data collection, preprocessing and filtering, data splitting, model development, and model evaluation.

The next step is splitting the final dataset into training and validation subsets, 80% and 20% respectively. After developing the four ML and two DL models, the training subset is used to train each model, so that it thereafter yields the PBS station-level demand predictions. Then, having the PBS station demand prediction model, the validation subset is used to validate the models' performance by utilizing evaluation metrics. Depending on the results, hyperparameter tuning is used to improve the models' performance.

A summary of the process described above and illustrated in Figure 2 is the following:

1. Data collection:
    - Collect docked PBS trip data and PBS station data for Zurich.
    - Collect additional data, such as population density, POIs, and land use, utilized as influencing factors for the PBS station-level demand prediction.
2. Preprocessing and filtering:
    - Use trip start times to calculate the station-level demand.
    - Create 300-meter buffer zones to calculate the independent variables for each PBS station.
    - Calculate variables such as hub distance, and land use entropy.

- Create a geojson file with station id, demand, and the sixteen influencing factors.

3. Data splitting into training (80%) and validation (20%) subsets
4. Model development:
    - Four ML and two DL models developed.
    - Training subset used to train each model for the PBS station-level demand prediction.
    - Tune hyperparameters to enhance models' performance.
5. Model validation:
    - Apply PBS station-level demand prediction model to the validation subset.
    - Evaluate model performance with evaluation metrics.

3.3 Machine learning model development

The datasets used in this master thesis are processed with four machine learning methods: Multiple Linear Regression (MLR), Multilayer Perceptron (MLP) Regressor, Support Vector Machine (SVM), and Random Forest (RF) Regressor, to compare with the performance of the GCNN model. The sklearn library was used to implement these models. Multiple Linear Regression is a classic and interpretable model that works well when there is a linear relationship between the independent and the dependent variables. MLP regressor, as a neural network, is a powerful tool for capturing complex and non-linear patterns in the data. SVM is well-suited for regression tasks and works well when predicting a continuous variable. RF regressor can capture complex relationships and provide reliable predictions. These models were chosen to provide a diverse set of ML models that cover different aspects of the data.

3.3.1 Multiple Linear Regression

Multiple Linear Regression is a fundamental machine learning model for predicting continuous numeric values based on multiple input features (Hasan et al., 2022). It is similar to the concept of Simple Linear Regression with the difference that Multiple Linear Regression incorporates multiple independent variables, influencing the dependent variable (the one to be predicted). The model assumes a linear relationship between the input features and the output variable. Coefficients are estimated for each input feature, which represent the strength and the direction of the relationship between that feature and the predicted value. Therefore, Multiple Linear Regression aims to find the best-fitting linear equation that minimizes the difference between the predicted and actual values of the output variable.

3.3.2 MLP regressor

The Multilayer Perceptron is one of the most widely known and used kinds of neural networks (Popescu et al., 2009). It consists of multiple layers of interconnected nodes, and organized feedforward. It is proven to be a fundamental architecture in deep learning, and adequate for several machine learning tasks such as classification regression, and pattern recognition.

Similarly with Random Forest Regressor, MLP Regressor is used exclusively for regression tasks, specifically for predicting continuous numerical values. The best hyperparameters found are shown in Appendix E (Supplementary table 2).

3.3.3 Support Vector Machine

Support Vector Machine (SVM) is a powerful and versatile machine learning model for classification and regression tasks (Evgeniou & Pontil, 2001). SVM aims to find the hyperplane that best separates data points belonging to different classes in a high-dimensional space. The input data are represented as vectors. Then, an appropriate kernel function is chosen for non-linear classification, such as linear, polynomial, radial basis function (RBF), and sigmoid kernels. During the model training, the SVM tries to find the decision boundary that maximizes the margin between the different classes, in order to determine the support vectors and the weights of the decision boundary. Finally, new data points are classified based on the side of the decision boundary they fall into. The best hyperparameters are shown in Appendix E (Supplementary table 3).

3.3.4 Random Forest Regressor

Random Forest is an ensemble of decision tree predictors, where each tree's predictions are influenced by a randomly selected set of values (Breinman, 2001). These values are sampled independently and come from the same distribution for all the trees in the forest. As the number of trees increases, the generalization error for the forests converges to a specific limit. The resilience of the individual trees and the correlation between them determines how effective a forest of tree predictors is. The final output is generated by the ensemble mean of all the predictions from each tree.

The individual trees' performance and their correlations significantly impact the accuracy of a forest of tree classifiers. Random feature selection offers improved strength against noise in the data. Furthermore, evaluating factors such as error, strength, and correlation offer valuable insights into how the forest responds to increasing the number of features used for splitting nodes. Also, the importance of the different variables used in the model can be found by accessing these factors, facilitating more accurate predictions.

Random Forest Regressor is a specific implementation of the Random Forest algorithm designed for regression tasks. Therefore, the main difference between Random Forest and Random Forest Regressor lies in the target variables and the intended use cases. Random Forest can handle both classification and regression problems, while Random Forest Regressor focuses on predicting numeric values. The best hyperparameters are shown in Appendix E (Supplementary table 4).

3.4 SRGCNN development workflow and architecture

Figure 3 illustrates the methodology for a spatial regression analysis, as described by Zhu et al. (2022). A spatial weights adjacency matrix W is constructed for all the stations that have spatial features including both the independent variables X (influencing factors) and the dependent variable y (PBS station-level demand). Then, the following generalized formula is used to specify the spatial regression model:

$$y = f_{\Theta}(y, W, X)$$

(2) Spatial Regression Model

For the fitted model, **Θ** is estimated, which includes all spatial and non-spatial effects when the model is fitted to the observations. Finally, the fitted model is used to predict the PBS station demand. In this master thesis, GCNNs are combined with traditional spatial regression and geographically weighted spatial regression.

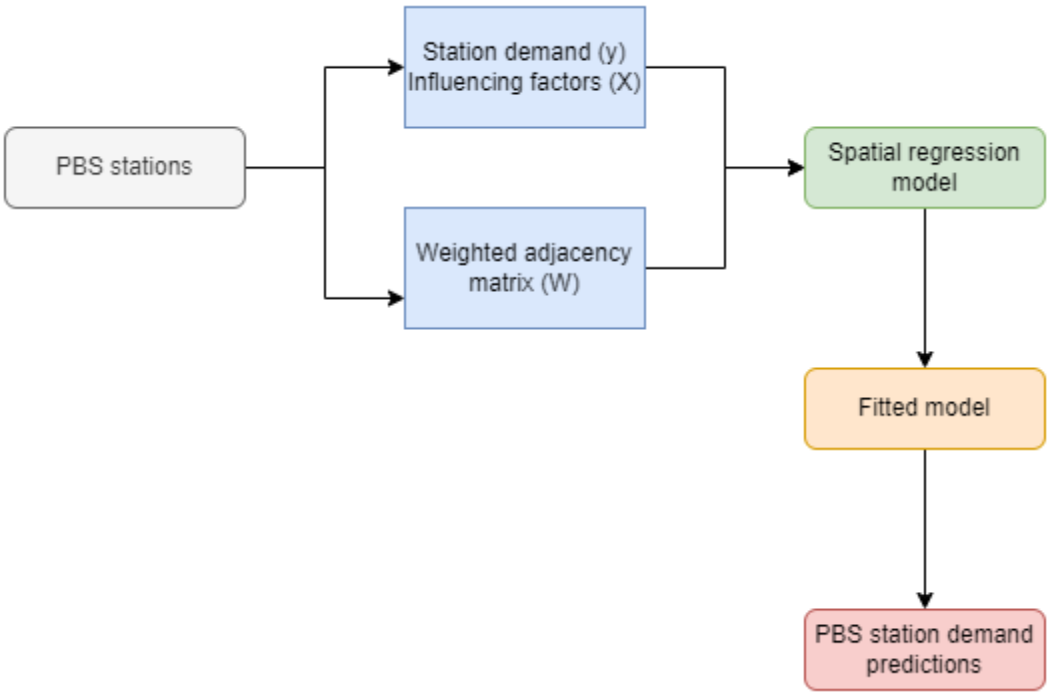


Figure 3. Illustration of a typical workflow for the spatial regression analysis. Four major steps are marked as 1) collecting the cross-sectional data and constructing the weighted adjacency matrix, 2) specifying a regression model, 3) estimating parameters in the fitted model, 4) predicting values for the dependent variable.

3.4.1 Graph structure

Graph Convolutional Neural Networks (GCNNs) are neural networks that operate on graph structure. Graphs consist of nodes and edges. By conceptualizing graphs, nodes are the network

objects, and edges can be any relation the nodes can have. Moreover, nodes and edges can have additional attributes. In this thesis, nodes demonstrate the PBS stations and edges the number of trips done between the stations. Also, the stations have additional features. Specifically, the independent variables analyzed in the previous section are the influencing factors of the PBS station-level demand prediction.

The figure below shows a directed graph representation comprised of four nodes (the blue circles) and five edges (the black arrows). Graphs can be directed or undirected. Directed graphs (Figure 4) demonstrate asymmetric relationships. Supposing that the nodes and the edges demonstrate the PBS stations and the number of trips respectively, there are trips only from station 361 to station 425. On the contrary, undirected graphs (Figure 5) have bidirectional edges. Therefore, there could be trips from station 361 to station 425 and from station 425 to station 361.

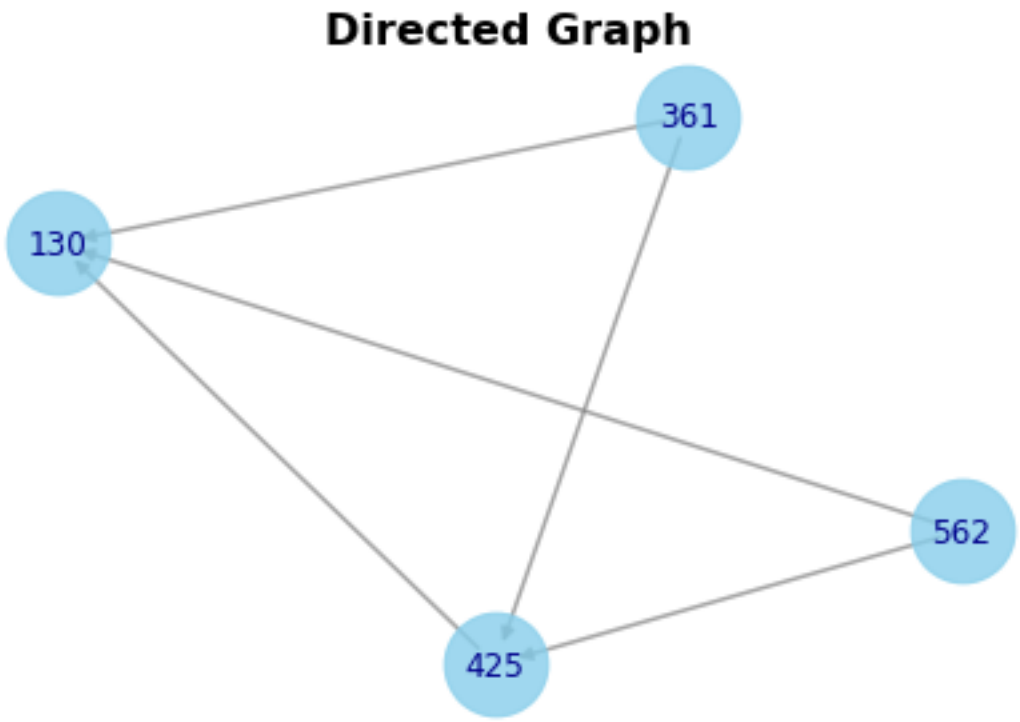


Figure 4: Directed graph representation. Four PBS stations are randomly selected.

Graph neural networks (GNNs) can be seen as a version of Convolutional neural networks (CNNs) that can be applied to graphs. CNN utilizes the convolution operation to compute, for instance, the hidden attributes for different pixels based on the neighboring pixel values, by using a kernel filter over the input image. Supposing that the nodes and the edges of a graph structure are the pixels and the distance of the adjacent pixels (resolution) respectively, then we have the Graph Convolutional Neural Networks (GCNNs).

GNNs are useful since we can use non-Euclidean spatial data. Contrary to the image representation by pixels, graph structures can represent reality as a set of unordered nodes and edges. In fact, this lays the foundation for numerous advanced applications that could be divided into three categories: graph, edge, and node classification.

Supposing that we have an undirected graph (Figure 5), therefore containing nodes with symmetric relationship, with attribute values attached to the nodes and the edges. In this thesis, the attribute values attached to the nodes and the edges are the influencing factors and the number of trips respectively.

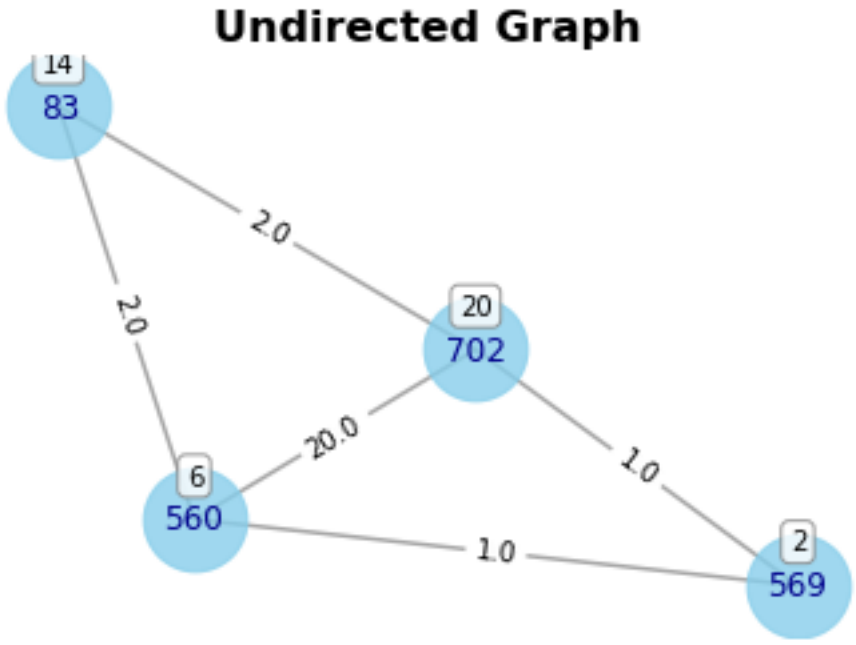


Figure 5: Undirected graph representation. Four PBS stations are randomly selected and plotted using Python. Each station is represented as a node and is denoted by a station ID (located at the circle's center). A single attribute is associated with each station, which corresponds to the count of public transport stops within the 300-meter buffer zone. In cases where trips exist between the stations, edges are established to connect the nodes. The edges are additionally annotated with numeric values at their centers, representing the number of trips between the linked stations.

Figure 5 is expressed by the adjacency matrix and the feature matrix. The adjacency matrix (denoted by A) indicates which nodes are connected or not in the graph. Moreover, this matrix is symmetrical with zero diagonal (Equation 3). The feature matrix includes (denoted by H) the attributes of the nodes. In this case, the adjacency matrix (A) demonstrates which PBS stations are connected or not and the number of trips between them, and the feature matrix (H) illustrates the influencing factors of each PBS station (in this example only the number of public transport stations in the vicinity of each PBS station are illustrated).

$$A = \begin{matrix} 83 \\ 560 \\ 569 \\ 702 \end{matrix}\begin{bmatrix} 0 & 2 & 0 & 2 \\ 2 & 0 & 1 & 20 \\ 0 & 1 & 0 & 1 \\ 2 & 20 & 1 & 0 \end{bmatrix}; H = \begin{matrix} 83 \\ 560 \\ 569 \\ 702 \end{matrix}\begin{bmatrix} 14 \\ 6 \\ 2 \\ 20 \end{bmatrix}$$

(3) Examples of Adjacency and Feature matrices.

Then, a new feature vector is generated to reveal the attributes of the neighboring stations for each station based on the connection of those neighboring stations. When estimating the attributes of the neighboring stations, the attributes of the particular station being examined are excluded, which is invalid. Thereafter, the identity matrix (I) is created and added to the adjacency matrix (A) in order to alter the zeros to ones. The modified adjacency matrix will be:

$$\tilde{A} = A + I$$

(4) Modified adjacency matrix.

Furthermore, the values of the feature matrix are relatively high which may lead to computational instability. One way to tackle this issue is to normalize the feature matrix by dividing it by the graph degree matrix (D). The degree matrix includes information about the number of trips corresponding to each station. Therefore, the final normalized feature vector is estimated by the following formula:

$$H' = D^{-1}\tilde{A}H$$

(5) Normalized feature vector.

In this thesis, two graph structures are utilized. The first graph structure, used for the SRGCNN model, is based on the weighted adjacency matrix, built upon the number of trips between the PBS stations in Zurich as mentioned above. The second graph structure is based on the k nearest neighbors (KNN), inspired by Zhu et al., 2022. KNN considers up to the k^{th} nearest neighbors regarding each PBS station. Supposing that the number of GCNN layers is 2 and the sampling ratio of training is 80%, the smallest number of k should satisfy $k^2 \geq 80$, so that the model covers all the stations after the self-multiplication of weights matrix when the training features are propagated in the GCNN. So, the required k needs to be at least 9. To make sure that all stations are covered and after hyperparameter tuning, the k value was set to 10 and 35. It should be mentioned that the graph structures and, subsequently, the definition of the spatial weights matrix, significantly affect the regression analysis. Hence, different graph structures could lead to better model fitting and higher prediction accuracy (Zhu et al., 2020).

3.4.2 SRGCNN and SRGCNN-GW

Since GCNNs can be applied to spatial analysis, in this master thesis Spatial Regression Convolutional Neural Networks (SRGCNN) is used to conduct spatial regression and predict

the demand of the PBS stations in Zurich. SRGCNN facilitates the understanding of graph convolution mechanisms and spatial regression models (Zhu et al., 2022).

The construction of SRGCNNs begins by gathering data that includes the PBS station locations and cross-sectional information, i.e., the PBS demand as the depend variable y and the influencing factors as the independent variables X. A graph is then created to represent the adjacency matrix. The nodes on the graph are initialized with the known values of the influencing factors. These values are then propagated through a specific GCNN architecture. The values of the PBS demand, which are available at the training nodes, are input to the last GCNN layer to calculate the output errors and enable backpropagation. SRGCNNs optimize the parameters of its GCNN model using a semi-supervised learning strategy which considers the weights between all the given nodes even though the demand values are only observed at the training nodes. By leveraging the propagation mechanism and the spatial locality nature of GCNNs, SRGCNNs optimize the GCNN model to approximate spatial relationships accurately and predict the PBS station-level demand values.

Spatial regression models do not necessarily allow the parameters to change spatially (Zhu et al., 2022). The SRGCNN is a global learning model. It utilizes shared parameters to distinguish the stationary knowledge of geographical relationships and does not take into account the heterogeneity between locations. On the contrary, local models such as SRGCNN-GW consider the fact that spatial relationships can vary between different spatial locations. This model uses a spatial weight matrix (W) and additional parameters at each location. Hence, SRGCNN-GW approach enables a learning process that adapts to the unique characteristics of different geographical areas. The spatial weights matrix is built based on k nearest neighbor inspired by Zhu et al., 2022.

Supposing we have N nodes, C_{in} input features and C_{out} output features, the SRGCNN can be described by the following layer-wise graph convolution:

$$X^{(l+1)} = W \times X^{l} \times \Theta^{l}$$

(6) SRGCNN formula

Where:

- $X^{(l+1)} \in \mathbb{R}^{N \times C_{out}}$
- $X^{(l)} \in \mathbb{R}^{N \times C_{in}}$
- $W \in \mathbb{R}^{N \times N}$
- $\Theta^{l} \in \mathbb{R}^{C_{in} \times C_{out}}$

The parameters Θ^{l} are common to all neurons within the GCNN, resulting in a global spatial regression across a locally connected graph structure. On the other hand, the SRGCNN-GW model, with a geographically weighted layer-wise graph convolution will be as follows:

$$X^{(l+1)} = W \times \left(X^l \otimes \Theta_{local}^{(l)}\right) \times \Theta^l$$

(7) SRGCNN-GW formula

Where $\Theta_{local}^{(l)} \in \mathbb{R}^{N \times C_{in}}$ and contains the geographically weighted parameters $X^l \otimes \Theta_{local}^{(l)}$. Therefore, the features are parameterized, with each node having an independent set of trainable parameters.

The hyperparameters used for the SRGCNN and SRGCNN-GW are shown in Appendix E (Supplementary table 5 and 6).

3.5 Machine Learning and Deep Learning techniques

Some of the techniques used to build and train the ML and DL models are mentioned below.

Overfitting and underfitting

When a model learns the training data it might capture noise and fluctuations rather than the underlying patterns, which is primarily the goal of applying ML and DL models (Bashir et al., 2020). This is called overfitting and it results in poor model performance and inaccurate predictions. Therefore, an overfit model fits the training data very well; however, its performance on the validation data is worse. On the other hand, a simplistic model fails to learn from the training data and, subsequently, performs poorly on both the training and the test data. This is called underfitting. Underfit models have usually high bias and low variance.

Epoch

A complete pass through the entire training dataset during the model training is called epoch. Especially in DL models, as done in this thesis, multiple epochs are used for model training in order to enhance the model's ability to learn complex patterns and achieve better results.

Hyperparameter tuning (Random search)

The hyperparameters are defined by the user before training the model. Each ML and DL model have their own set of hyperparameters. The process of finding the best combination of hyperparameters in order to achieve better model performance is called hyperparameter tuning. In this thesis, random search (Bergstra & Bengio, 2012) was applied to determine the best combination of hyperparameters for the ML models. For the DL models, different hyperparameters have been applied after experimentation.

Regarding random search, the hyperparameters and their range of values is predefined. After a specified number of iterations, random search randomly selects a set of hyperparameters and yields the best performance on the validation data. It is a computationally efficient hyperparameter optimization technique since it allows a wider exploration of the hyperparameter space. Nonetheless, it is not guaranteed that random search results in the best possible hyperparameter combination. Hence, most of the times the process needs to be done multiple times with different hyperparameters and ranges of values to enhance the models'

performance results by finding the best set of hyperparameters. In this thesis, random search is utilized to determine the hyperparameters for MLP regressor, SVM, and RF regressor.

The steps of random search are the following:

1. Define the hyperparameters to be tuned and their potential values.
2. Define the number of iterations, i.e., the number of hyperparameter combinations to be randomly selected and assessed.
3. Randomly select a set of hyperparameters for each iteration.
4. Model training using the selected set of hyperparameters on the training data.
5. Performance evaluation on the validation set.
6. Record the model's performance evaluation metrics for the selected set of hyperpararameters.
7. Repeat steps 3-6 for the specified number of iterations.
8. Select the hyperparameters that led to the best performance based on the evaluation metrics.

Activation functions

Activation functions allow neural networks to learn complex rationships in the data by introducing non-linearity. Some common functions are ReLU(Rectified Linear Unit), sigmoid, and tanh.

Dropout

Dropout is a technique utilized in neural networks to prevent overfitting. During model training, some neurons are randomly ignored which improves model performance and robustness.

3.6 Experimental settings

3.6.1 Settings for machine learning methods

The following settings were used to process the data for the machine learning methods:

- The entire dataset is split into training and test samples. Specifically, 80% is used for training, and 20% is used for testing. The same training and test samples are used for all the ML models in order to compare the performance evaluation results.
- Choose the best hyperparameters to improve the model performance.
- The entire training sample is used to fit each ML model.
- The entire test sample is used to evaluate each ML model's performance.
- Performance metrics are estimated for each ML model such as MSE, RMSE, MAE, MAPE, and R-squared.

3.6.2 Settings for DL models

The following settings were used to process the data for the DL models:

- The entire dataset is split into training and test samples. Specifically, 80% is used for training, and 20% is used for testing. The same training and test samples are used for all the ML and DL models in order to compare the performance evaluation results.
- Calculate the spatial variables for each PBS station in Zurich. The spatial variables are calculated in QGIS. 300-meter buffer zones are created for each PBS station and each variable is estimated for each buffer zone.
- Construct graph structures in two ways to model the spatial dependency effects among the PBS stations. The first graph structure, used for SRGCNN, is constructed based on the trips occurring between two stations. The second graph structure, used for SRGCNN-GW, is built based on k nearest neighbor.
- Calculate the demand for each PBS station. The trip dataset is used to calculate the demand by summing up the number of trips for each PBS station. The start of each trip is used to calculate the demand.
- Train and test each GCNN model with the same datasets used for the ML models in order to compare the performance metrics.
- Estimate the performance metrics such as MSE, RMSE, MAE, MAPE, and R-squared.

3.7 Evaluation metrics

Several metrics are used to evaluate the performance of the ML and the GCNN models (Chicco et al., 2021). These metrics are namely Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-squared. One adjacency matrix is used for the GCNN models: the weighted adjacency matrix. The weights of adjacency matrix are the number of trips between the PBS stations.

Mean Square Error (MSE)

MSE is one of the most common metrics which calculates the average squared difference between the observed and the predicted values of the dependent variables in a dataset, the PBS demand in this case. The formula for calculating MSE is the following:

$$MSE = \left(\frac{1}{n}\right) * \sum (y_i - \hat{y}_i)^2$$

(8) MSE

Where:

- n is the total number of the observations
- y_i is the observed value for the i-th observation
- $\hat{y}_i$ is the predicted value for the i-th observation

By calculating MSE how well a model's predictions match the actual observed values can be shown. A low MSE value suggests that the model's predictions are closer to the actual values; therefore, the model performs well. One of its limitations is that due to the squaring operation,

if outliers or extreme values are present in the dataset, MSE can be relatively high. However, MSE is widely used in model training and optimization processes due to its ease of interpretation, when the aim is to detect model parameters that minimize the MSE and enhance the model's overall accuracy.

Root Mean Square Error (RMSE)

RMSE is derived from MSE and calculates the average magnitude of the prediction errors in the same units as the original observations. RMSE is estimated from the following formula:

$$RMSE = \sqrt{MSE}$$

(9) RMSE

The difference between RMSE and MSE lies in the scale of the metric. MSE value is in squared units, whereas RMSE value is the original units of the data which facilitates interpretation and comparison against the original observations.

Mean Absolute Error (MAE)

MAE estimates the average magnitude of errors between the observed and the predicted values in a dataset. Similar to MSE and RMSE, MAE is utilized to evaluate the model performance, but it focuses on the absolute differences between actual and prediction values. The formula for calculating MAE is the following:

$$MAE = \left(\frac{1}{n}\right) * \sum |y_i - \hat{y}_i|$$

(10) MAE

Where:

- n is the total number of the observations
- y_i is the observed value for the i-th observation
- $\hat{y}_i$ is the predicted value for the i-th observation

In contrast to MSE and RMSE, where the differences are squared and thereby larger errors are more heavily penalized, with MAE all errors are equally treated. Hence, MAE is a more interpretable metric of the model performance and, subsequently, the prediction accuracy. It should be noted that MAE is particularly useful when outliers or extreme values are present in the dataset, since it is less sensitive compared to squared error metric such as MSE.

Mean Absolute Percentage Error (MAPE)

MAPE considers the average percentage differences between the observed and the predicted values. It is useful especially when the goal is to access the relative accuracy of predictions, especially when dealing with data of varying scales. The formula for calculating MAPE is:

$$MAPE = \left(\frac{1}{n}\right) * \sum (|y_i - \hat{y}_i|/y_i) * 100$$

(11) MAPE

Where:

- n is the total number of the observations
- y_i is the observed value for the i-th observation
- $\hat{y}_i$ is the predicted value for the i-th observation

So, the error is expressed as a percentage. One drawback of MAPE is its sensitivity to cases where the observed values are close to zero, which can lead to extremely large percentage errors.

R-squared

R-squared or the coefficient of determination evaluates the goodness-of-fit of a regression model. It accounts for how well the independent variables in a regression model explain the variability of the dependent variable. The formula for calculating R-squared is:

$$R^2 = 1 - (SSR/SST)$$

(12) R-squared

Where:

- Sum of Squared Residuals (SSR) is the sum of the squared differences between the actual values of the dependent variable and the predicted values by the regression model, and

$$SSR = \sum (y_i - \hat{y}_i)^2$$

(13) SSR

- Total Sum of Squares (SST) is the sum of the squared differences between the actual values of the dependent variable and the mean of the dependent variable.

$$SST = \sum (y_i - \bar{y})^2$$

(14) SST

The values of R^2 can range from zero to one. If $R^2 = 0$ the model fails to explain any variability in the dependent variable. Contrastingly, if $R^2 \approx 1$ the model explains a high proportion of the variability in the dependent variable.

Nonetheless, it should be mentioned that if the R^2 value turns out to be high (close to one) it cannot be assumed that the model is a good fit or that the predicted values are accurate. For example, by adding more independent variables to the model the R^2 tends to get higher. This leads to overfitting where the model performs well on the training data but shows poor

performance on the predicted values. On the contrary, is R^2 is close to zero it might indicate that the model is relatively simple and, therefore, fails to capture underlying patterns. However, it does not necessarily mean that the model is unreliable. Moreover, a low R^2 value could be valuable in case the model offers significant insights. The interpretation of R^2 highly relies on the context of the data and the aim of the analysis. Therefore, alternative factors should be considered when evaluating the quality and the accuracy of a regression model such as the selection of the independent variables, assumption been made, and the context of the data.

4. Results

4.1. Maps of the independent and dependent variables

The PBS demand is the dependent variable. It is used to train and test the different models implemented at a later stage of this thesis. Figure 6 shows the distribution of the PBS demand against the frequency within the dataset. The demand for most of the PBS stations is around 500. There are some PBS stations with greater demand, approximately 1500-2000, and 1 PBS station with demand value close to 3500. The mean and the standard deviation of the PBS demand are 497 and 321 respectively.

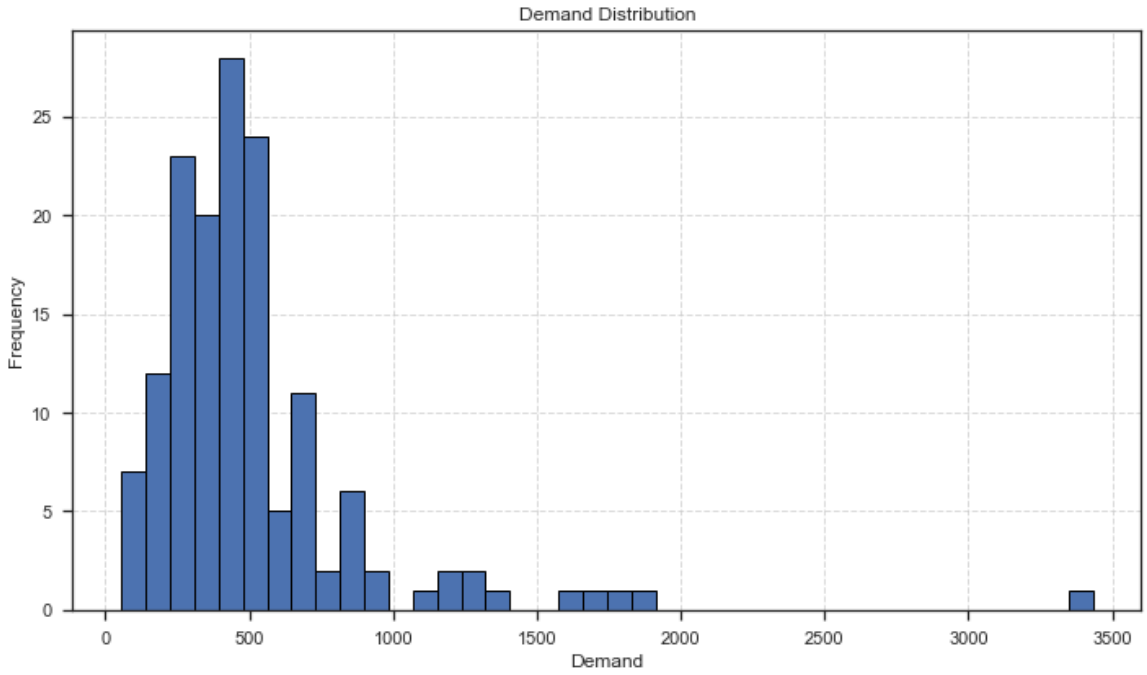


Figure 6. Demand distribution of the Public Bike-Sharing stations in Zurich.

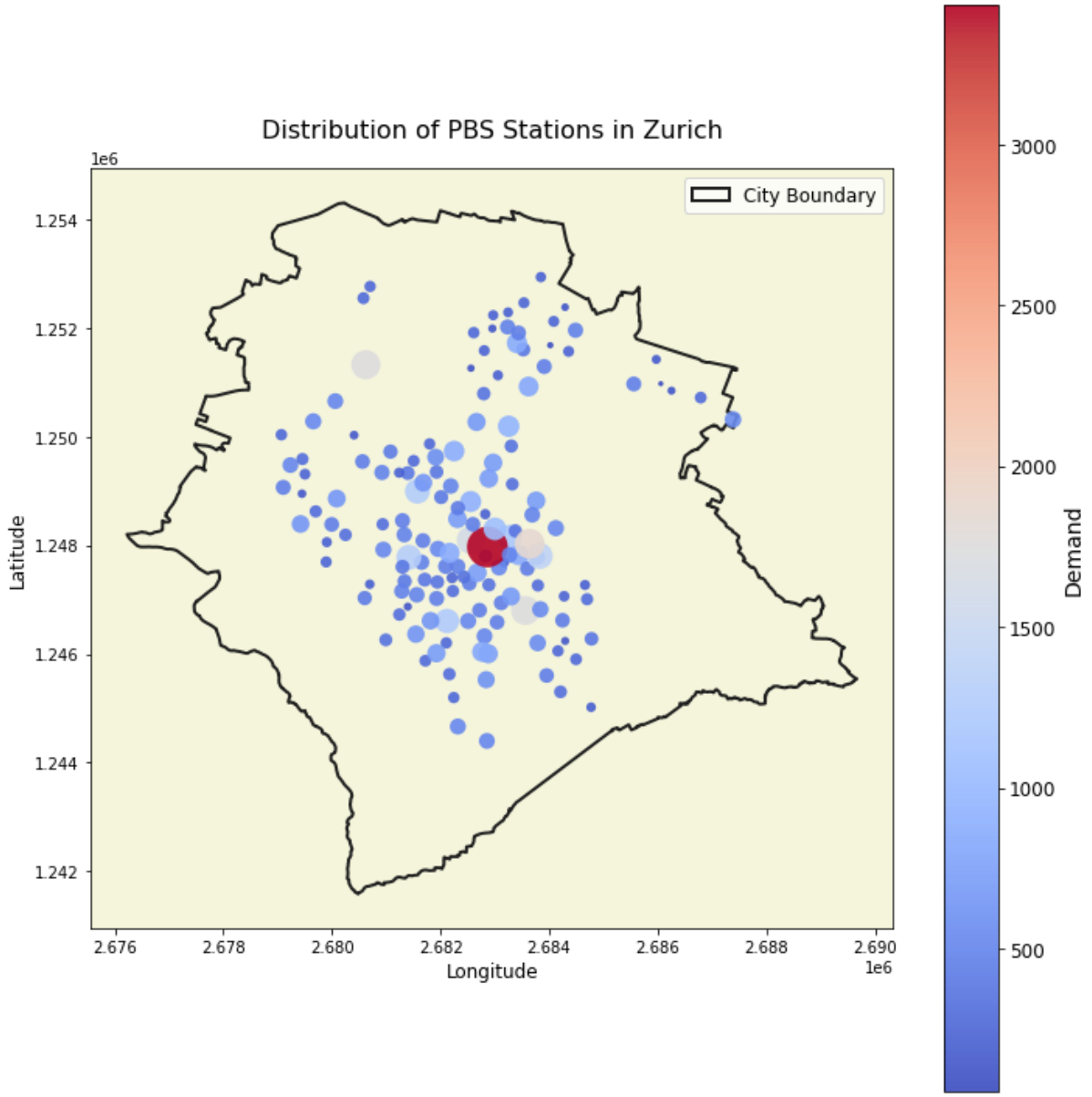


Figure 7. Network plot featuring the Public Bike-Sharing stations situated in Zurich. This visualization, generated using Python, serves to illustrate the demand at the station level. Notably, the size of each circle corresponds to the magnitude of demand, with larger circles indicating higher demand. Additionally, a color ramp is used to demonstrate the station-level demand, with warmer color indicating higher demand.

As we can see from figure 7, the PBS demand is greater in the city center. The further we go from the city center, the lower the demand becomes. Even though the PBS station with demand value close to 3500 could be considered as an outlier, it has been kept in the data analysis and model training since it is in the city center, and it plays a significant role in the demand predictions and the whole analysis. However, it might have influenced significantly the prediction results and the accuracy of the models.

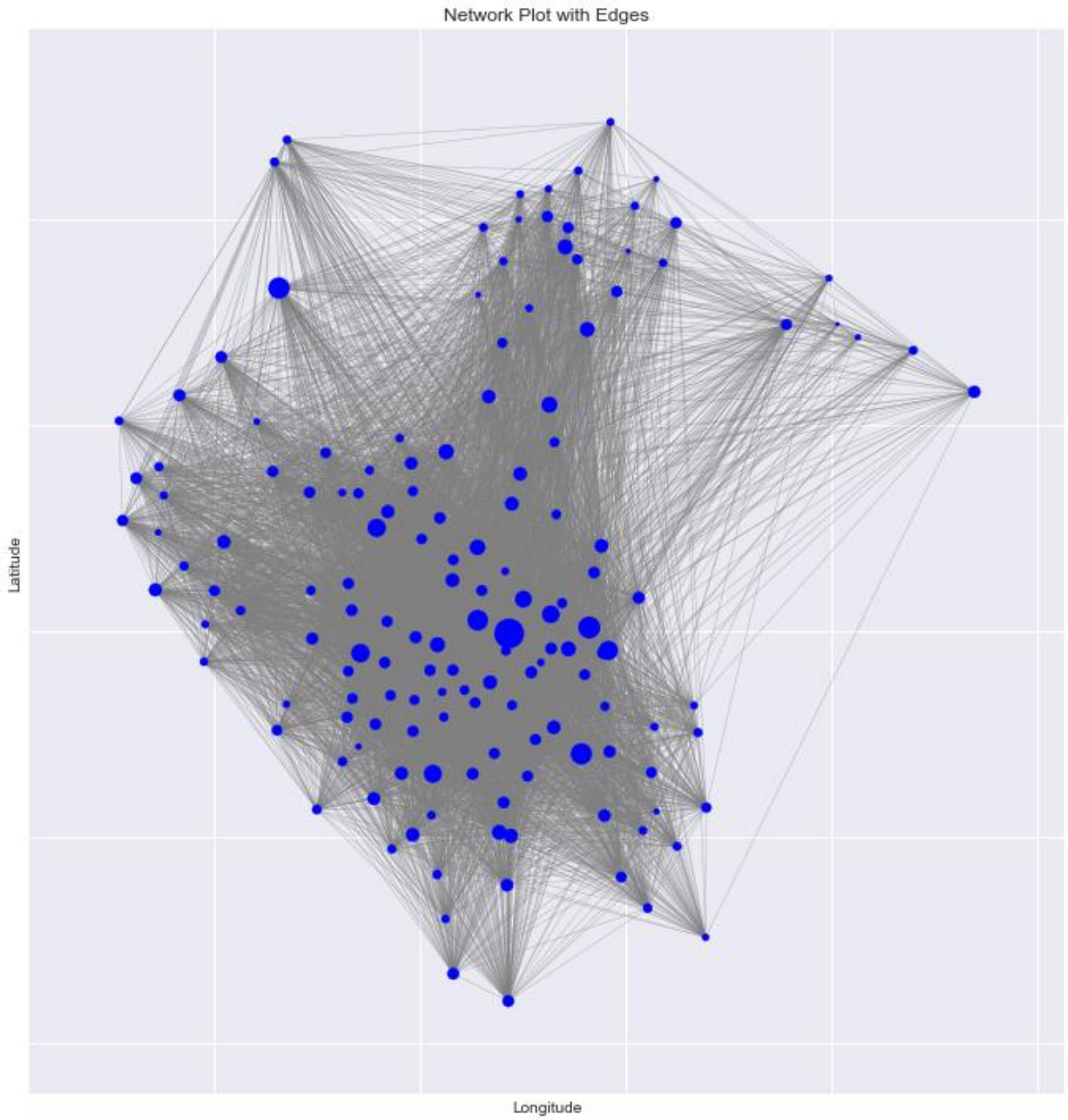


Figure 8. Network plot of the Public Bike-Sharing stations in Zurich. Whenever trips exist between these stations, the nodes in the network are linked by edges.

Figure 8 demonstrates the PBS network with the edges. Each node (PBS station) is connected with other nodes by the edges based on the adjacency matrix which is built upon the number of trips between the stations. It can be observed that it is rather complex network. Hence, there is a need for computational power, and a model that can capture the underlying patterns within this network.

As discussed in the data processing module, the different influencing factors are calculated using different methods. Some of the independent variables are illustrated as examples on maps. Figure 9 shows the land use mixture estimated by the entropy index. The land use types utilized for the analysis are residential, commercial, and industrial. Then the entropy is estimated. The stations with entropy equal to zero have only one land use type. The greater the

entropy, the more types of land use are within the 300-meter buffer zones of the stations. The minimum value is 0, and the maximum value is 1,06.

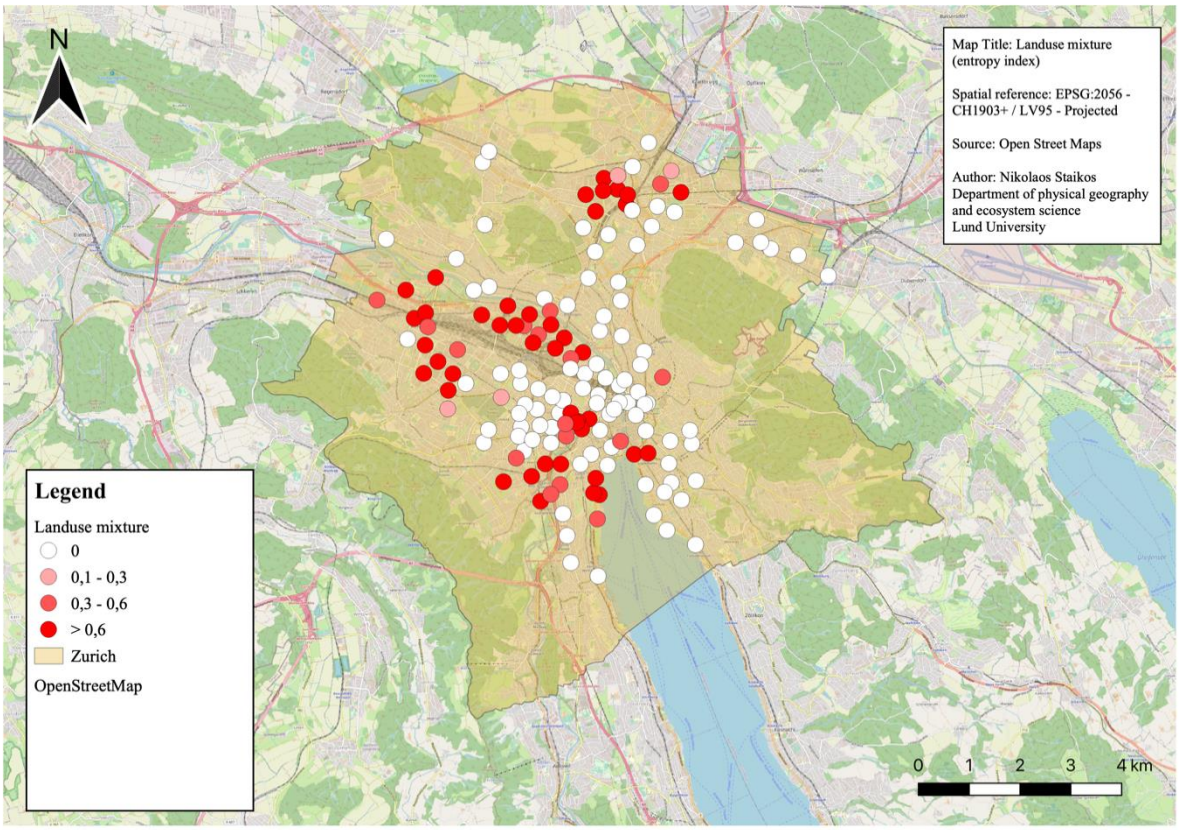


Figure 9: Map showing the land use mixture within the 300-meter buffer zones of each Public Bike-Sharing station in Zurich. The land use mixture is calculated by the entropy index of three land use types: residential, commercial, industrial.

Figure 10 demonstrates the population density corresponding to each 300-meter buffer zone of each PBS station. There are five classes. The minimum value is 144, and the maximum value is 7030 people. Figure 11 illustrates the errands density, i.e., the everyday shops and stores within the buffer zones of each PBS station. The minimum value is 1 and the maximum value is 272. Five classes are created.

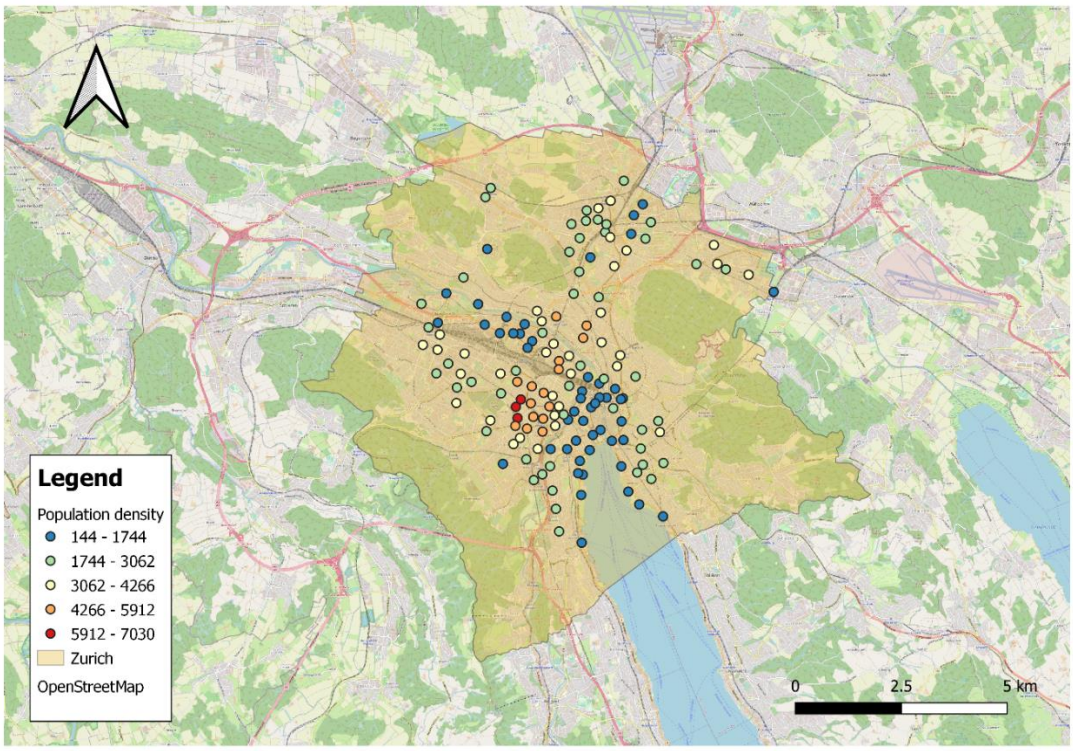


Figure 10: Map showing the population density within the 300-meter buffer zones of each Public Bike-Sharing station in Zurich.

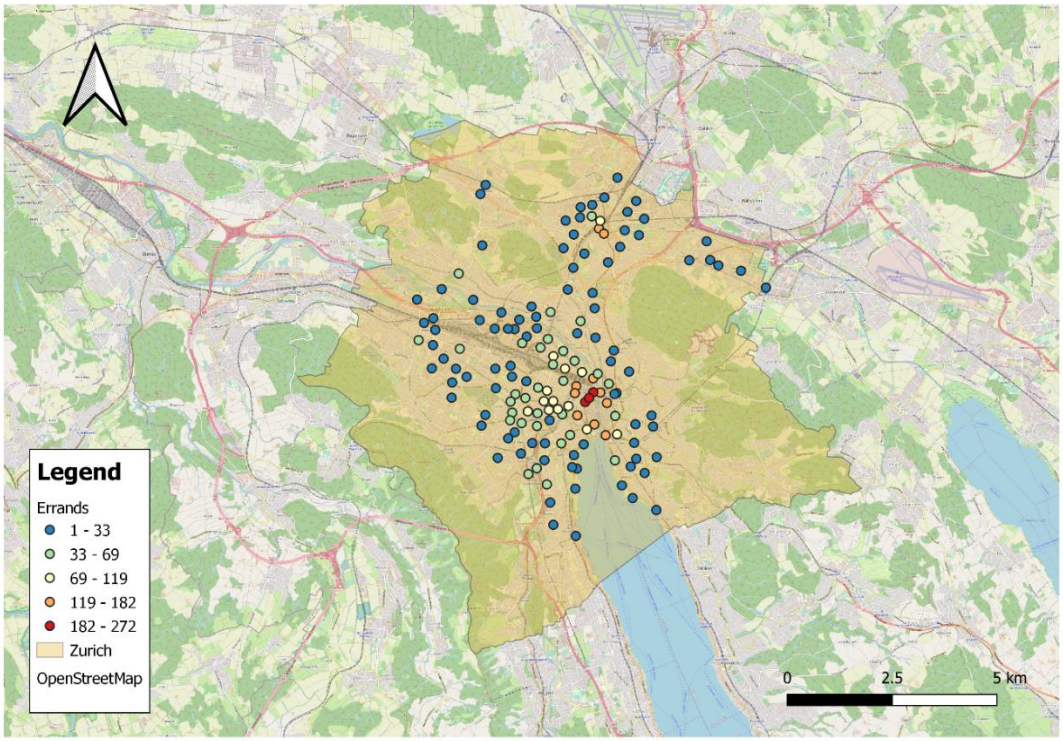


Figure 11: Map showing the errand density within the 300-meter buffer zones of each Public Bike-Sharing station in Zurich.

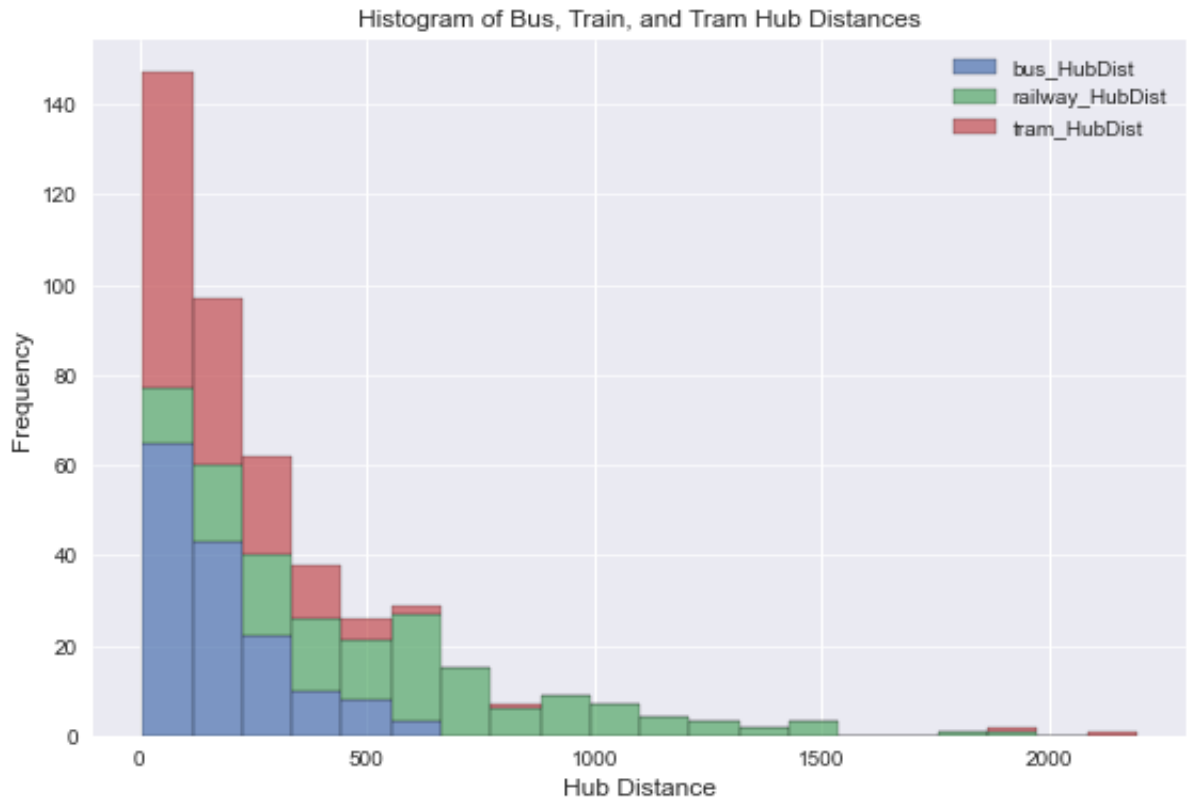


Figure 12. Histogram illustrating the hub distance from the Public Bike-Sharing stations to the closest bus, train, tram station in Zurich.

Figure 12 illustrates a histogram of the closest distance from the PBS stations to the bus, train, and tram stations against their frequency within the dataset. As we can see, the Hub Distance to bus and tram stations is lower than 500 meters for most of the PBS stations. On the contrary, the Hub Distance to railway stations reaches 1500 meters or more.

4.2 PBS station-level demand predictions

The final output from all the models is the PBS station-level demand prediction shown in Figure 13. Four ML and two DL models are developed to predict the PBS station-level demand. It can be observed from the figures in Appendix A that there are differences in the prediction results. First, the range of the demand prediction values is slightly different from model to model. Multilinear regression and MLP regressor appear to have relatively similar range of values, from around 400 to 1200. Also, it can be seen from the spatial distribution of the demand prediction that the results are relatively the same; higher demand is expected in the city center and the further we go from the city center the lower the demand. For SVM and RF regressor it can be seen that more stations in the city center are expected to have higher demand values. SRGCNN demand prediction exhibits the lowest range of values. Additionally, most of the stations exhibit demand values in the upper limit of the range of values. Finally, the demand values from SRGCNN-GW show that they are expected to be very high in the city center and very low in the outer area of Zurich.

The figures in Appendix B demonstrate the errors (real – predicted) for each ML and DL model. A color ramp is used to show if the station-level demand prediction values are underestimated or overestimated. Also, the size of the circles, representing the PBS stations, accounts for the amount of the absolute prediction error for each station. The following formula was used to calculate the error plots:

$$error = real\ demand - predicted\ demand$$

(15) Demand prediction error

Figures 13 and 15 illustrate the relationship between the actual and the predicted station-level demand values of the SRGCNN and SRGCNN-GW models. As we can see from the figures, there is a wide range of demand values. Taking the demand distribution into account (Figure 6), only a few PBS stations have station demand greater than 1000. It can be observed that the actual station demand in the city center (highest actual demand value) has a very high error value (Figures 16 and 18). This is why the ML and DL models were run again excluding the station in the city center with a demand value of 3432. The results can be seen in the following sections.

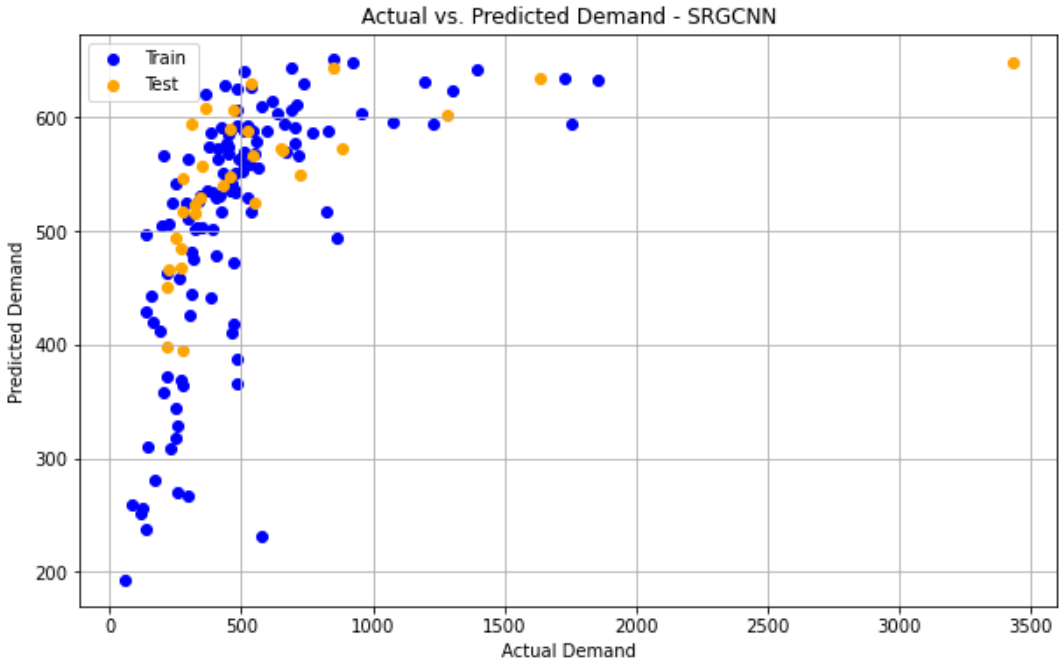


Figure 13. Scatter Plot of Actual vs. Predicted Demand for Train and Test Samples of Spatial Regression Graph Convolutional Neural Network. The plot illustrates the relationship between the actual demand values and the predicted demand values for the train and test samples.

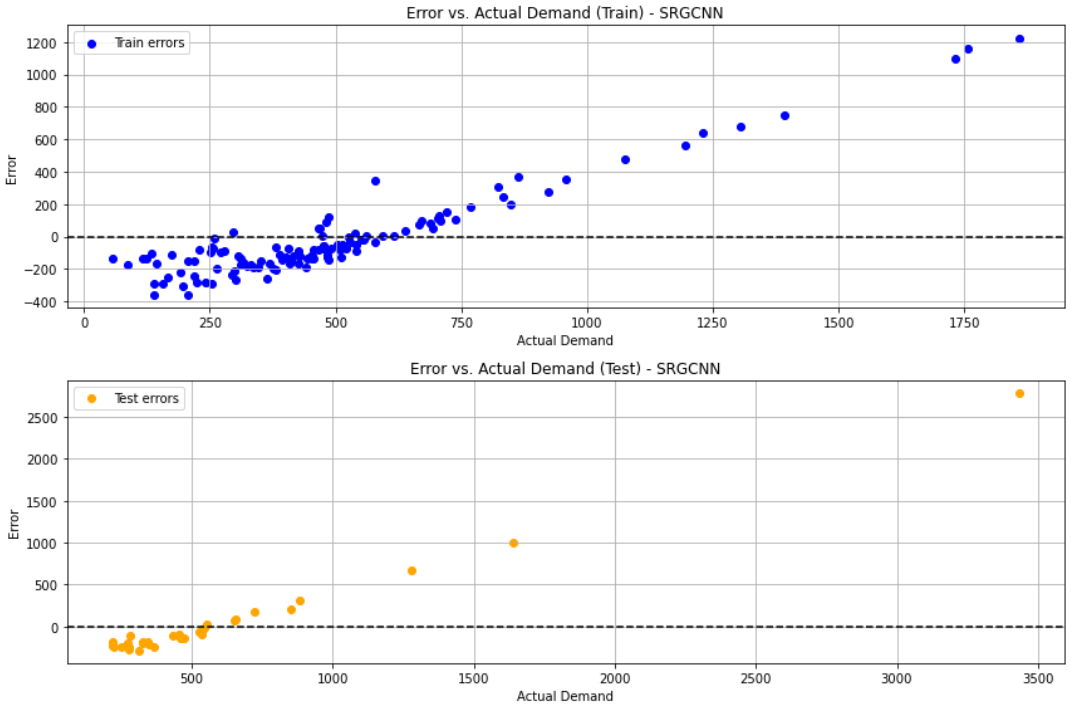


Figure 14. Combined Error Plots for Train and Test Samples of Spatial Regression Graph Convolutional Neural Network. The upper subplot displays the error distribution for train samples, while the lower subplot shows the error distribution for test samples.

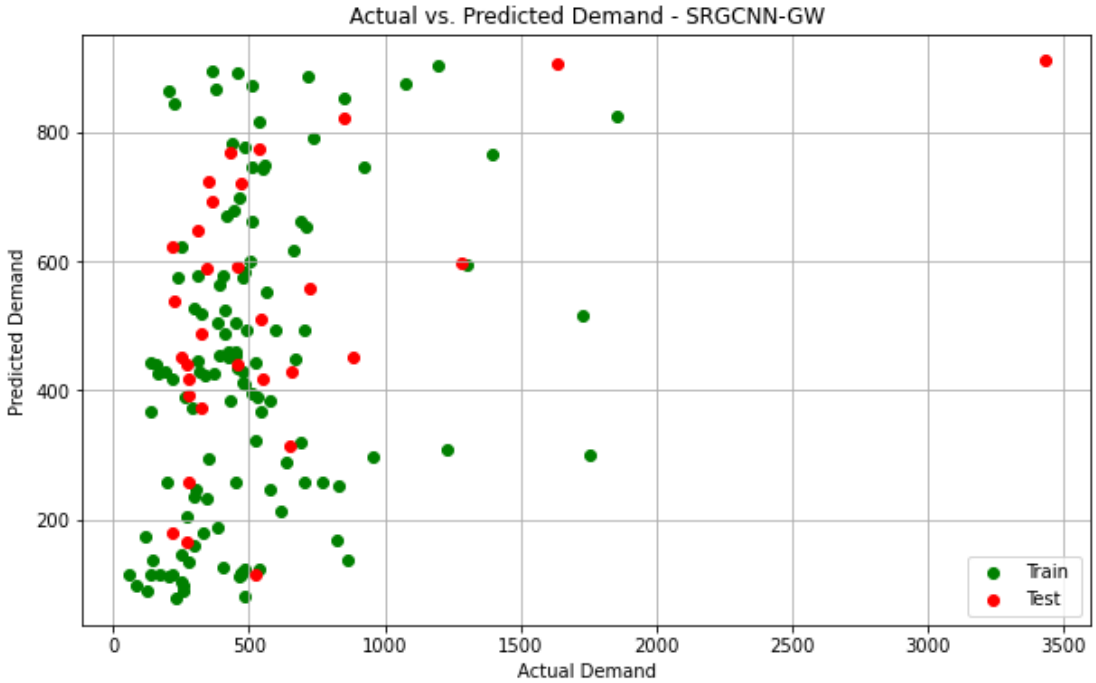


Figure 15. Scatter Plot of Actual vs. Predicted Demand for Train and Test Samples of Spatial Regression Graph Convolutional Neural Network – Geographically Weighted. The plot

illustrates the relationship between the actual demand values and the predicted demand values for the train and test samples.

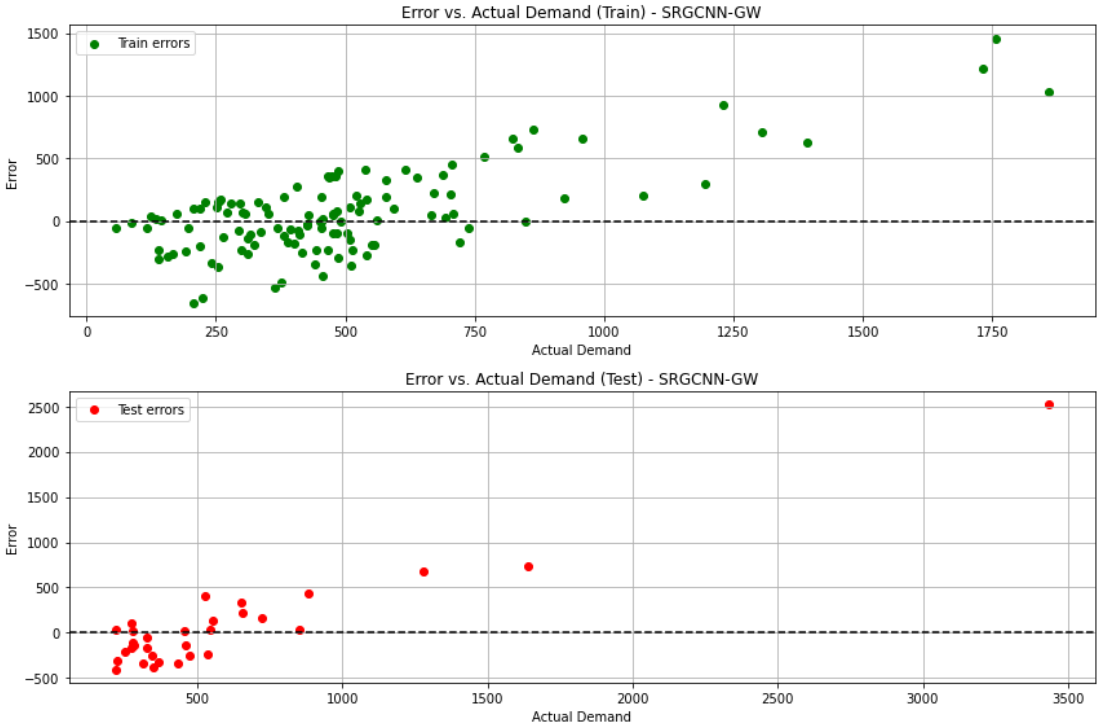


Figure 16. Combined Error Plots for Train and Test Samples of Spatial Regression Graph Convolutional Neural Network – Geographically Weighted. The upper subplot displays the error distribution for train samples, while the lower subplot shows the error distribution for test samples.

From the error plots (Figures 14 and 16), it can be observed that the station-level demand predictions are overestimated for actual demand values of approximately 500 or lower. On the contrary, for actual demand values of around 500 or higher, the station-level demand prediction values are underestimated.

4.3 DL performance evaluation

Several metrics are used to evaluate the performance of the ML and the DL models. These metrics are namely Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-squared.

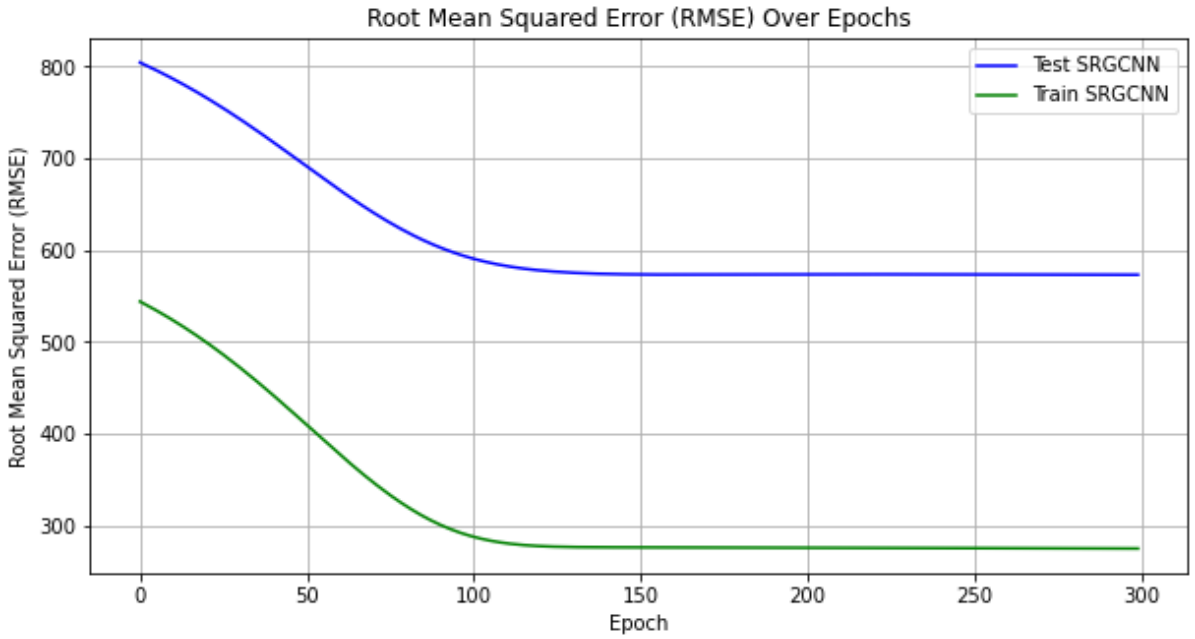


Figure 17. Root mean square error for Spatial Regression Graph Convolutional Neural Network model

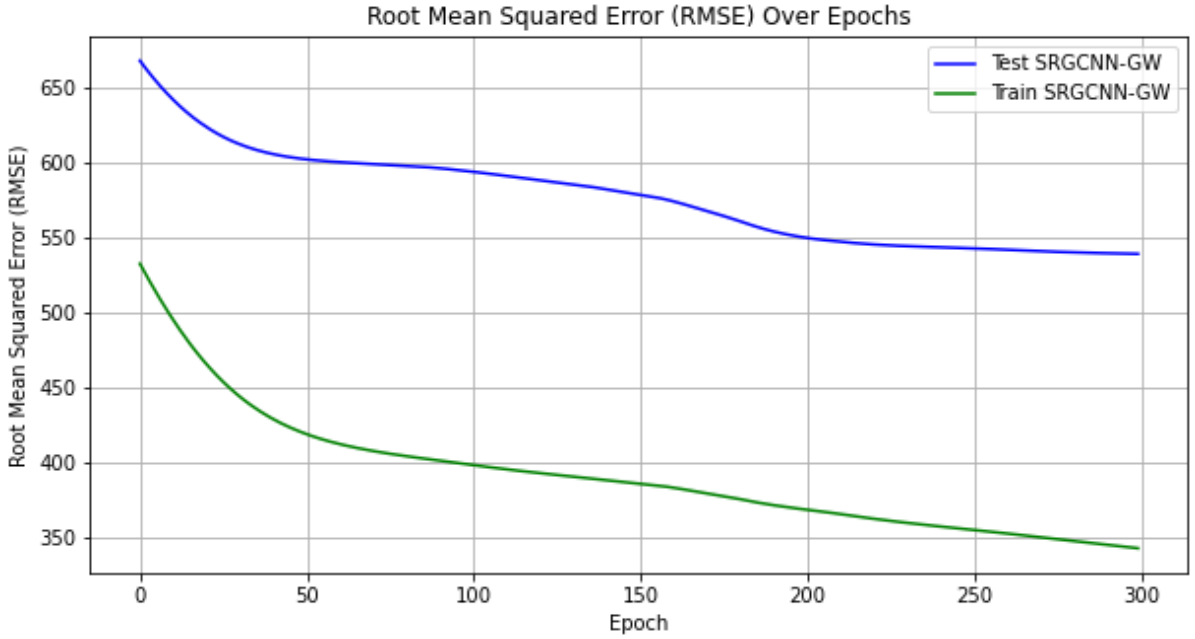


Figure 18. Root mean square error for Spatial Regression Graph Convolutional Neural Network -Geographically Weighted model

As we can see from Figure 17, the RMSE for the train and test data of the SRGCNN model is relatively high during the first 100 epochs, which means that the model's predictions have a significant average magnitude of error compared to the observed values. The RMSE gradually decreases, hence improving the prediction accuracy. After 100 epochs, the RMSE remains high, indicating that the model cannot make accurate predictions. The RMSE for the SRGCNN-GW (Figure 18) indicates that the RMSE values are lower compared to the SRGCNN model.

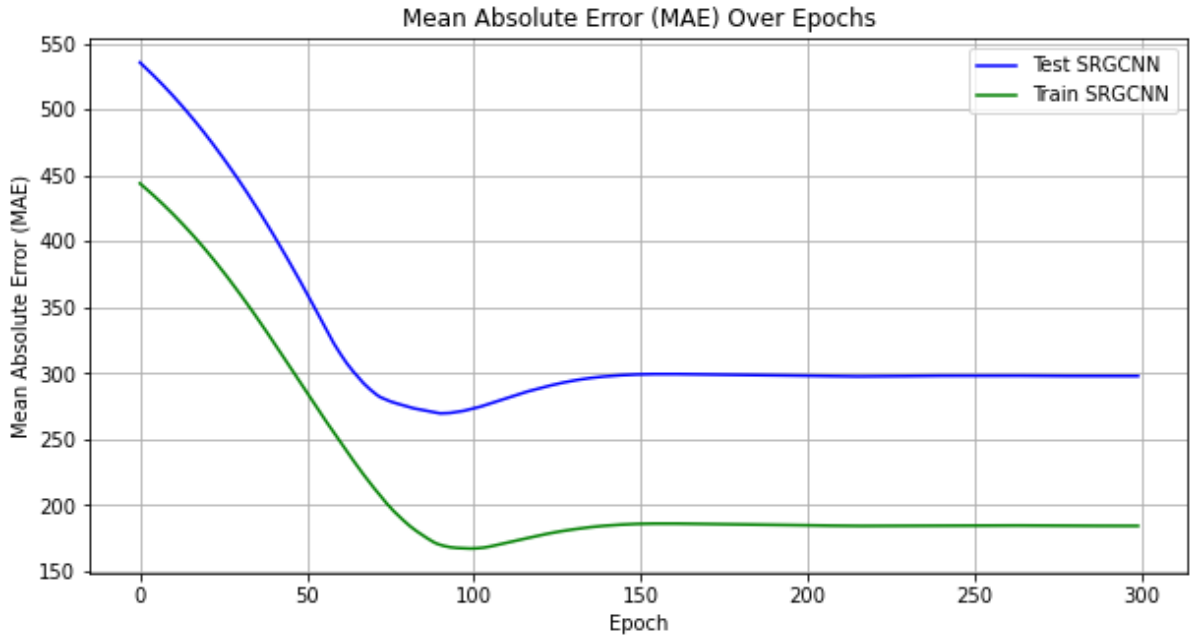


Figure 19. Mean Absolute Error for Spatial Regression Graph Convolutional Neural Network model

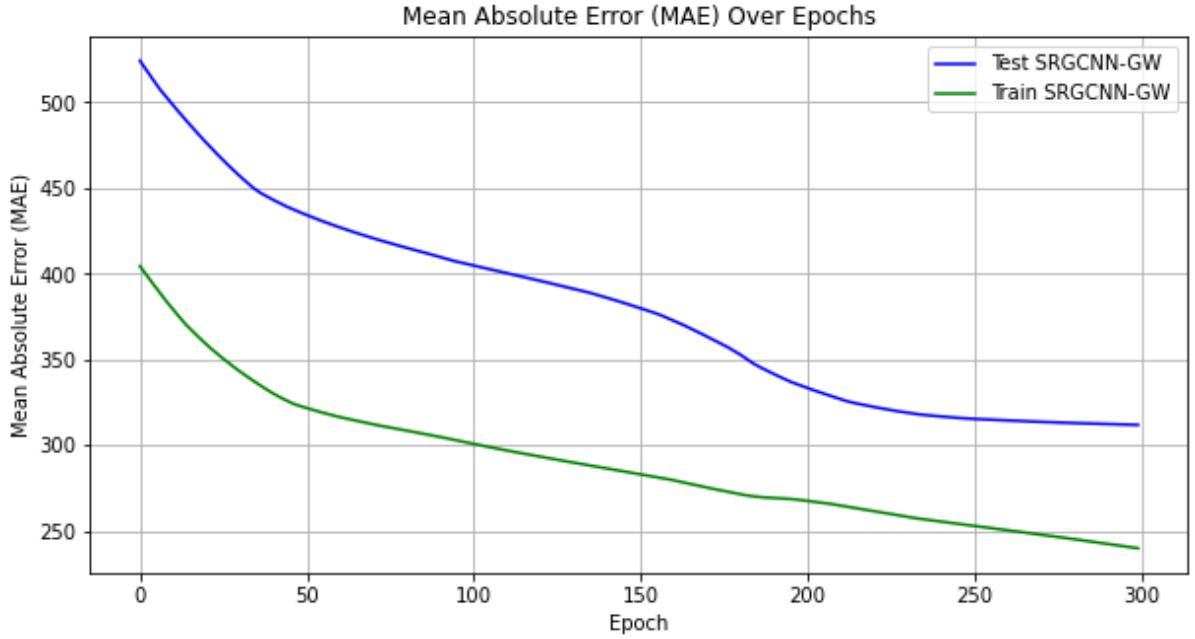


Figure 20. Mean Absolute Error for Spatial Regression Graph Convolutional Neural Network -Geographically Weighted model

In the first epochs, MAE values are relatively high for the SRGCNN model (Figure 19), around 450 and 550 for the train and test data, respectively. This indicates that, on average, the model's predictions significantly differ from the actual values. MAE gradually decreases as training progresses; therefore, the model's ability to make predictions with lower absolute errors improves. MAE values for the SRGCNN-GW (Figure 20) gradually decrease over 300 epochs.

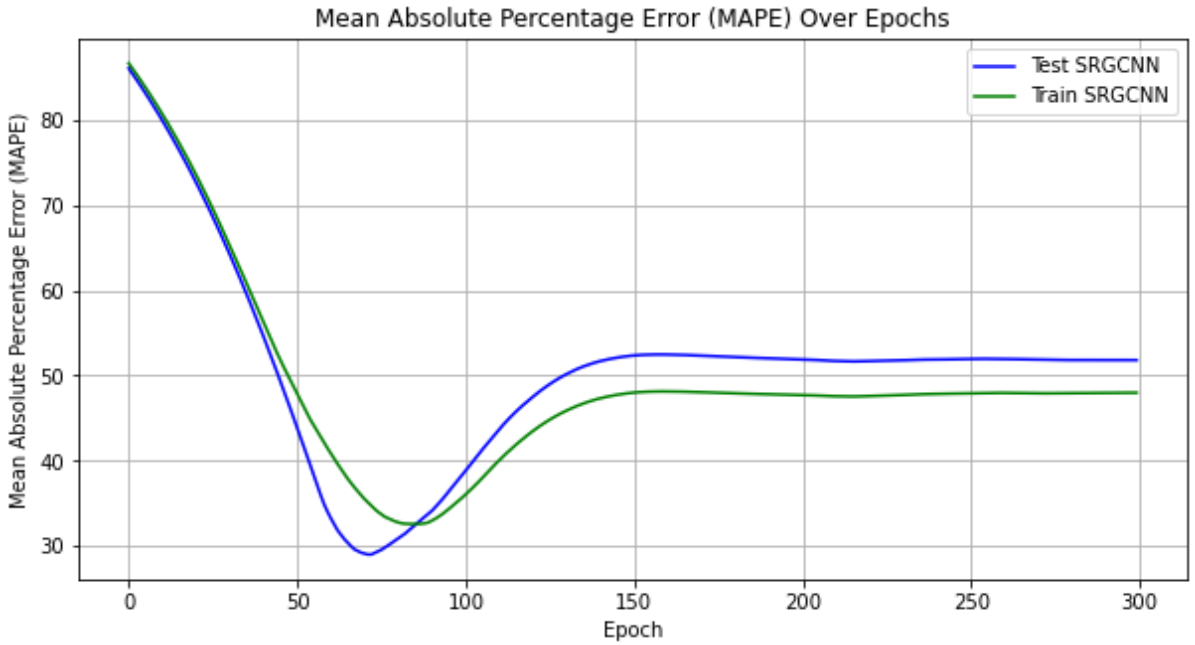


Figure 21. Mean Absolute Percentage Error for Spatial Regression Graph Convolutional Neural Network model

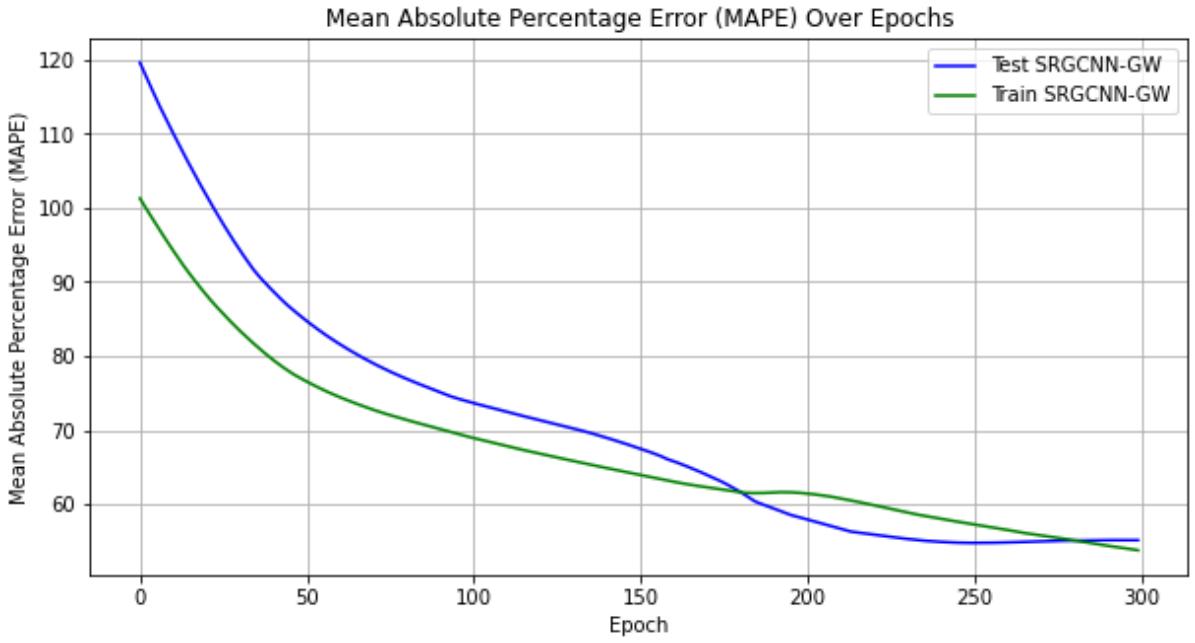


Figure 22. Mean Absolute Percentage Error for Spatial Regression Graph Convolutional Neural Network -Geographically Weighted model

Figures 21 and 22 illustrate the MAPE values for the SRGCNN and the SRGCNN-GW models. Both models begin with high MAPE values, which is typical for the initial stages of the training process. Since MAPE decreases for both models, the predictive accuracy improves. However, the final MAPE values are around 49-55% indicating that the models' predictions might not be highly accurate.

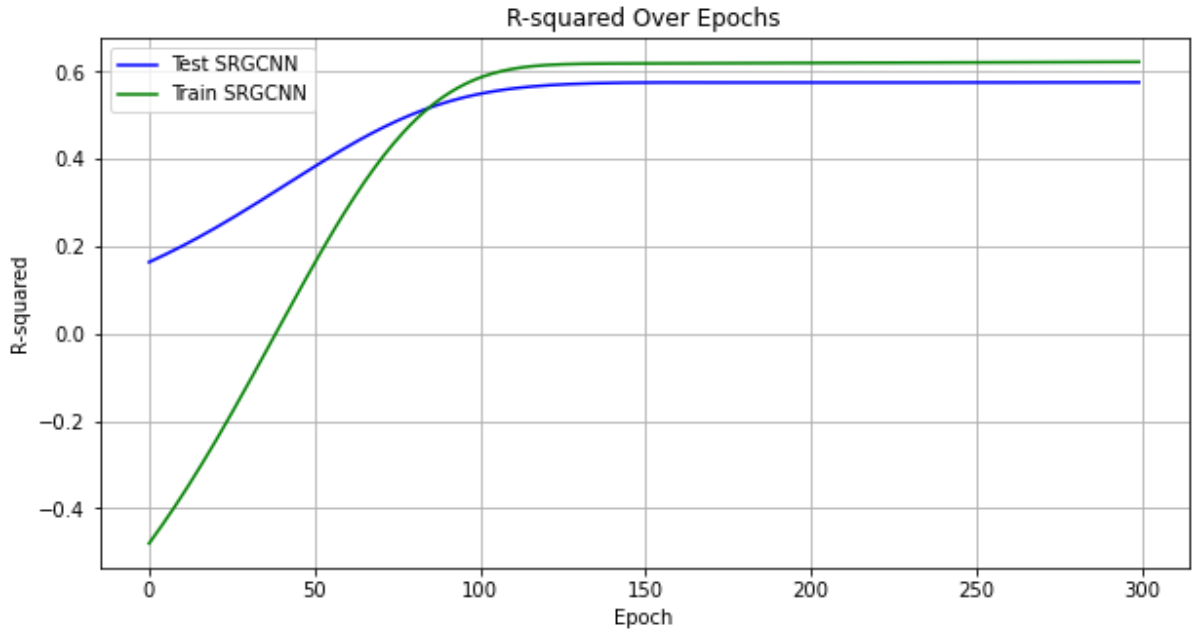


Figure 23. R-squared for Spatial Regression Graph Convolutional Neural Network model

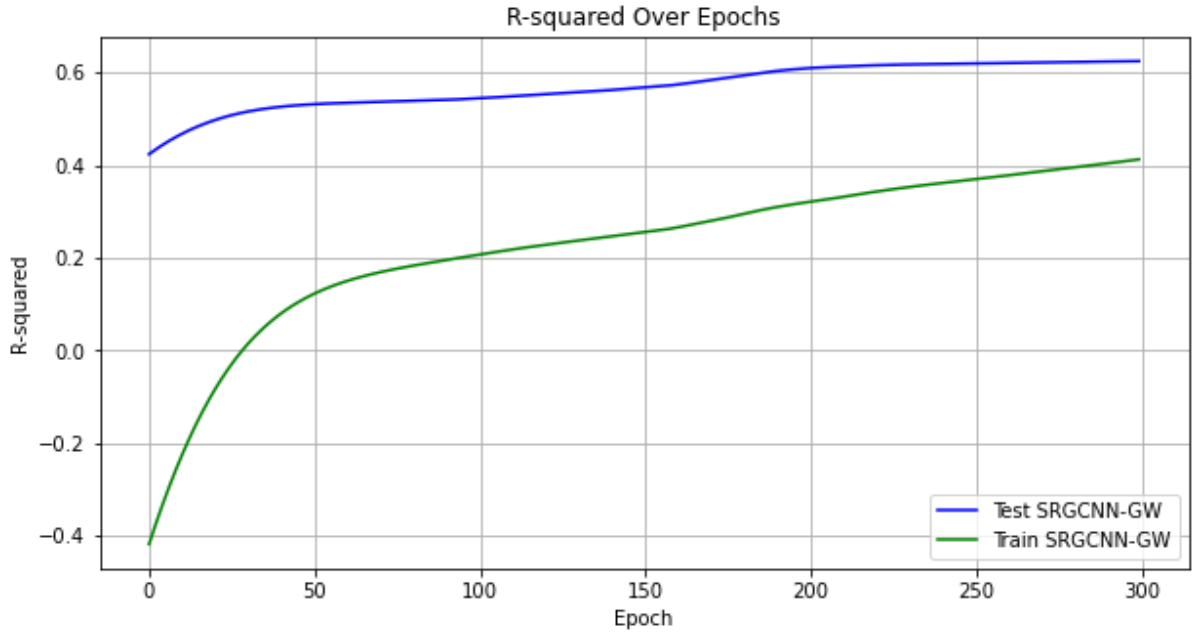


Figure 24. R-squared for Spatial Regression Graph Convolutional Neural Network - Geographically Weighted model

As we can see from Figures 23 and 24, the R^2 for both models begin at a low value of approximately -0.4 for the training model, which indicates that the variability in the target variable poorly explains the model's predictions. As training progresses, the R^2 values increase, and therefore improving the model's ability to capture the variability in the data. It is shown that after 100 and 150 epochs for the SRGCNN and the SRGCNN-GW models respectively, the R^2 values are around 0.6, which suggests that both models improve their

ability to capture and explain the variability in the dependent variable, resulting in better overall fit to the data.

4.4 Comparison of ML and DL models

In this section evaluation metrics are compared to assess the ML and DL models' performance. Two distinct scenarios were implemented in order to scrutinize the different cases. In the initial scenario, all the PBS stations in Zurich are used to train the models and generate demand predictions. In the second scenario, PBS stations with demand exceeding 2000 were excluded from the analysis. As mentioned in section 4.1, the mean demand of the PBS stations is 497. Therefore, PBS stations with demand exceeding 2000 would greatly impact the models' performance results.

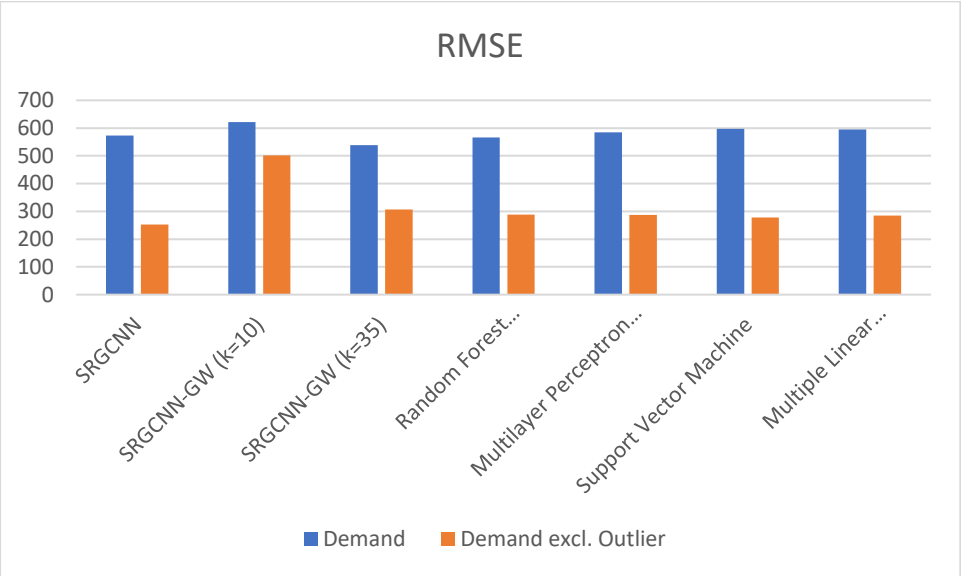


Figure 25. Comparison of deep learning and machine learning models based on root mean square error

Figure 25 demonstrates the RMSE values for each ML and DL model. Among all models, the SRGCNN achieves the lowest RMSE, followed by the rest of the models. SRGCNN outperforms the rest of the ML and DL models when it comes to RMSE, which means that they provide more accurate predictions. The SRGCNN model has the lowest RMSE among all models, which suggests that it is the most accurate model based on this metric.

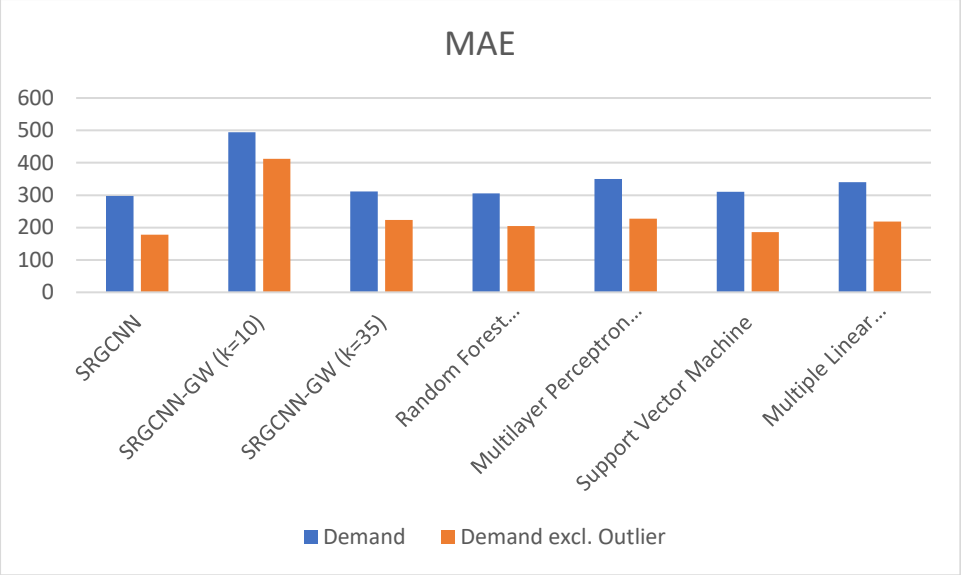


Figure 26. Comparison of deep learning and machine learning models based on Mean Absolute Error

Regarding MAE values (Figure 26) among all models, the SRCGNN has the lowest MAE, followed by RF regressor. SRGCNN and SVM exhibit relatively similar MAE values. Therefore, the SRGCNN model appears to be the most accurate in making predictions with lower absolute errors.

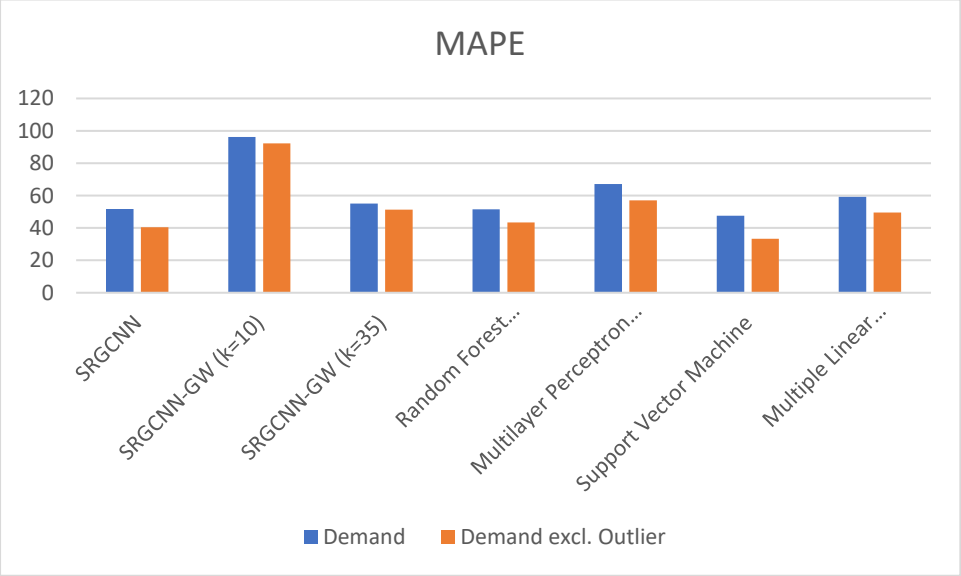


Figure 27. Comparison of deep learning and machine learning models based on Mean Absolute Percentage Error

As shown in Figure 27, SVM has the lowest MAPE among all models. The DL models and RF regressor have competitive MAPE values compared to the rest of the ML models. However, SVM appears to be the most accurate model based on this metric, with the DL models and RF regressor being a good alternative.

Figure 28. Comparison of deep learning and machine learning models based on R-squared

Figure 28 illustrates the R-squared values among all the models implemented in this thesis. For the ML models, the Random Forest Regressor before excluding the outliers and SVM after excluding the outliers achieve the highest R-squared value, indicating a good performance in terms of explaining the variability in the data. The DL models have significantly higher R-squared values than the ML models. Hence, SRGCNN and SRGCNN-GW achieve the best fit to the data compared to the ML models.

Overall, SRGCNN and SRGCNN-GW outperform the ML models in most metrics used to evaluate the models' performance, with RF regressor being a good alternative since it achieves the lowest MSE and RMSE values.

Table 9 (Appendix C) shows all the models' performance results of the initial data sample and the data sample excluding the outliers (demand values less than 2000). Comparing these results for demand excluding outliers, SRGCNN still shows the lowest RMSE, MAE, and MAPE, as well as the highest R-squared. SRGCNN-GW (k=35) also performs well in these metrics. Random Forest Regressor, Multilayer Perceptron Regressor, and Multiple Linear Regression show relatively lower performance, especially in terms of R-squared. It's clear that considering outliers in demand has an impact on the model performance, and DL models such as SRGCNN seem to handle these cases better in terms of prediction accuracy and explanation of variance.

4.5 Feature importance

Feature importance is a method to determine the relative importance of the independent variables when predicting the dependent variable. This is of paramount importance in feature engineering and the interpretation of ML models. Several studies have utilized different feature importance methods (Strobl et al., 2007; Williamson et al., 2023). The most common approach is the feature ranking method. This method assigns a score to each variable depending on how it contributes to the model's accuracy. In this thesis, feature permutation importance is utilized to determine the feature importance of the independent variables for the four ML models (Pedregosa et al., 2011).

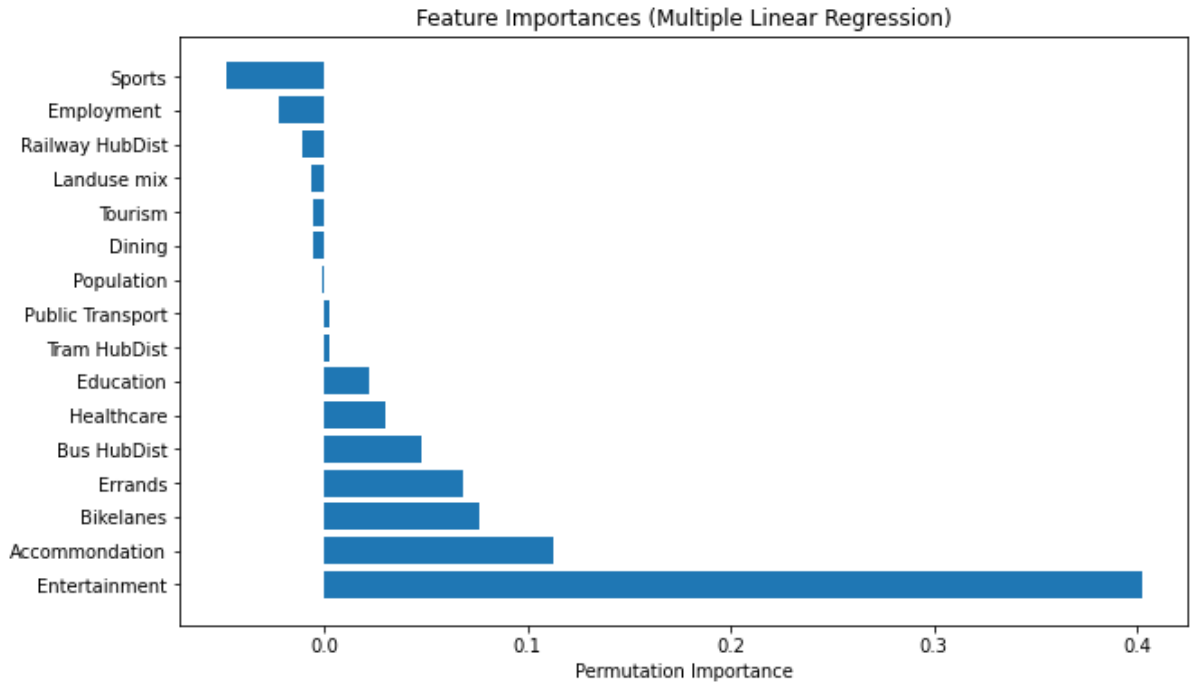


Figure 29. Feature importance for Multiple Linear Regression. A feature with high importance contributes more to the model's accuracy than a feature with low importance.

For Multiple Linear Regression it is found that the most important features are sports, closest distance to bus stop, errands, bike lanes, and accommodation. Particularly noteworthy is the prominence of entertainment as the most influential feature, evident from the observations in Figure 29. For MLP regressor, the most important features are entertainment and accommodation, followed by bike lanes, dining, education and errands (Figure 30). Moreover, for SVM, healthcare, closest distance to bus stop, sports, and public transport are the most important features, with entertainment exhibiting the highest importance of all (Figure 31). Finally, population, dining, entertainment, and errands are the most important features for RF regressor (Figure 32).

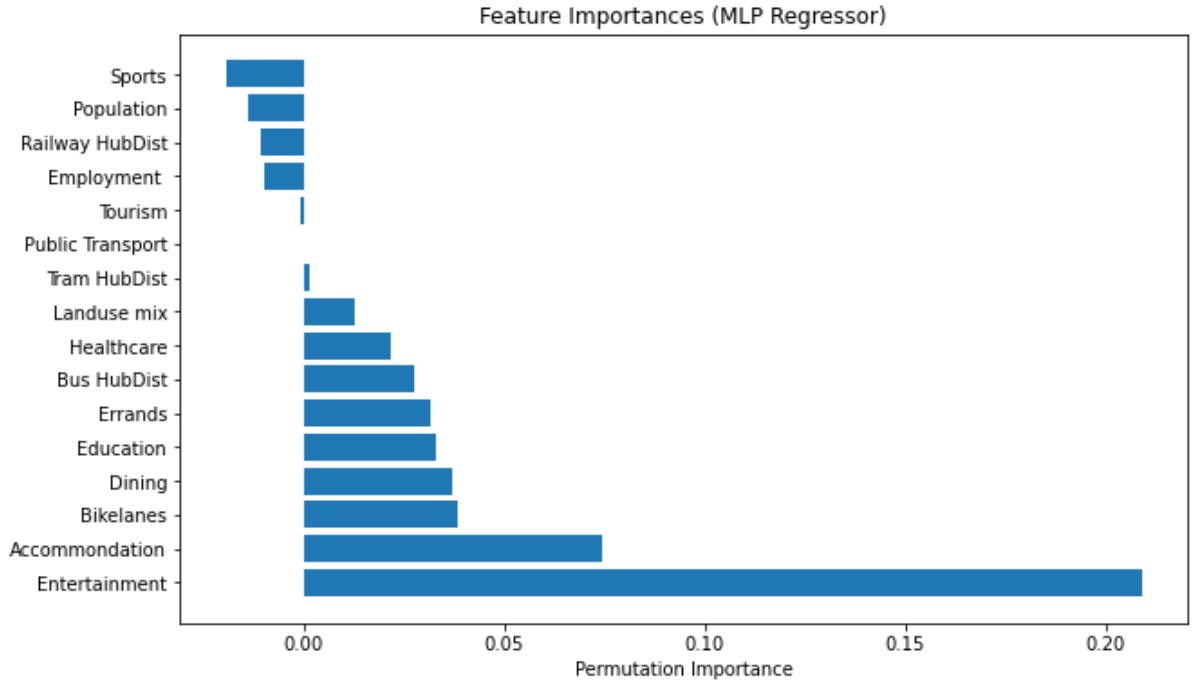


Figure 30. Feature importance for Multilayer Perceptron Regressor

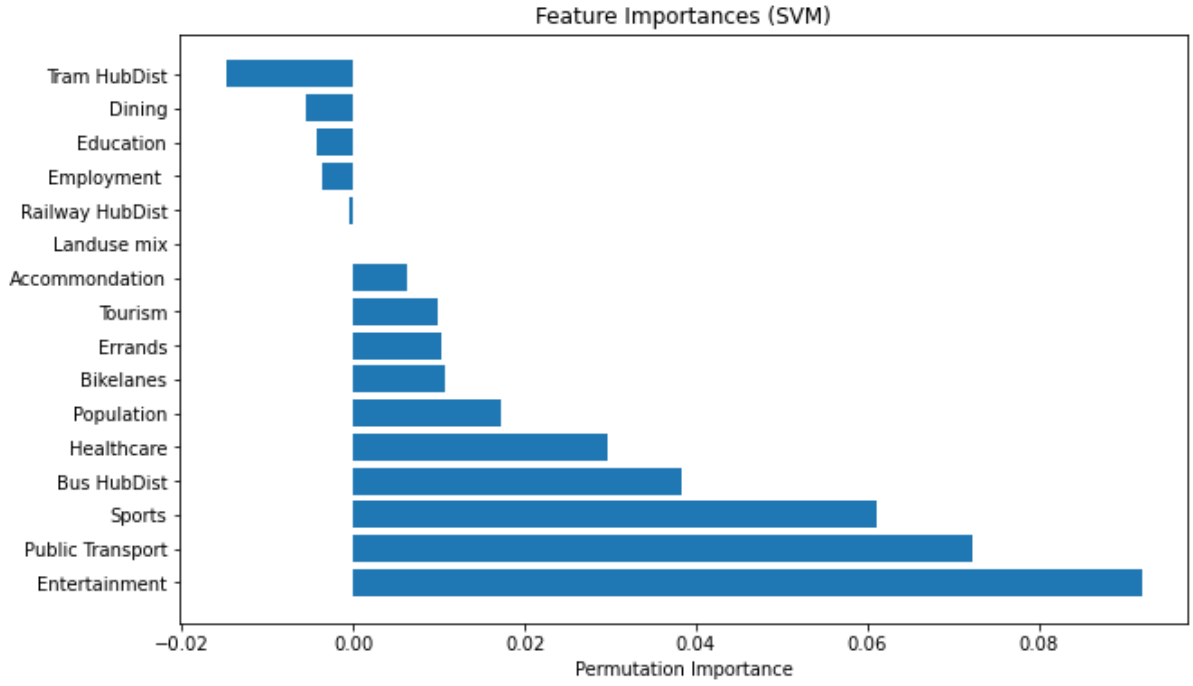


Figure 31. Feature importance for Support Vector Machine

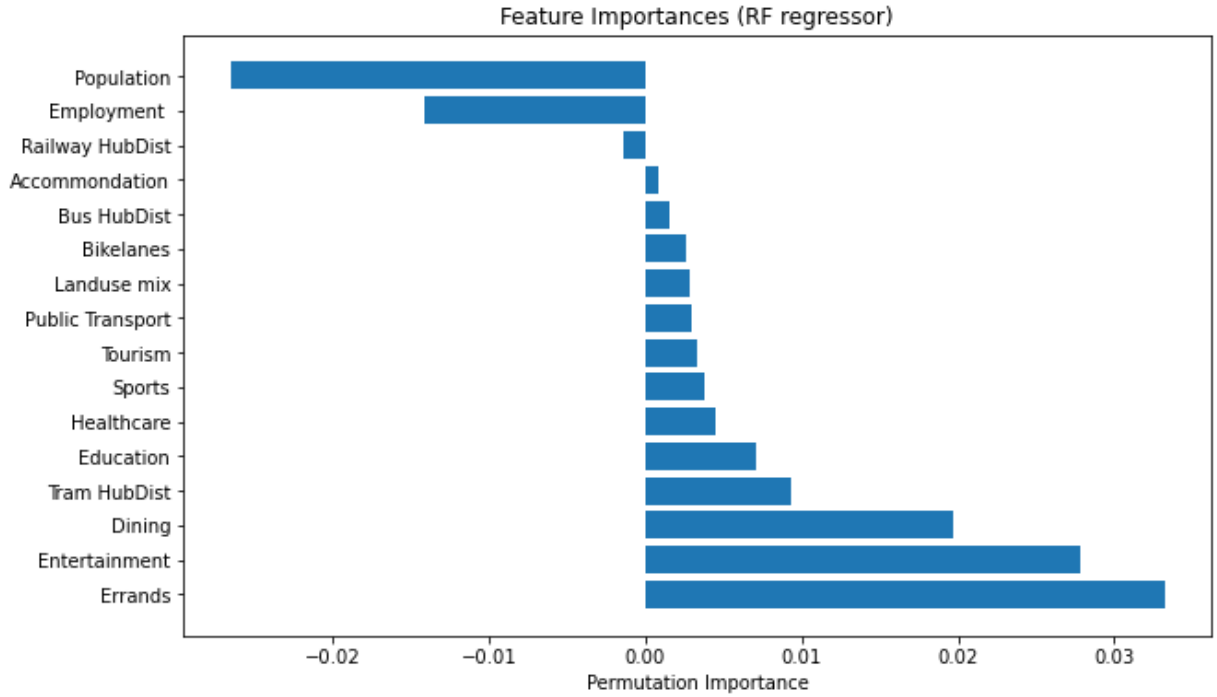


Figure 32. Feature importance for Random Forest Regressor.

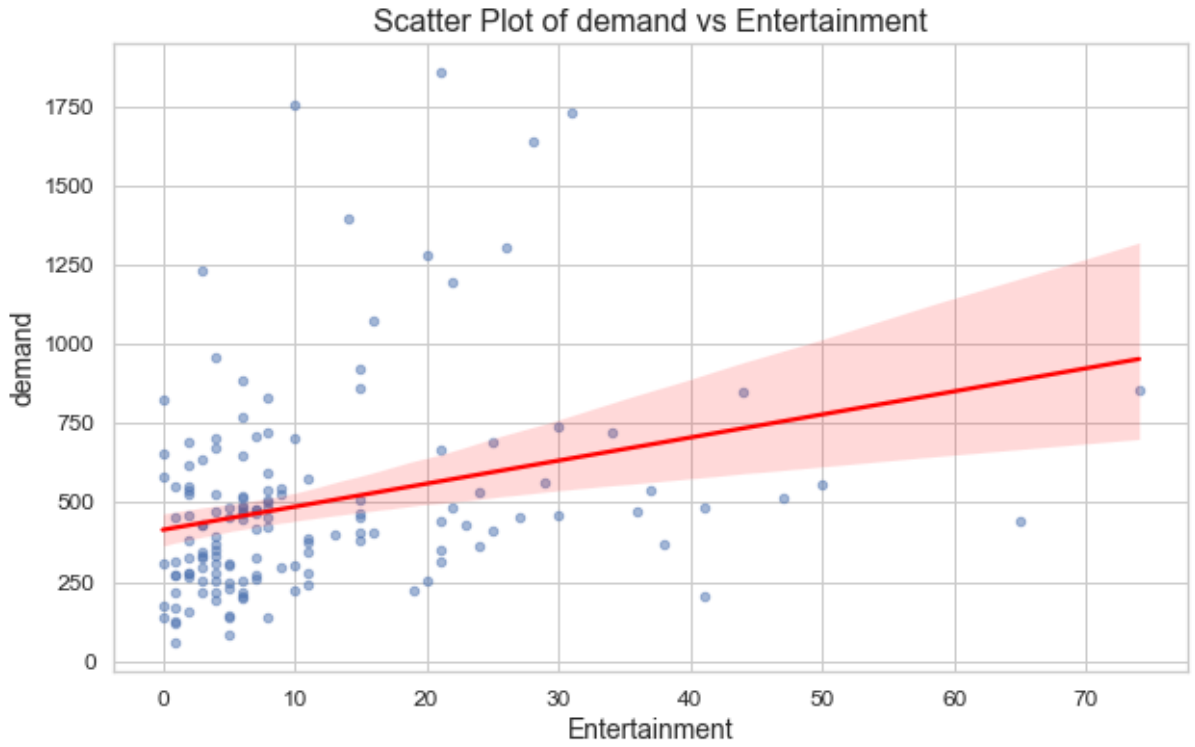


Figure 33. Scatter Plot of 'Public Bike-Sharing Demand' vs. Entertainment

Overall, the most important features for all the ML models are sports, closest distance to bus stop, bike lanes, accommodation, errands, dining, and education. Entertainment shows the highest importance for all the ML models. As we can see from the regression line from figure 33, entertainment density is increasing with higher demand values. Therefore, it is likely that there is a correlation between places for entertainment and the PBS station demand. Regarding

the closest distance to bus stop, it can be seen from figure 34 that it exhibits the opposite pattern. The further a bus stop is located from a PBS station, the less the PBS station demand will be.

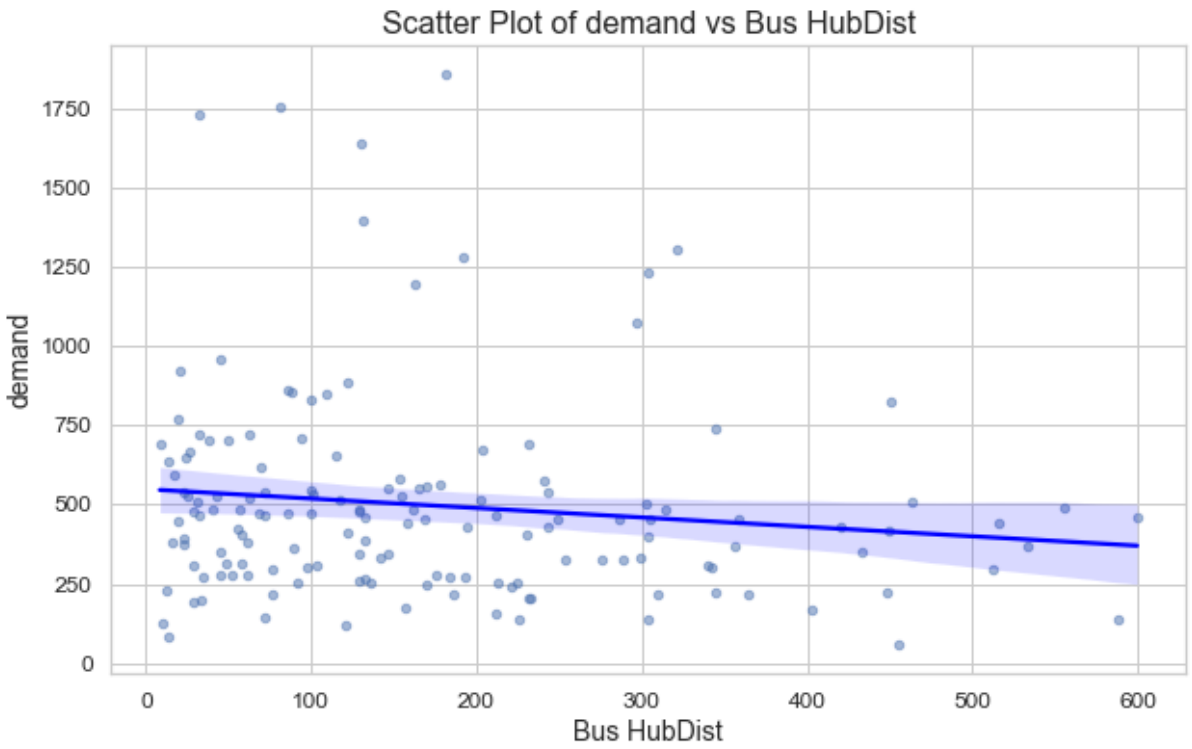


Figure 34. Scatter plot of 'Public Bike-Sharing demand' vs. 'Closest distance to Bus stop'

5. Discussion

As described in the previous sections of this thesis, four ML models and two DL models are implemented in order to predict the station-level demand for the PBS system in Zurich. Taking the models' performance results into consideration, the DL models outperform the ML models' performance accuracy. Specifically, the SRGCNN-GW model shows superior performance across several metrics such as RMSE, MAE, MAPE, and R-squared. This means that the SRGCNN-GW model is more capable of predicting the station demand and capturing the underlying patterns in the PBS data and the influencing factors compared to the other models.

On the contrary, the MLP regressor shows poor performance with low R-squared and high MAE and RMSE values. Therefore, it struggles to capture the relationships within the dataset and might not be a reliable model for predicting the station demand. The performance of the Random Forest regressor and SVM is relatively similar, with varying strengths and weaknesses in different metrics. Overall, the DL models, especially the SRGCNN-GW, appear to be a

better choice when it comes to predicting the PBS station demand with the influencing factors used in this thesis.

Nevertheless, it should be noted that the DL models' performance is still not optimal since the values in several metrics, such as RMSE, MAE and MAPE, are still relatively high. Several factors could influence the models' performance. For instance, complex models, such as SRGCNN and SRGCNN-GW, are potentially better at capturing intricate relationships in the data. However, they are more prone to overfitting, especially if the dataset is relatively small. This thesis uses 151 PBS stations and 16 influencing factors to train the models and predict the station demand. Even though the dataset size appears sufficient, more PBS stations or influencing factors could lead to better generalization and enhanced performance. For instance, Zhu et al. (2022) utilized approximately 6000 nodes in the network. Hence, the DL models were more capable of capturing the underlying patterns since the sample was greater.

Moreover, the selection of hyperparameters, such as learning rates, hidden layers, and dropout, can play a significant role in model training and can affect the model performance. In this thesis, random search is utilized to define the hyperparameters for the ML models. The number of iterations while randomly searching the best hyperparameters is being experimented in order to achieve the best performance. For the DL models, hyperparameter tuning is conducted through experimentation by comparing the evaluation metrics' results. Nonetheless, the ML and DL models still exhibit relatively poor performance.

Additionally, the quality and relevance of the data used for model training can significantly affect the results. In this case, all the influencing factors are selected carefully, based on previous studies on PBS station forecasting. However, further studies can reveal alternative feature selection, or better combinations of spatial features. The influencing factors can differ depending on the nature of the dependent variable to be predicted, the study area and its surroundings. Also, the buffer zone size has played a significant role in the analysis. 300-meter buffers are created around each PBS station to calculate the 16 independent variables. A smaller or larger buffer size could have led to slightly different results.

Regarding the independent variables, some of them, especially the POIs, are probably inappropriate for this task. For example, sports and accommodation variables have a relatively small range of values and for a lot of PBS stations they were zero. This significantly affects the model training and, therefore, influences the model performance. Furthermore, it can be seen from figure 6, showing the PBS station demand distribution, that most of the PBS stations have a demand of around 500. However, a few stations in the city center have a demand of approximately 1500-2000 and 3500. This is why the ML and DL models were applied for both the initial dataset and without the outlier. The results showed that all the models performed better without the outlier and, therefore, the prediction accuracy was significantly improved.

Furthermore, the hub distance between the PBS stations and the closest bus, train, and tram stations is calculated in QGIS, using the Euclidean distance. Even though it is a suitable method, calculating the network distance might have resulted in better performance. Also, the

bike lane or cycle path datasets found needed to be more extensive and lacked a lot of information. Hence, the road dataset from GeoFabrik is used to determine the bike lanes by excluding all the road types inappropriate for cycling. This has been done empirically and might have some underlying error. The models' performance might be improved if there is a specific and reliable dataset for cycle paths in future case studies.

Another important aspect regarding the data is the season. The data utilized for the station-level demand predictions were exclusively during the summer season, which could have led to some limitations. The data can be used to predict the short-term demand around this period of the year, but not the long-term demand since it cannot account for varying weather conditions and seasons. Furthermore, the usage of public bikes may differ significantly during the summer months, with individuals engaging in distinct activities compared to other seasons. Therefore, this variance in user behavior added another layer of complexity to the demand patterns which should be considered in future endeavors.

The graph structure implemented in this thesis is based on the number of trips between the PBS stations. Different graph structures could improve the models' performance. For example, alternative graph structures could be based on the duration of the trips, the Euclidean distance between the stations, or the network distance between the stations. Then, a comparative analysis could be implemented in order to determine which is the most optimal and appropriate graph structure for the task. When it comes to data analysis, standard scaler library is used to normalize the influencing factors. There are different normalizing techniques that can greatly affect the results.

6. Conclusion

The advent of docked public bicycle-sharing systems has drawn much attention recently. Many cities have introduced PBS systems to facilitate the daily commute of their citizens. Machine Learning and Deep Learning methods can be utilized in order to predict the PBS station-level demand and evaluate the spatial distribution of the stations. In this thesis, two DL models, namely SRGCNN and SRGCNN-GW, are used for PBS station demand forecasting and compared with four traditional ML models. Spatial features such as POIs, population density and other factors influencing the station demand are used to train the models. 300-meter buffer zones around the PBS stations are utilized to calculate the spatial features. Evaluation metrics, namely RMSE, MAE, MAPE, and R-squared, are used to compare the ML with the DL models' performance. The evaluation metrics demonstrate that the DL models have similar performance and are better at predicting the PBS demand in terms of RMSE, MAE and R-squared, than the rest of the ML models. Two graph structure are applied in the DL models based on the number of trips between the PBS stations and the k nearest neighbor for the SRGCNN and the SRGCNN-GW model, respectively.

Several constraints have been identified within this study, which could be the focus of future research endeavors. One of them is the number of stations which is relatively low and,

subsequently, insufficient for both ML and DL model training. For instance, Zhu et al. 2022 used approximately 6000 nodes to create the network. Hence, future studies could consider more nodes to enhance the prediction accuracy. Different graph structures could also be implemented, based on the trip duration, the Euclidean distance, or the Network distance between the stations. This could provide different results and show which graph structure plays a significant role in the station demand prediction. Moreover, in this study only the start of each trip is considered to calculate the station demand. In future studies, each trip's start and end of could be considered. In this way, PBS demand underlying patterns could be more easily revealed and potentially improve the PBS station demand prediction accuracy.

Overall, this thesis managed to examine and reveal the most important factors influencing the PBS station-level demand in Zurich and predict the demand through four ML and two DL models. Evaluation metrics are used to assess the models' performance. The goal of this thesis, to lay the foundation for a complete analysis that could be applied to different cities around the world, has been met. In this way, such a study can facilitate PBS network evaluation and planning.

References

Bao, J., Shi, X., & Zhang, H. (2018). Spatial Analysis of Bikeshare Ridership With Smart Card and POI Data Using Geographically Weighted Regression Method. *IEEE Access*, *6*, 76049–76059. https://doi.org/10.1109/access.2018.2883462

Bashir, D., Montañez, G. D., Sehra, S., Segura, P. S., & Lauw, J. (2020). An information-theoretic perspective on overfitting and underfitting. AI 2020: Advances in Artificial Intelligence, 347–358. https://doi.org/10.1007/978-3-030-64984-5_27

Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *The Journal of Machine Learning Research*, *13*, 281–305. https://dl.acm.org/doi/10.5555/2188385.2188395

Brian D. Williamson, Peter B. Gilbert, Noah R. Simon & Marco Carone (2023) A General Framework for Inference on Algorithm-Agnostic Variable Importance, Journal of the American Statistical Association, 118:543, 1645-1658, DOI: 10.1080/01621459.2021.2003200

Breiman, L. (2001). Random Forests. Machine Learning, 45, 5-32. http://dx.doi.org/10.1023/A:1010933404324

Chai, D., Wang, L., & Yang, Q. (2018). Bike flow prediction with multi-graph convolutional networks. *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. https://doi.org/10.1145/3274895.3274896

Chen, C., Li, K., Teo, S. G., Chen, G., Zou, X., Yang, X., Vijay, R. C., Feng, J., & Zeng, Z. (2018). Exploiting Spatio-Temporal Correlations with Multiple 3D Convolutional Neural Networks for Citywide Vehicle Flow Prediction. *2018 IEEE International Conference on Data Mining (ICDM)*. https://doi.org/10.1109/icdm.2018.00107

Chen, C., Li, K., Teo, S. G., Zou, X., Li, K., & Zeng, Z. (2020). Citywide Traffic Flow Prediction Based on Multiple Gated Spatio-temporal Convolutional Neural

Networks. *ACM Transactions on Knowledge Discovery from Data*, *14*(4), 1–23. https://doi.org/10.1145/3385414

Chen, J., Li, K., Li, K., Yu, P. S., & Zeng, Z. (2021). Dynamic Planning of Bicycle Stations in Dockless Public Bicycle-sharing System Using Gated Graph Neural Network. *ACM Transactions on Intelligent Systems and Technology*, *12*(2), 1–22. https://doi.org/10.1145/3446342

Chen, K., Chen, F., Lai, B., Jin, Z., Liu, Y., Li, K., Wei, L., Wang, P., Tang, Y., Huang, J., & Hua, X.-S. (2020). Dynamic Spatio-Temporal Graph-Based CNNs for Traffic Flow Prediction. *IEEE Access*, *8*, 185136–185145. https://doi.org/10.1109/access.2020.3027375

Chicco, Davide & Warrens, Matthijs & Jurman, Giuseppe. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Computer Science. 7. e623. 10.7717/peerj-cs.623.

Evgeniou, T., & Pontil, M. (2001). Support Vector Machines: Theory and applications. In *Springer eBooks* (pp. 249–257). https://doi.org/10.1007/3-540-44673-7_12

Faghih-Imani, A., & Eluru, N. (2016). Incorporating the impact of spatio-temporal interactions on bicycle sharing system demand: A case study of New York CitiBike system. *Journal of Transport Geography*, *54*, 218–227. https://doi.org/10.1016/j.jtrangeo.2016.06.008

Hasan Phudinawala, & Mayank Jha. (2022). Prediction of demand for shared bikes using multiple linear regression model. International Journal of Advanced Research in Science, Communication and Technology, 481–486. https://doi.org/10.48175/ijarsct-2577

Jaber, A.; Abu Baker, L.; Csonka, B. The Influence of Public Transportation Stops on Bike-Sharing Destination Trips: Spatial Analysis of Budapest City. Future Transp. 2022, 2, 688-697. https://doi.org/10.3390/futuretransp2030038

Li, A., Zhao, P., Haitao, H., Mansourian, A., & Axhausen, K. W. (2021). How did micro-mobility change in response to COVID-19 pandemic? A case study based on spatial-temporal-semantic analytics. *Computers, Environment and Urban Systems*, *90*, 101703. https://doi.org/10.1016/j.compenvurbsys.2021.101703

Luo, J., Zhou, D., Han, Z., Xiao, G., & Tan, Y. (2021). Predicting Travel Demand of a Docked Bikesharing System Based on LSGC-LSTM Networks. *IEEE Access*, *9*, 92189–92203. https://doi.org/10.1109/access.2021.3062778

Ma, Y., Qin, X., Xu, J., & Wang, W. (2016). A hierarchical public bicycle dispatching policy for dynamic demand. *2016 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*. https://doi.org/10.1109/soli.2016.7551680

Mavoa, S., Eagleson, S., Badland, H. M., Gunn, L., Boulange, C., Stewart, J., & Giles-Corti, B. (2018). Identifying appropriate land-use mix measures for use in a national walkability index. *Journal of Transport and Land Use*, *11*(1). https://doi.org/10.5198/jtlu.2018.1132

Pedregosa, et al. (2011) Scikit-Learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

Popescu, Balas, V. E., Perescu-Popescu, L., & Mastorakis, N. E. (2009). Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems Archive*, *8*(7), 579–588. https://doi.org/10.5555/1639537.1639542

Scarselli, F., Gori, M., Ah Chung Tsoi, Hagenbuchner, M., & Monfardini, G. (2009). The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80. https://doi.org/10.1109/tnn.2008.2005605

Shen, Y., Zhang, X., & Zhao, J. (2018). Understanding the usage of dockless bike sharing in Singapore. *International Journal of Sustainable Transportation*, *12*(9), 686–700. https://doi.org/10.1080/15568318.2018.1429696

Si, H., Shi, J., Wu, G., Chen, J., & Zhao, X. (2019). Mapping the bike sharing research published from 2010 to 2018: A scientometric review. *Journal of Cleaner Production*, *213*, 415–427. https://doi.org/10.1016/j.jclepro.2018.12.157

Snehanshu Banerjee, Md. Muhib Kabir, Nashid K. Khadem, Celeste Chavis, 2020. Optimal locations for bikeshare stations: A new GIS based spatial approach, Transportation Research Interdisciplinary Perspectives, Volume 4, 100101, ISSN 2590-1982, https://doi.org/10.1016/j.trip.2020.100101.

Song, Y., Merlin, L., & Rodriguez, D. (2013). Comparing measures of urban land use mix. *Computers, Environment and Urban Systems*, *42*, 1–13. https://doi.org/10.1016/j.compenvurbsys.2013.08.001

Strobl, C., Boulesteix, AL., Zeileis, A. et al. Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC Bioinformatics 8, 25 (2007). https://doi.org/10.1186/1471-2105-8-25

Xiao, G., Cheng, Q., & Zhang, C. (2019). Detecting travel modes from smartphone-based travel surveys with continuous hidden Markov models. *International Journal of Distributed Sensor Networks*, *15*(4), 155014771984415. https://doi.org/10.1177/1550147719844156

Xiao, G., Wang, R., Zhang, C., & Ni, A. (2020). Demand prediction for a public bike sharing program based on spatio-temporal graph convolutional networks. *Multimedia Tools and Applications*, *80*(15), 22907–22925. https://doi.org/10.1007/s11042-020-08803-y

Zhang, L., & Song, J. (2022). The periodicity and initial evolution of micro-mobility systems: a case study of the docked bike-sharing system in New York City, USA. *European Transport Research Review*, *14*(1). https://doi.org/10.1186/s12544-022-00549-y

Zhang, L., Zhang, J., Duan, Z., & Bryde, D. (2015). Sustainable bike-sharing systems: characteristics and commonalities across cases in urban China. *Journal of Cleaner Production*, *97*, 124–133. https://doi.org/10.1016/j.jclepro.2014.04.006

Zhang, M., & Zhao, P. (2017). The impact of land-use mix on residents' travel energy consumption: New evidence from Beijing. *Transportation Research Part D: Transport and Environment*, *57*, 224–236. https://doi.org/10.1016/j.trd.2017.09.020

Zhang, Y., Thomas, T., Brussel, M., & van Maarseveen, M. (2017). Exploring the impact of built environment factors on the use of public bikes at bike stations: Case study in Zhongshan, China. *Journal of Transport Geography*, *58*, 59–70. https://doi.org/10.1016/j.jtrangeo.2016.11.014

Zhu, Y., & Diao, M. (2019). Understanding the spatiotemporal patterns of public bicycle usage: A case study of Hangzhou, China. *International Journal of Sustainable Transportation*, 1–14. https://doi.org/10.1080/15568318.2018.1538400

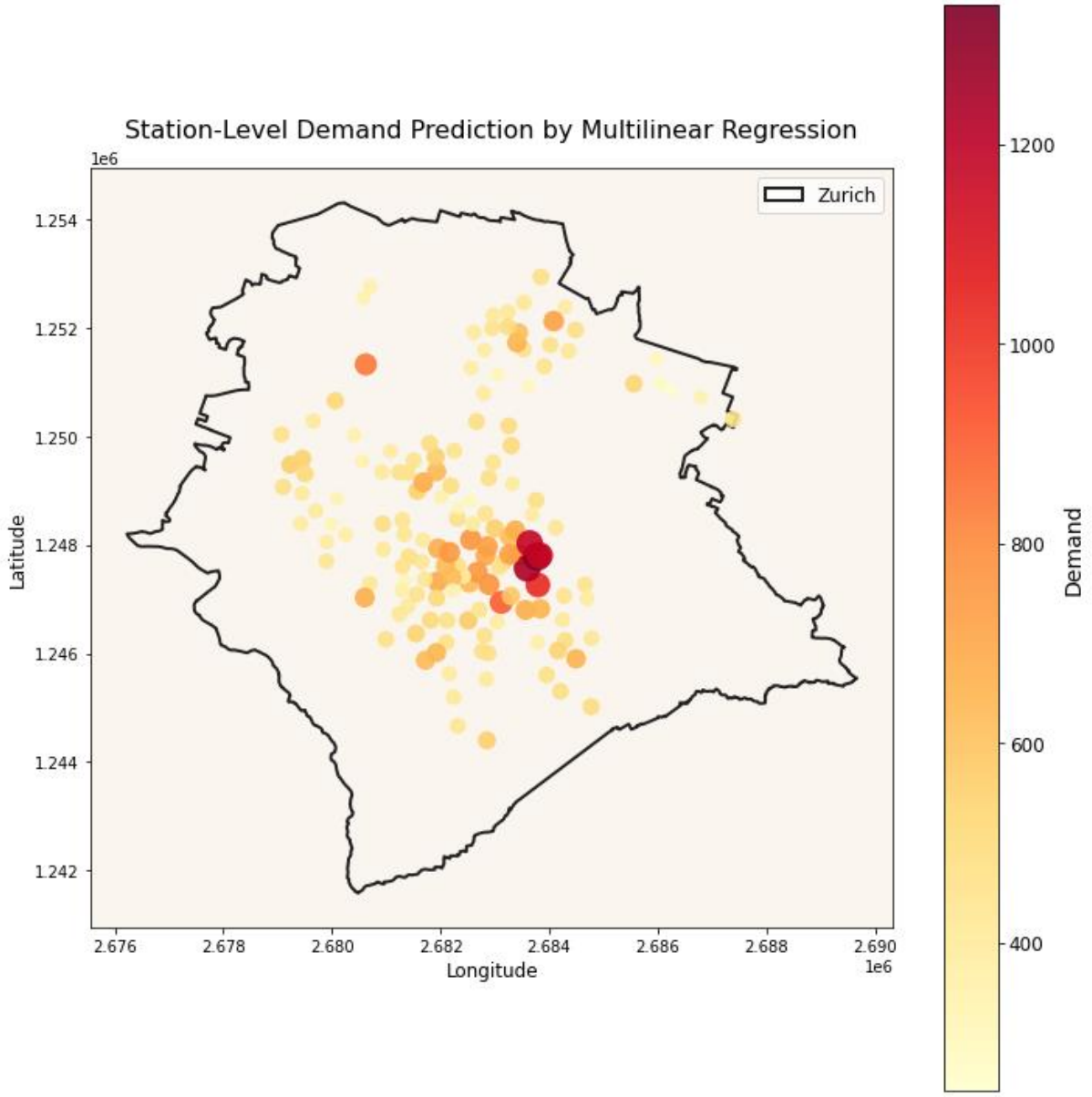
Zhu D, Zhang F, Wang S, Wang Y, Cheng X, Huang Z, Liu Y (2020). Understanding place characteristics in geographic contexts through graph convolutional neural networks. Annals of the American Association of Geographers, 110:2, 408-420. https://doi.org/10.1080/24694452.2019.1694403

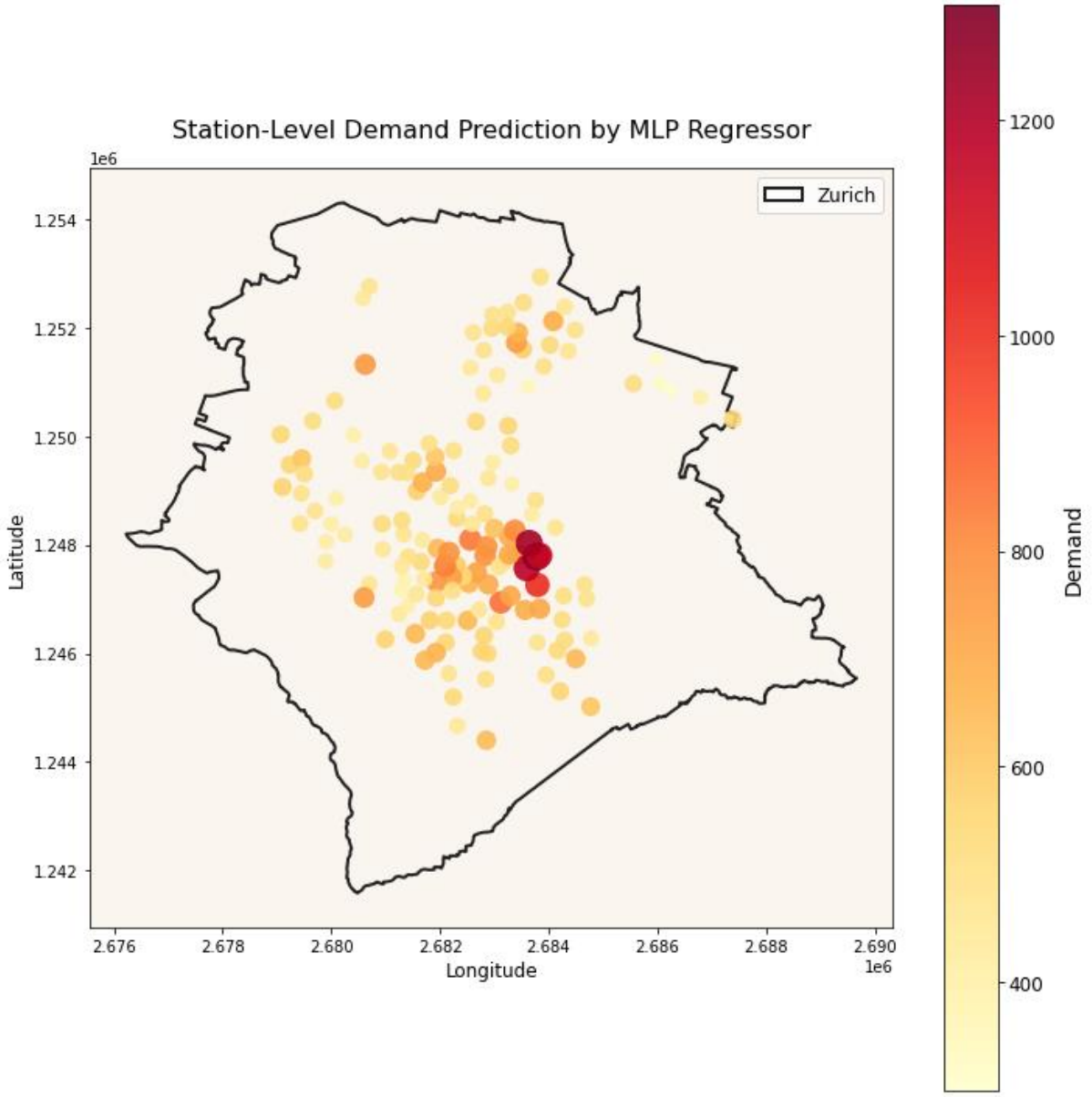
Zhu, D., Liu, Y., Yao, X. *et al.* Spatial regression graph convolutional neural networks: A deep learning paradigm for spatial multivariate distributions. *Geoinformatica* **26**, 645–676 (2022). https://doi.org/10.1007/s10707-021-00454-x

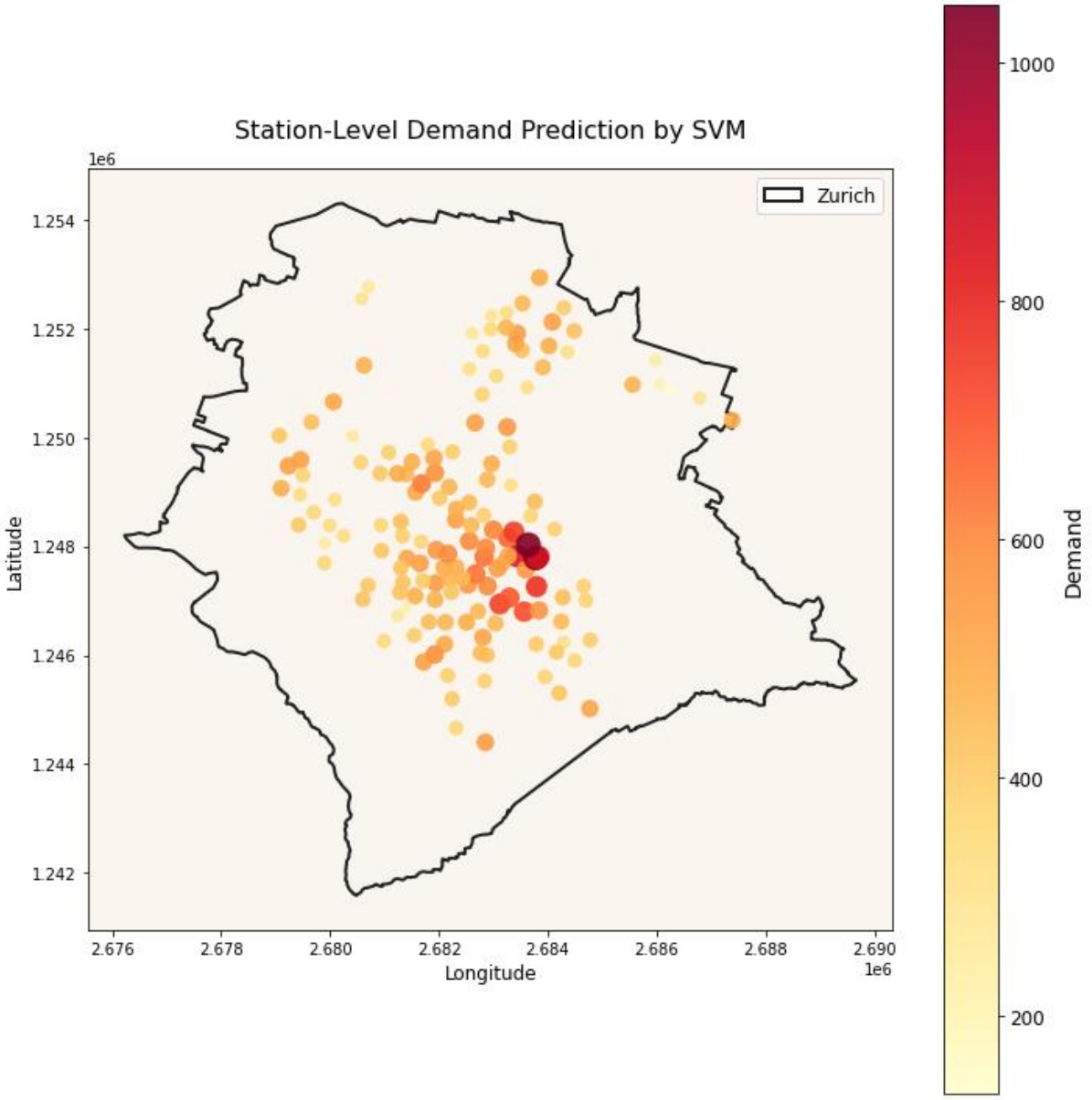
Zong, F., Tian, Y., He, Y., Tang, J., & Lv, J. (2019). Trip destination prediction based on multi-day GPS data. *Physica A: Statistical Mechanics and Its Applications*, *515*, 258–269. https://doi.org/10.1016/j.physa.2018.09.090

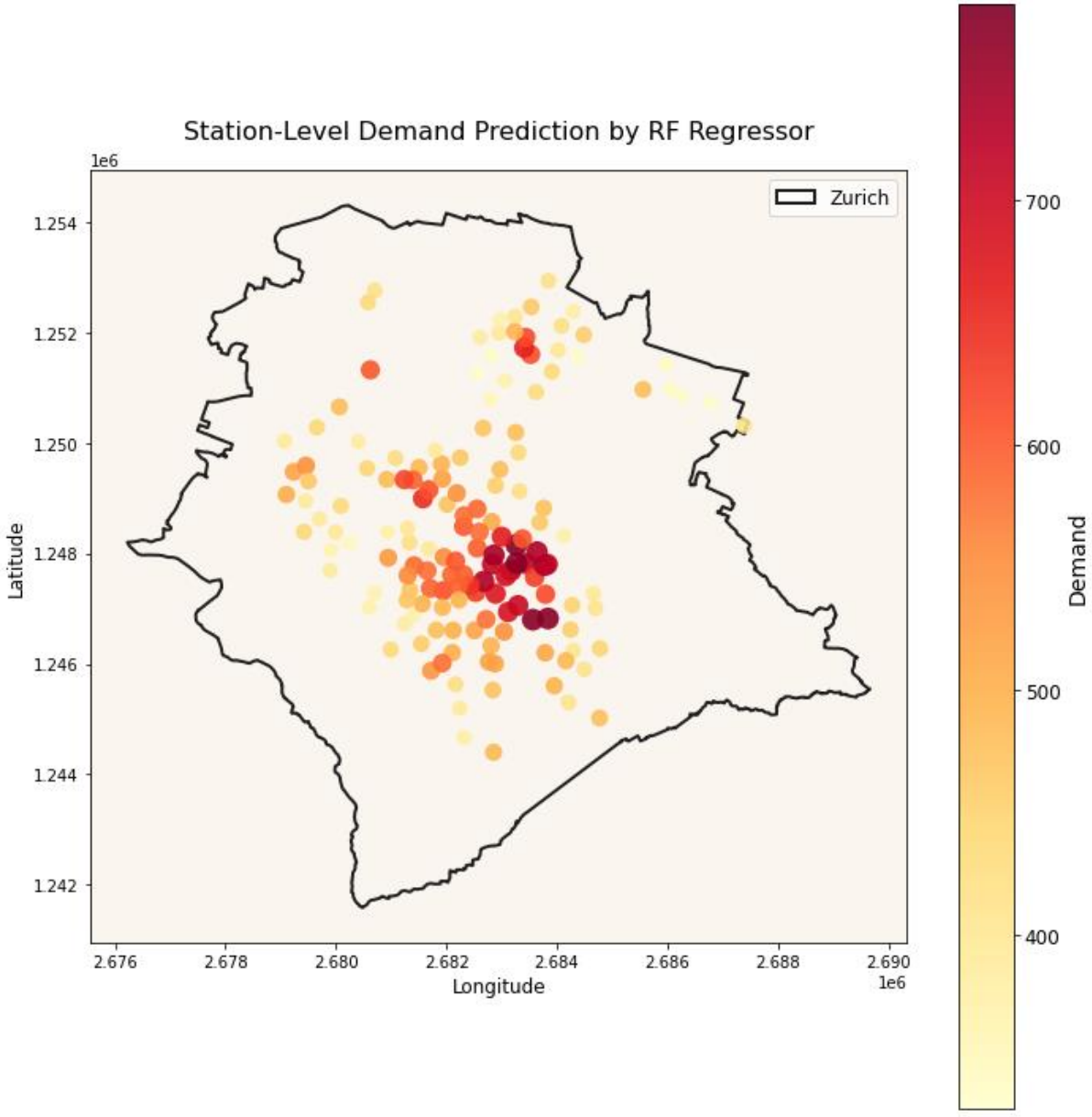
Appendix A

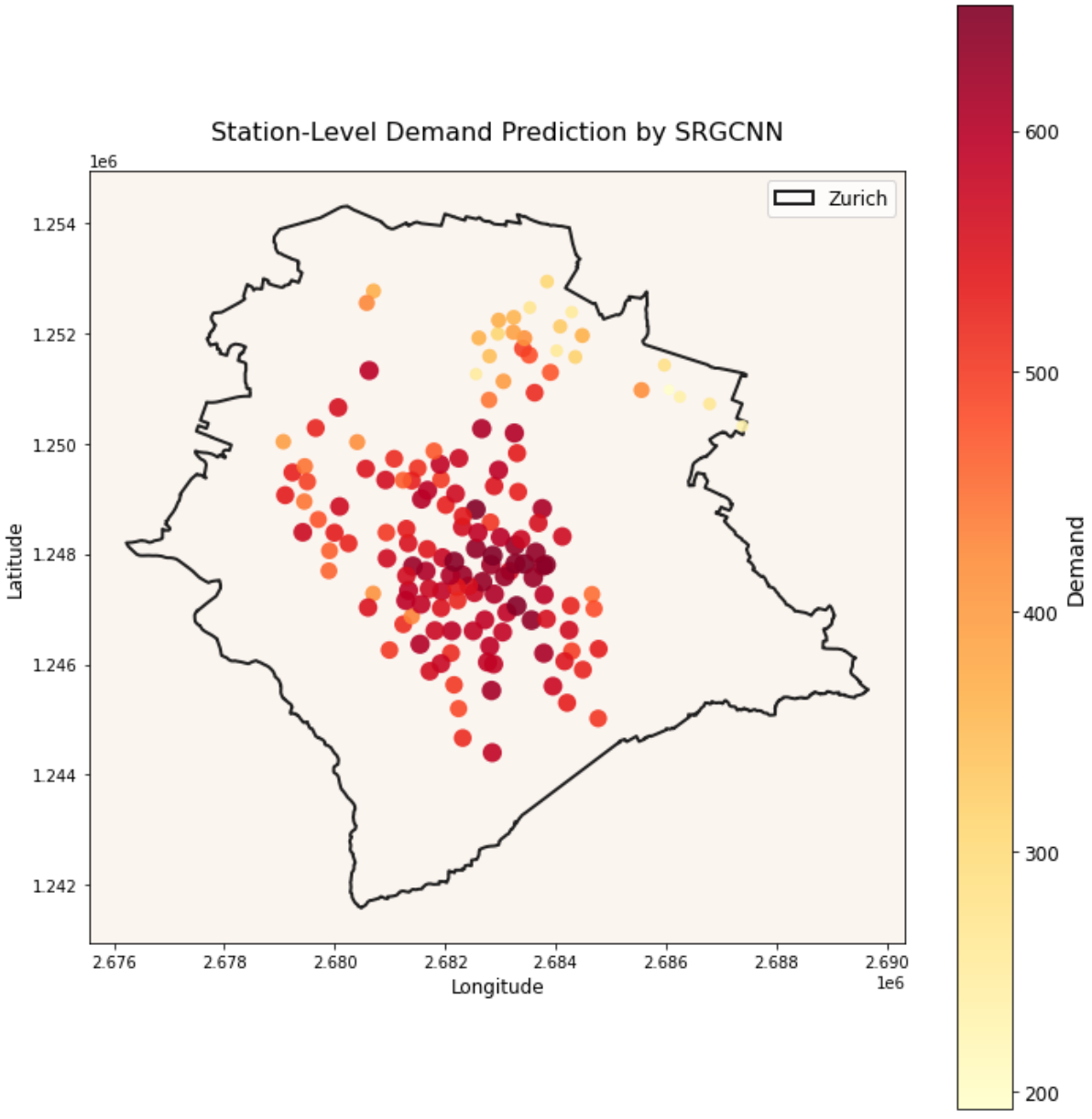
Maps of the Spatial Distribution of PBS station-level demand prediction values in Zurich. The ML and DL models used to implement the predictions are Multilinear regression, MLP regressor, SVM, RF Regressor, SRGCNN, and SRGCNN-GW.

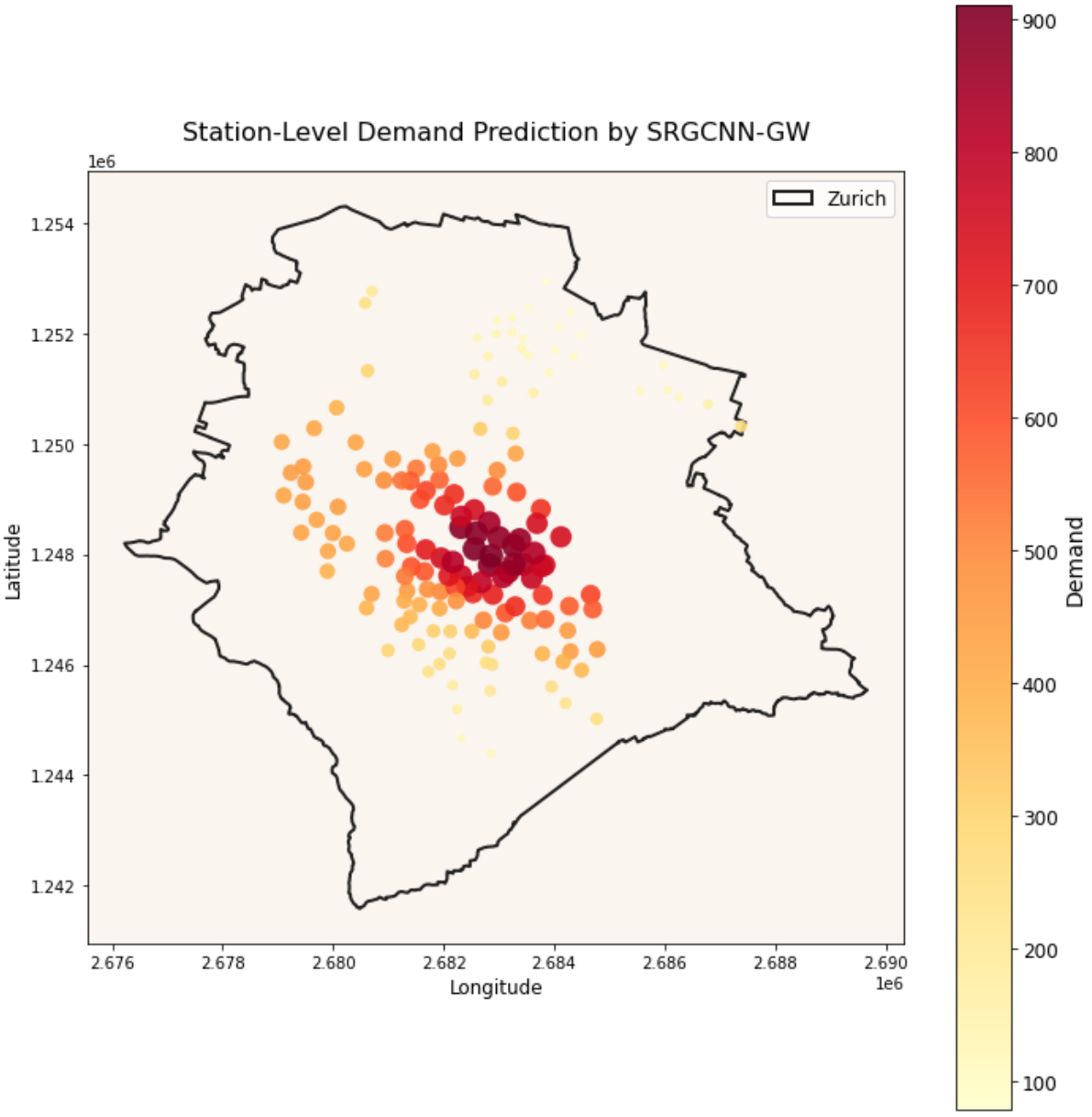



Station-Level Demand Prediction by MLP Regressor
Zurich
1e6
1.254
1.252
1.250
1.248
1.246
1.244
1.242
Latitude
2.676
2.678
2.680
2.682
2.684
2.686
2.688
2.690
1e6
Longitude
1200
1000
800
600
400
Demand


Station-Level Demand Prediction by SVM
Zurich
1e6
1.254
1.252
1.250
1.248
1.246
1.244
1.242
Latitude
2.676
2.678
2.680
2.682
2.684
2.686
2.688
2.690
1e6
Longitude
1000
800
600
400
200
Demand

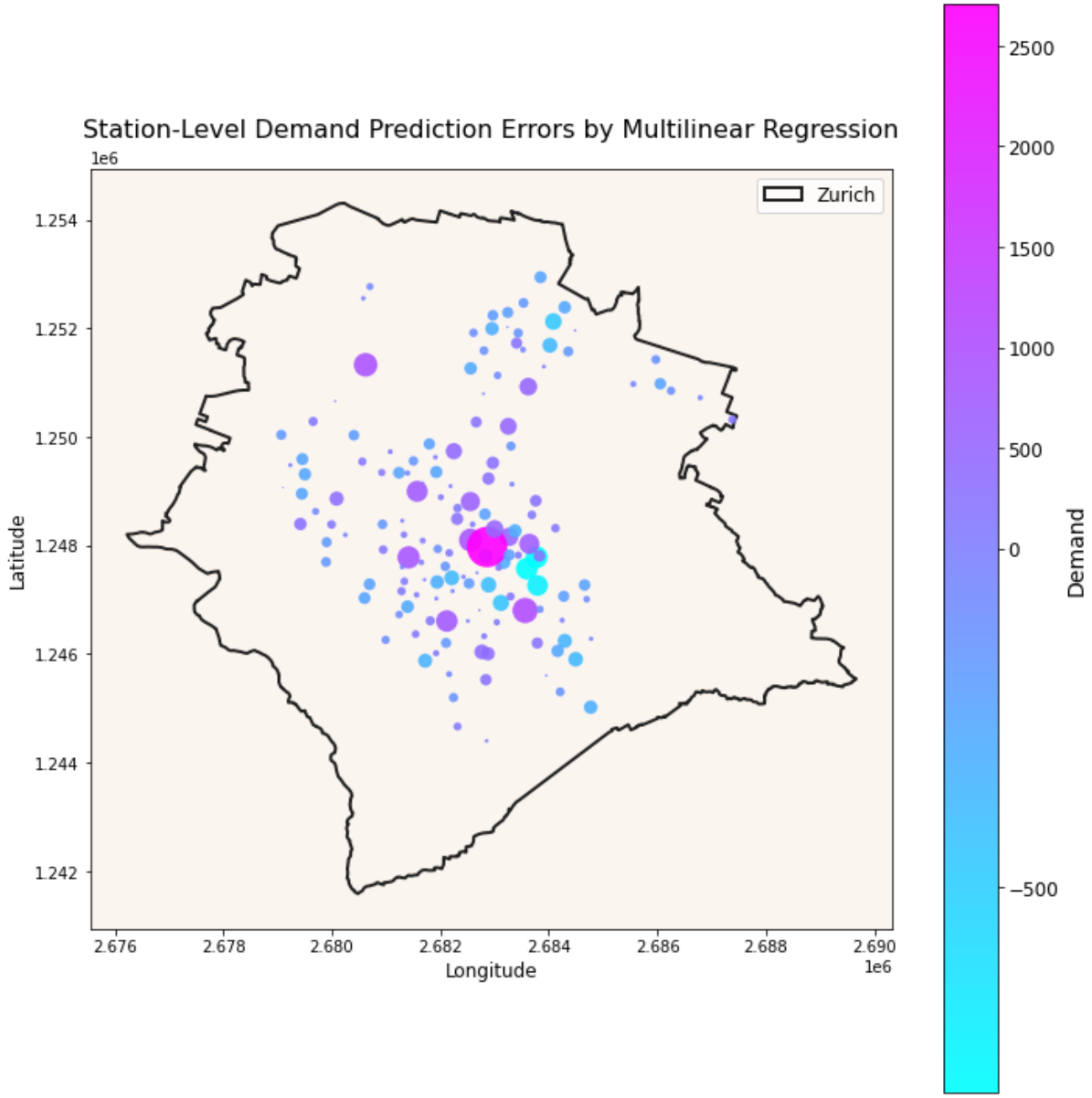

Station-Level Demand Prediction by RF Regressor
Zurich
1e6
1.254
1.252
1.250
1.248
1.246
1.244
1.242
Latitude
2.676
2.678
2.680
2.682
2.684
2.686
2.688
2.690
1e6
Longitude
700
600
500
400
Demand

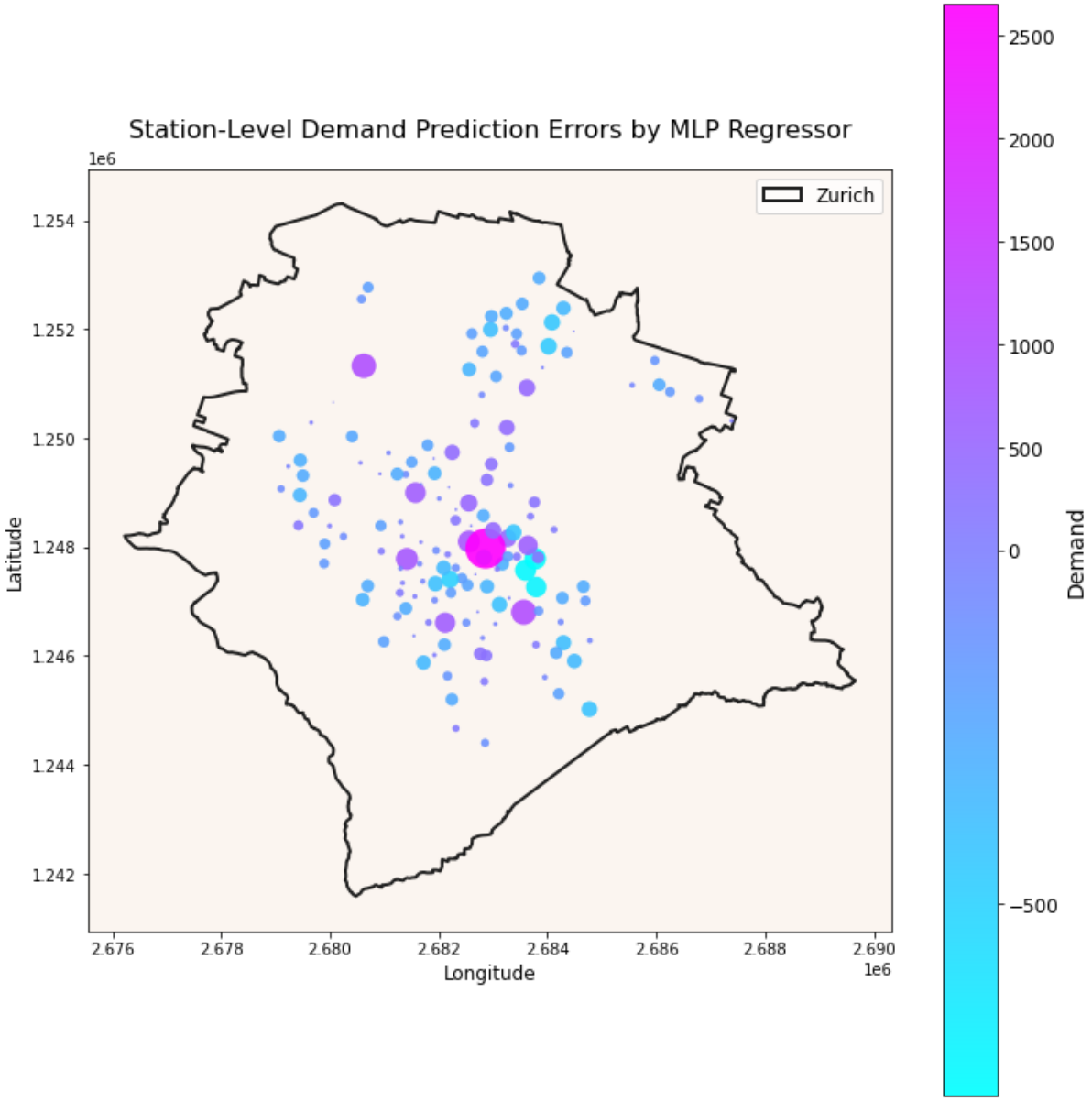

Station-Level Demand Prediction by SRGCNN
Zurich
1e6
1.254
1.252
1.250
1.248
1.246
1.244
1.242
Latitude
2.676
2.678
2.680
2.682
2.684
2.686
2.688
2.690
1e6
Longitude
600
500
400
300
200
Demand

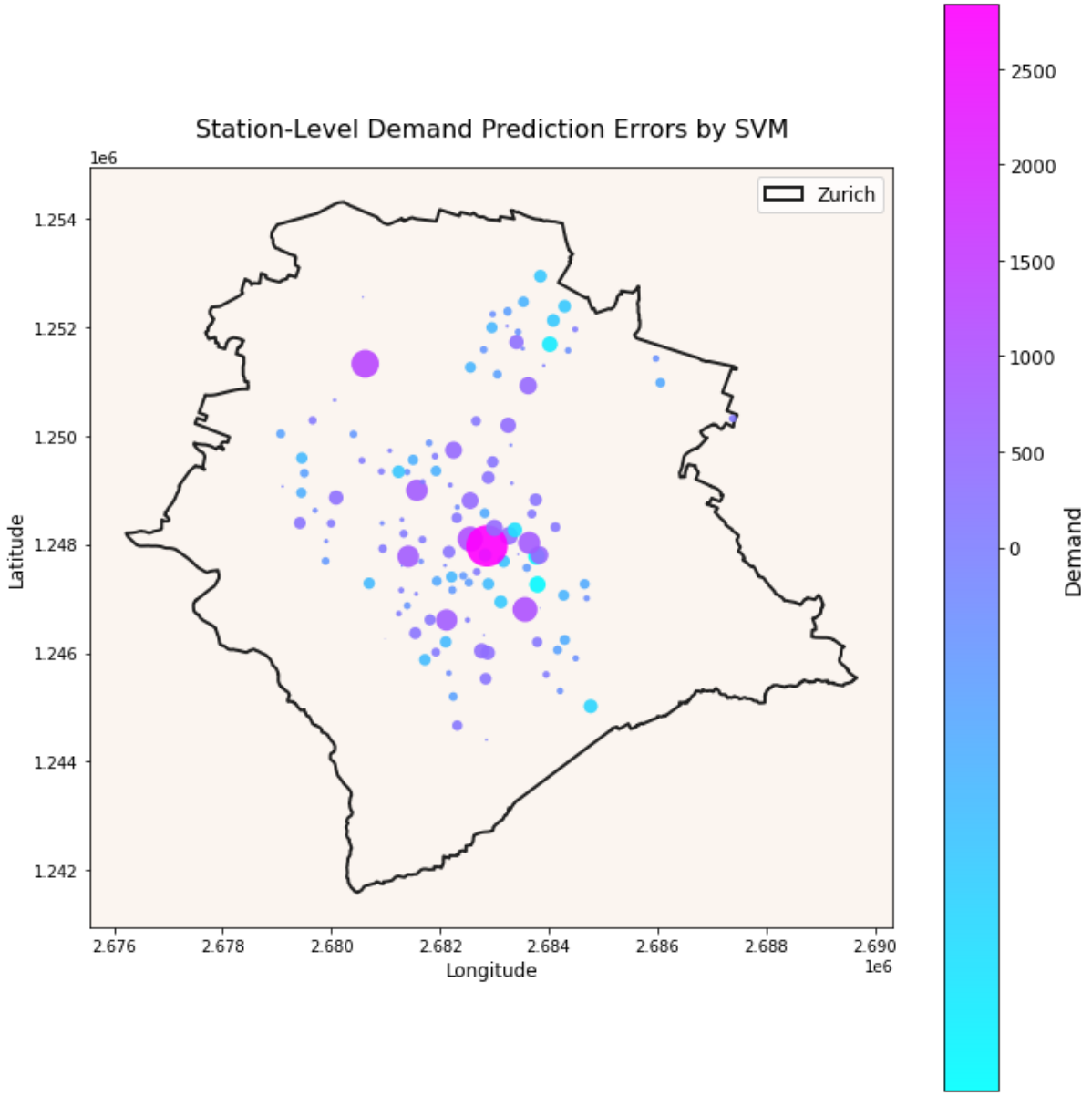

Station-Level Demand Prediction by SRGCNN-GW
Zurich
1e6
1.254
1.252
1.250
1.248
1.246
1.244
1.242
Latitude
2.676
2.678
2.680
2.682
2.684
2.686
2.688
2.690
1e6
Longitude
900
800
700
600
500
400
300
200
100
Demand

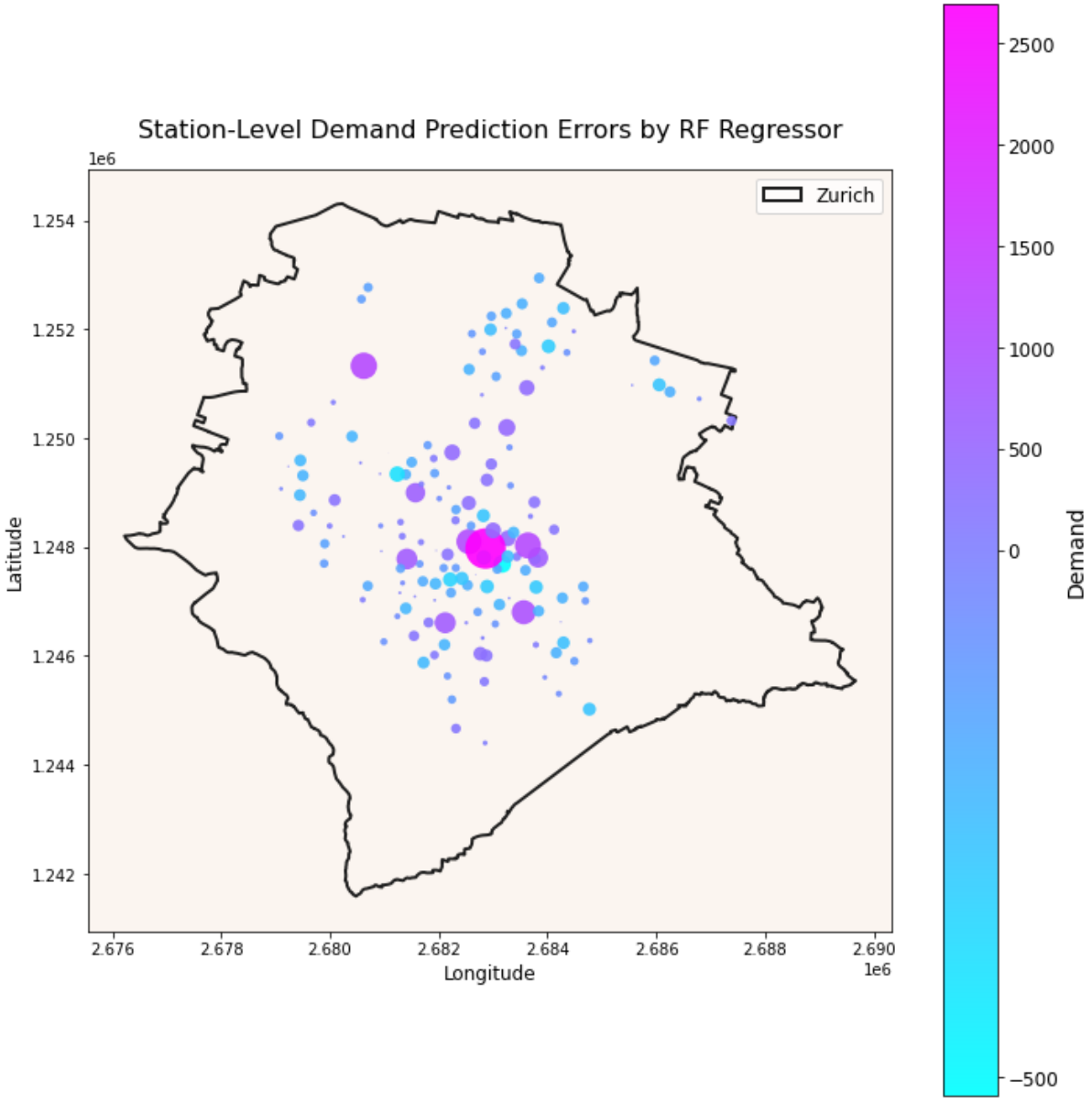
Appendix B

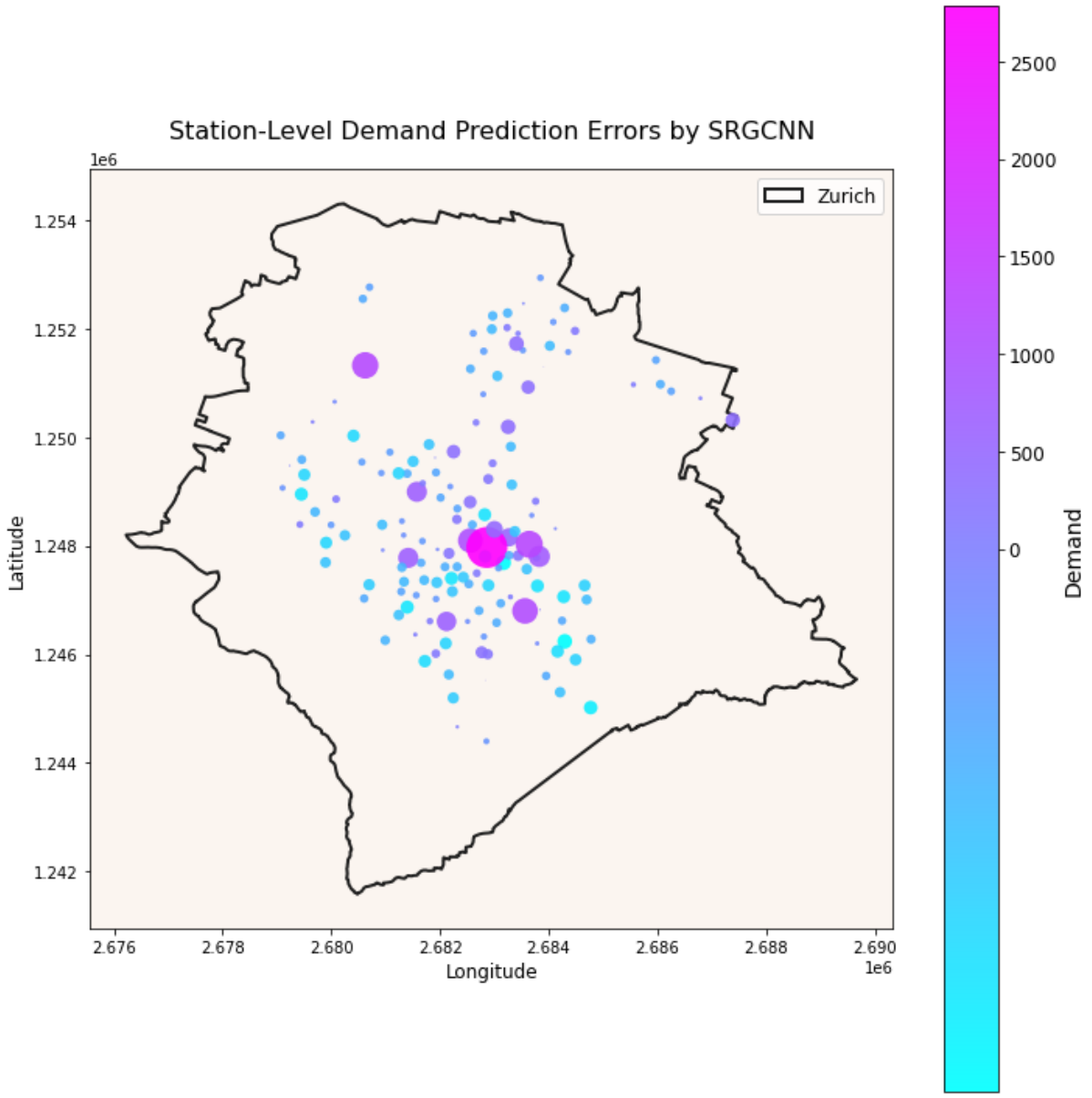
Maps of the Spatial Distribution of PBS station-level demand prediction errors in Zurich. The ML and DL models used to implement the predictions are Multilinear regression, MLP regressor, SVM, RF Regressor, SRGCNN, and SRGCNN-GW.

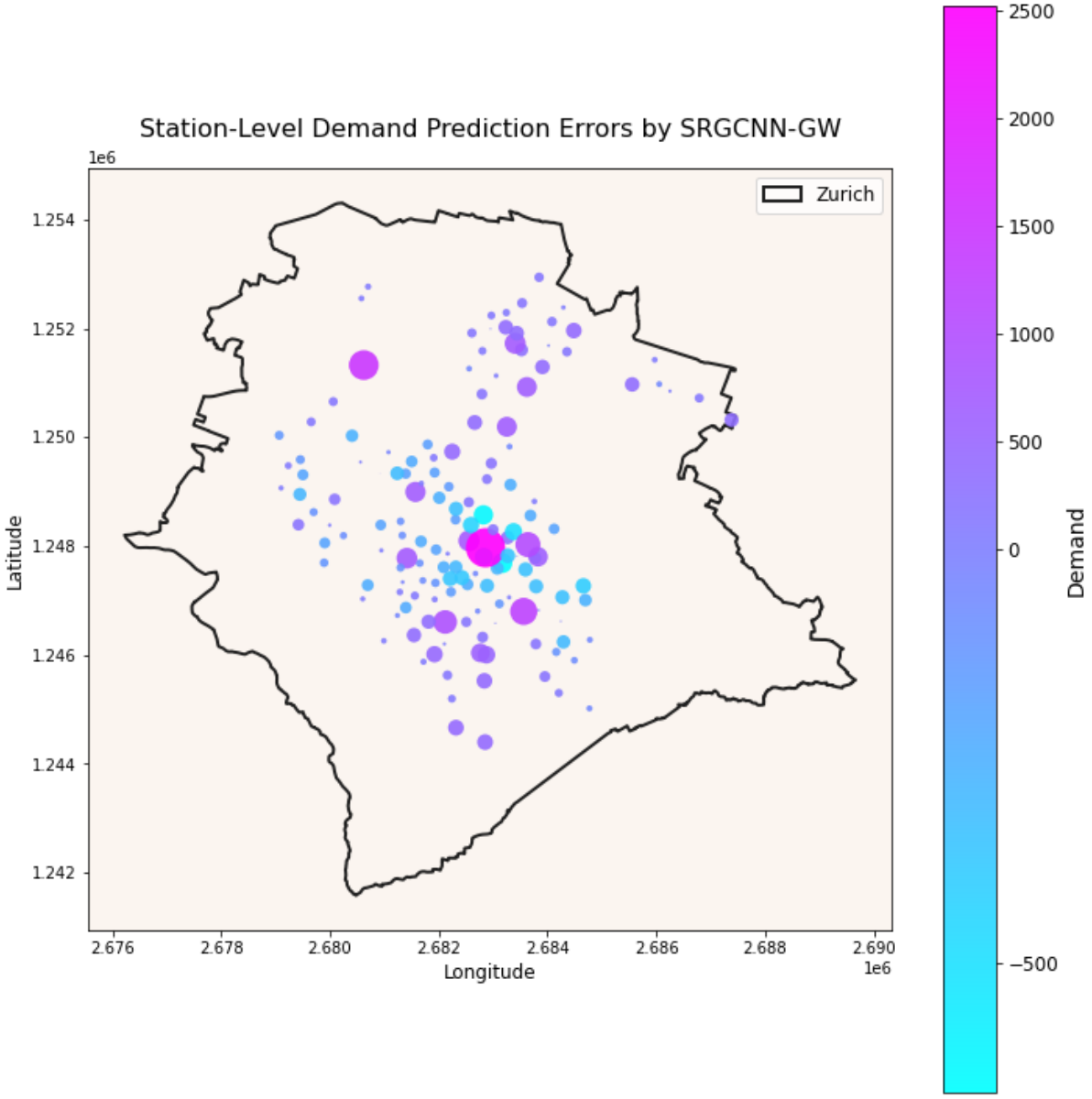



Station-Level Demand Prediction Errors by MLP Regressor
Zurich
1e6
1.254
1.252
1.250
1.248
1.246
1.244
1.242
Latitude
2.676
2.678
2.680
2.682
2.684
2.686
2.688
2.690
1e6
Longitude
2500
2000
1500
1000
500
0
−500
Demand


Station-Level Demand Prediction Errors by SVM
Zurich
1e6
1.254
1.252
1.250
1.248
1.246
1.244
1.242
Latitude
2.676
2.678
2.680
2.682
2.684
2.686
2.688
2.690
1e6
Longitude
2500
2000
1500
1000
500
0
Demand


Station-Level Demand Prediction Errors by RF Regressor
Zurich
1e6
1.254
1.252
1.250
1.248
1.246
1.244
1.242
Latitude
2.676
2.678
2.680
2.682
2.684
2.686
2.688
2.690
1e6
Longitude
2500
2000
1500
1000
500
0
−500
Demand


Station-Level Demand Prediction Errors by SRGCNN
Zurich
1e6
1.254
1.252
1.250
1.248
1.246
1.244
1.242
Latitude
2.676
2.678
2.680
2.682
2.684
2.686
2.688
2.690
1e6
Longitude
2500
2000
1500
1000
500
0
Demand


Station-Level Demand Prediction Errors by SRGCNN-GW
Zurich
1e6
1.254
1.252
1.250
1.248
1.246
1.244
1.242
Latitude
2.676
2.678
2.680
2.682
2.684
2.686
2.688
2.690
1e6
Longitude
2500
2000
1500
1000
500
0
−500
Demand

Appendix C

Supplementary Table 1. Evaluation metrics for validation data for ML and DL models

	Metrics							
	RMSE		MAE		MAPE(%)		R-squared	
Models	Demand	Demand excl. Outlier	Demand	Demand excl. Outlier	Demand	Demand excl. Outlier	Demand	Demand excl. Outlier
SRGCNN	573	253	298	178	52	41	0.58	0.88
SRGCNN-GW (k=10)	622	502	494	412	96	92	0.50	0.51
SRGCNN-GW (k=35)	539	307	312	223	55	51	0.62	0.82
Random Forest Regressor	566	288	306	205	51	44	0.12	0.06
Multilayer Perceptron Regressor	585	288	350	227	67	57	0.06	0.07
Support Vector Machine	598	278	311	186	48	33	0.02	0.13
Multiple Linear Regression	595	285	340	218	59	50	0.03	0.08

Appendix D

The table shows the factors influencing the public bike-sharing station-level demand prediction. The influencing factors are the independent variable used for the analysis and station-level demand prediction at a later stage. A description for each variable is provided. Also, the variables are divided into the following categories: statistics, POIs, land use, and hub distance.

Variables	Description	Category
Population density	Population per 100m x 100m	Statistics
Employment density	Employees per 100m x 100m	
Bike lane density	Total length within buffer zone	Points of interest (roads)
Public Transport stops	Total number within buffer	Points of interest
Closest distance to bus stops	Euclidean distance from PBS station to bus stop	Hub distance
Closest distance to tram stops	Euclidean distance from PBS station to tram stop	
Closest distance to railway stops	Euclidean distance from PBS station to railway stop	
Land use mixture	Entropy (residential, commercial, industrial)	Land use
Education density	university, library	Points of interest
Tourism density	archeological, attraction, castle, memorial, monument, museum, tourist info	Points of interest
Healthcare density	clinic, dentist, doctors, nursing home	Points of interest
Entertainment density	arts center, bar, biergarten, café, cinema, nightclub, pub, theater	Points of interest
Sports density	sports center, swimming pool	Points of interest
Accommodation density	guesthouse, hostel, hotel	Points of interest
Errand density	ATM, bakery, bank, beauty, bike rental, bike shop, book, butcher, car wash, clothes, community centre, computer, convenience, fast food, florist, food court, furniture, garden centre, gift, greengrocer, hairdresser, jeweller, kiosk, laundry, mall, marketplace, mobile phone, optician, outdoor, post box, post office, shoe, recycling, sports, pharmacy, post office, shoe, sports shop, supermarket, toy, video	Points of interest
Dining	Restaurants	Points of interest

Appendix E

Supplementary Table 2. Hyperparameters used for Multilayer Perceptron Regressor

MLP Regressor – Hyperparameters	
solver	adam
learning rate	constant
hidden layer sizes	(200,200,200)
alpha	0.1
activation	identity

Supplementary Table 3. Hyperparameters used for Support Vector Machine

SVM - Hyperparameters	
kernel	rbf
gamma	0.01
epsilon	0.001
C	1000

Supplementary Table 4. Hyperparameters used for Random Forest Regressor

RF Regressor - Hyperparameters	
Number of estimators	100
min samples split	2
min samples leaf	10
max features	sqrt
max depth	20
bootstrap	True

Supplementary Table 5. Hyperparameters used for Spatial Regression Graph Convulotional Neural Network method

SRGCNN - Hyperparameters	
Number of labels	1
Number of features	16
epochs	300
learning rate	0.01
hidden	8*Number of features
dropouts	0.3

Supplementary Table 6. Hyperparameters used for Spatial Regression Graph Convulotional Neural Network – Geographically Weighted method

SRGCNNGW - Hyperparameters	
Number of labels	1
Number of features	16
epochs	300
learning rate	0.001
hidden	32*Number of features
dropouts	0.1
k1	10
k2	35