Parking availability prediction in the Seattle city using spatio-temporal features

MSc Research Project
Data Analytics

Tejas Sanjay Shinde
Student ID: 18180159

School of Computing
National College of Ireland

Supervisor: Mr. Hicham Rifai

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Tejas Sanjay Shinde
Student ID:	18180159
Programme:	Data Analytics
Year:	2020
Module:	MSc Research Project
Supervisor:	Mr. Hicham Rifai
Submission Due Date:	28/09/2020
Project Title:	Parking availability prediction in the Seattle city using spatio-temporal features
Word Count:	7500
Page Count:	23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	27th September 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Parking availability prediction in the Seattle city using spatio-temporal features

Tejas Sanjay Shinde
18180159

Abstract

Management of parking systems is a challenge considering the substantial growth of parking demand and the restricted capacity of the cities, which further leads to trouble locating a suitable parking. A solution to this would be an accurate parking prediction system. Researchers have presented numerous approaches to predict the availability using limited features. However, none of the studies in the literature have considered the exogenous spatial factors and the Park & Ride systems which are widely used by the commuters for switching to the public mode of transportation. To address this gap, this study unfolds a novel approach in predicting the availability using spatial factors such as the walking distances from the closest public transport stations and the Seattle financial centre along with the temporal factors including weather. Machine learning models such as Back-Propagation Neural Network(BPNN), Random Forest(RF), and Extreme Gradient Boosting(XGBoost) are used in the presented study. Their performance is then optimized using GridSearchCV and assessed using R^2, RMSE, and MAE. The XG-Boost accomplished the highest R^2 of 97.65% with an error close to 0. This can help the commuters in accurately gauging the availability in advance. Also, the insights from this study will assist the Seattle Department of Transport in the management of parking demand.

1 Introduction

Car parking is an important challenge in front of all the major cities around the world. Due to the rapid growth of car ownership and the parking demand, locating a suitable spot has become troublesome. This leads to traffic congestion as highlighted by (Walaa et al.; 2017). It also affects the environment, as cars emit more pollutants when at the slow speed (Yu et al.; 2015). Therefore, an effective parking management system is a requisite. To meet the increasing parking demand different types of public and private parking facilities are made available for the people. Also, the major cities like Seattle (Zhao et al.; 2019) and Bath (Clayton et al.; 2014) have adapted park & ride(P&R) systems which make a provision for the commuters to park vehicles and change the mode of transportation to public. However, due to the limited parking capacity of the cities, management of the parking demand has become challenging. To address these issues major cities have availed "Smart City" schemes with an inclusion of the Intelligent Transport Systems(ITS) and Parking Guidance Systems(PGS). Transport authorities have developed several systems that allow commuters to pre-book parking spaces and check the current availability. However, such pre-booking options are expensive and inflexible. Also, the availability count by such systems is real-time and dynamic which fills up rapidly. Hence, when a journey is to be pre-planned such systems often mislead. To address these problems there is a need of a parking availability prediction system that can assist the drivers in gauging the possible availability in advance.

Numerous researchers have implemented various approaches to gauge the parking demand. A majority of those have used simply the temporal features such as weekend status, weather, hour, and day of the week. Whereas, some of them have used additional spatial variables such as pedestrian volume, traffic volume, distance from the nearby parking spots, and so on. However, none of these studies have considered the interspace between the public transport stations and the financial centre of the city. As presented by (Zhao et al.; 2019), in the Seattle city a majority of the commuters use public parking or P&R systems to park vehicles and switch their mode to the public transportation, which then affects the availability count. Therefore, such factors must be considered. Another key factor missed by these studies is the distance from the city financial center. As highlighted by the Seattle Department of Transportation (SDOT), nearly 28% of the office commuters use personal vehicles, whereas 9% use carpool. This adds to the parking demand. Therefore, considering the distance from the city financial center is critical as most of the workplaces reside near the financial hub of the city. None of these important factors are considered in the literature. Also, as highlighted by (Saharan et al.; 2020), the researchers have commonly emphasized only on the hourly predictions and therefore, they suggest the need for prediction in 15-minute intervals.

All of these areas of potential improvements observed in the literature are addressed in the presented study. This study presents the Seattle city parking demand prediction system. Here, spatial features such as the distances from the nearest bus station, rail station, water taxi station, and international airport are considered along with temporal features such as the day of the week, hour, and minute. Additionally, weather factors(temperature, wind, and precipitation) are also considered that have an impact on the commuter's decisions of driving personal vehicles as identified in the literature. Parking specific spatial features such as the side of the street and parking limit category are included as well. In addition to this, to address the future work suggestion by (Saharan et al.; 2020), the availability is predicted in the intervals of 15 minutes. Here, machine learning models such as RF, XGBoost, and BPNN are implemented with the Grid Search Cross Validation (GridSearchCV) optimization. Also, to compute the distances from the nearby public transport stations and city centre Haversine algorithm is utilized, which is explained in the pseudocode [1]. Overall, the study aims addressing the below research question:

"To what extent can the optimized machine learning techniques assist in gauging the Seattle city parking availability using spatial features such as the distance from the public transport stations(Bus, Rail, Water Taxi, and Airport) and financial center, etc. along with the temporal features such as hour, minute, weather, and day of the week?"

The objectives associated with this research question are as follows:

- Exploring the existing implementations from the literature to assist in the presented research
- Pre-processing and joining different data sources with Seattle city parking data using Scala and Python programming languages

- Transformation of the categorical features using one-hot encoding and feature scaling using min-max normalization
- Implementation of machine learning models such as RF, XGBoost and BPNN
- Performance optimization and cross validation of these models using GridSearchCV technique.
- Identification of the importance of the proposed feature set in the parking availability prediction
- Evaluation of these models using RMSE, MAE and R^2

The contribution of the presented study is threefold. At first, it will help the local commuters in gauging the availability and planing the journey in advance. Secondly, it will also assist the SDOT in effective management of the parking locations. This study presents a novel approach in parking availability prediction using exogenous spatial features which are not considered in the existing literature. Therefore, the insights of this study can assist the fellow researchers in future explorations in the field of ITS and PGI.

The detailed flow of the research is as follows. The Section 2 covers a critical review of related research in the field of parking availability prediction. The methodology and design followed in the presented study is explained in the Sections 3 and 4 respectively. The application of the machine learning techniques is depicted in the Section 5. The Section 6 shows the evaluation of the experiments conducted in the study followed by the conclusion explained in Section 7

2 Related Work

This section covers a review of related work conducted by the fellow researchers since the year 2015 in the field of smart city parking systems. All the related research studies and their strategies are systematically documented and critically reviewed, to acquire the necessary domain knowledge and assistance in the presented research study. It is found that a majority of the researchers have considered only the temporal characteristics for the parking availability prediction. However, there are a few studies that considered spatial variables as well. Approaches of both these types are reviewed precisely in the following sections.

2.0.1 Review of previous research based on Temporal features

The researches that forecast parking availability utilizing only the temporal and time-series variables are summarised in the Table 1.

(Zhao and Zhang; 2020) performed a comparative study, where algorithms such as Support Vector Machines(SVM), ARIMA, Linear Regression(LR), and BPNN are applied on feature such as day of the week, week status, day and time. For the selection of best parameters, the authors have used a cross-validation approach. Overall, the study shows that SVM performed well. However, limited features are used in this study and high RMSE of 8.2 is obtained. In contrast to (Zhao and Zhang; 2020), a research by (Origlia et al.; 2019) treated the noise or inconsistencies in the parking data using Kalman filters before the Support Vector Regression(SVR) model application on the similar feature set. To adjust the parameters the authors have used cross-validation techniques. However,

Table 1: Previous work using temporal features

References	Dataset	Features	Models	Best	RMSE	MAE
(Zhao and Zhang; 2020)	China	Time, Day, Weekday & Weekend	LR,SVM, BPNN & ARIMA	SVM	8.2	6.1
(Origlia et al.; 2019)	San Francisco	Time, Day of the week	SVR & Kalman-SVR	Kalman-SVR	0.92	-
(Yanxu et al.; 2015)	Melbourne & San Francisco	Time, Day of the week	BPNN, SVR & RT	RT	0.044	0.032
(Junkai et al.; 2018)	-	Time, Day of the week	FOA-(BPNN,SVR,ELM &WNN)	FOA-SVR	4.90	4.12
(Yamin Siddiqui et al.; 2020)	Los Angeles and San Francisco	Time, weather	DELM	DELM	0.32	-
(Monteiro and Ioannou; 2018)	Los Angeles	Time & travel time	Mean-Var, Normal Variance, Normal & Poisson Distribution	Poisson Distribution	-	0.10
(Yu et al.; 2015)	Xinjiekou	Time	BPNN & ARIMA	ARIMA	4.47	2.22
(Kim and Koshizuka; 2019)	Seattle	Day of the week, Minute, Hour, Day & Month	GBR, XGBoost, LightGBM & RF	RF	-	0.039
(Li et al.; 2018)	Beijing	Holiday, Time & Weather	BPNN & LSTM	LSTM	5.42	-

it acquired a high RMSE of 8.2. Also, the authors have used limited data of 2 months and for each parking segment, the models are trained individually. In both these cases, the approaches failed to acquire the desired accuracy using the temporal features and in the future, inclusion of more features and deep learning models is suggested by the researchers.

Another study by (Yanxu et al.; 2015) presents a time series based approach in forecasting parking occupancy in San Francisco and Melbourne. The authors create three different sets of input features where the first one contains current time and day of the week. The second set contains historical observations of occupancy and the final set combines both of those together. The authors have used BPNN, SVR and Regression Tree(RT) models with 5 - fold cross validation for obtaining optimal values for the parameters. Here, the RT performed significantly better than the rest of the models. In the future research, the researchers have suggest use of additional features. As opposed to the k-fold cross validation used in (Yanxu et al.; 2015), (Junkai et al.; 2018) proposed use of Fruit Fly Optimization Algorithm(FOA) technique to optimize the hyperparameters involved in training of the models such as BPNN, Wavelet Neural Network(WNN), SVR, and Extreme Learning Machines(ELM). This approach is based on behaviour of the flies in detecting an optimal location of food. Additionally, the researchers have used normalization technique based on minimum and maximum scaling. Overall, it is observed that the FOA optimised SVR performed slightly better than the rest of the three models. However, there is only a slight improvement compared to the traditional configuration and acquired high RMSE of 4.90. Also, the authors have used a very small dataset of 2 days. In the future work, the authors suggest augmentation of additional features such as weather, environment, time, week status and so on. Unlike traditional ELM in (Junkai et al.; 2018), (Yamin Siddiqui et al.; 2020) used Deep Extreme Learning (DELM) model to classify the parking in Los Angeles and San Francisco. Weather conditions are considered in addition to the time which improved the performance of the model. However, it acquired slightly higher RMSE of 0.32.

On the other hand, (Monteiro and Ioannou; 2018) evaluates four different types of approaches. At first, it uses past data and moving average of the vacancy count without taking into account the distribution of it. Second and third approaches treat its distribution as normal, whereas the later one also takes into account the time required for a vehicle to travel. In contrast to all of these approaches the fourth one treats count of vehicles parking at or leaving the parking lot as a heterogenous Poisson distribution.

Overall, from this study it can be derived that the approach based on Poisson distribution performs well, however acquires a high MAE of 10. The authors suggest use of Neural Networks or SVM models in the field of parking vacancy prediction in future work. Another research by (Yu et al.; 2015) compared ARIMA and BPNN models in forecasting parking vacancies in off street parking of malls in Xinjiekou. To adjust the parameters of the ARIMA model, the authors have used Bayesian and Akaike Information Criterions (AIC & BIC). Also the BPNN is trained using matlab software. Overall, the authors concludes that ARIMA performs well, however acquired a high RMSE of 4.47. Also, the parameters of BPNN are not optimized and only 30 days of data is used.

In contrast to the rest of the researches, (Kim and Koshizuka; 2019) forecasts parking availability using features such as day of the week, minute, hour, day, and the month. The researchers have used four models such as Gradient Boosting Regression(GBR), XGBoost, RF, and LightGBM. The author managed to reduce the MAE to 0.039 as compared to the other studies. In the future work, the author highlights inclusion of large dataset with additional features such as weather and geographic factors. Another study by (Li et al.; 2018) highlights a cloud driven approach to provide a real time predictive model which can reduce computational cost, time, and can give efficient performance. The researchers have utilized temporal variables such as holiday, time, and weather which impacted the availability. The authors managed to reduce the time required for the process. However, obtained a high RMSE of 5.42

2.0.2 Review of previous research based on Spatio-Temporal features

Unlike researches discussed in section 2.0.1, few studies along with temporal features include exogenous spatial factors. Some of them are listed in table 2.

Table 2: Previous work using spatio-temporal features

Reference	Dataset	Features	Models	Best	MAE	R2
(Saharan et al.; 2020)	Seattle	Day,Hour, Side of street, Type of area/subarea, Parking category& Block face	LR,RF, BPNN & DT	RF	0.01	0.42
(Rajabioun and Ioannou; 2015)	San Francisco	Time, Distance from nearby parking	Auto regressive	Auto regressive	-	0.95
(Yang et al.; 2019)	Pittsburgh	Time, Day, Traffic speed, Weather, & Roadway network	GCNN +RNN +MLFFD	GCNN+ RNN+ MLFFD	1.39	-
(Ghosal et al.; 2019)	San Francisco	time, type of day, parking area, block name, weather & traffic counts	CNN + LSTM + FNN + CALM	CNN + LSTM + FNN + CALM	0.878	-
(Walaa et al.; 2017)	Melbourne	Traffic volume, pedestrians count , day of the week, and time	SVR, GBR & RT	GBR	0.16	-
(Errousso et al.; 2020)	Melbourne	Time of arrival/departure, Availability count, Parking duration & Day of week	RT, BPNN, XGBoost & RF	RF	2.17	-

(Saharan et al.; 2020) predicts the parking availability and the dynamic prices for the on-street parking locations in the Seattle city. Unlike most of the studies, here parking location specific spatial factors such as side of the street, type of the area/subarea, parking category, and block face are considered along with temporal features such as day and hour. Multicollinearity is used for feature selection, along with minimum and maximum scaling. Post processing the authors used models such as LR, BPNN, RF and Decision Tree(DT). As observed the performance of RF was better than the other algorithms with an R^2 of 42%. However, a few limitations are observed such as lack of hyperparameter tuning and small data of 30 days. In addition to this, the predictions are on hourly interval. To address these limitations in the future work, the researchers suggest prediction in 15 or

30 minutes intervals, which is achieved in this study. Unlike (Yu et al.; 2015), which considered only the temporal feature with the autoregressive model, (Rajabioun and Ioannou; 2015) proposed use of distance from the nearby parking locations as well. It is observed that the distance from the nearby parking spaces has a positive impact on the availability. To adjust the parameters of the model, the authors have used both batch and recursive processing algorithms. This model acquired an high accuracy of around 95% with a cost of high Mean Absolute Percentage Error(MAPE) of around 15%.

(Yang et al.; 2019) on the other hand proposed a unique approach in predicting parking availability by analysing the role of travel demand and weather on the availability count in Pittsburgh in 30 min interval. Spatio-temporal features such as time, day, occupancy, traffic speed, weather, and roadway network are used in this study, which are scaled using min-max scaling. At first, the temporal features are extracted using Graph Convolution Neural Network(GCNN), whereas LSTM based Recurrent Neural Network(RNN) is applied on spatial features. Post this both the extracted features are combined to predict the availability using Multi-Layer Feed Forward decoder(MLFFD). However, this model acquired a high error of 1.39. In the future work, the authors expect inclusion of additional parameters as well. A similar approach is also presented by (Ghosal et al.; 2019). The researchers extract the temporal characteristic of the parking locations based on the weather condition, time and occupancy using LSTM autoencoder. Also, the spatial features are extracted from the traffic condition, traffic count and parking area using CNN. After this feed forward neural network(FNN) is used for the prediction, where its performance is enhanced using Clustering Augmented Learning Method(CALM). Overall, the model performed better than (Yang et al.; 2019), with an MAE of 0.878. In the future studies, the authors suggest inclusion of additional spatial features. Both these approaches (Yang et al.; 2019; Ghosal et al.; 2019) depict that the traffic volume has an influence on the parking availability and acquire reasonable performance. However, both are are computationally expensive considering the training time required for three neural network models.

Similar to (Yang et al.; 2019; Ghosal et al.; 2019), another research by (Walaa et al.; 2017) used traffic volume to predict the parking availability in Melbourne. The authors have also used pedestrians count , day of the week, and time. The author implemented and compared different models such as SVR, RT, and GBR. Here, the GBR performed well and acquired a comparatively low MAE of 0.16. As observed, the pedestrian count has an influence on the parking availability and helps in improving the performance. However, exogenous factors such as different transport location or business areas could be the reason for the variability in pedestrians count. Hence, such exogenous factors must be evaluated as well. Another study by (Errousso et al.; 2020) predicts availability in Melbourne using temporal features such as time or arrival/departure, availability count, parking duration, and day of the week along with spatial feature such as area and subarea are used by the researchers. Here, regression based models such as RT, BPNN, XGBoost, RF and stacking are applied on the processed data. It was observed that the RF performed better. However it acquired high MAE of 2.17. In (Richter et al.; 2014) temporal and spatial clusters are used to forecast availability classes such as high, average, or low. The authors managed to acquire an accuracy of around 70%. However, the model only predicts the class of availability, which makes it is difficult for users to gauge the actual vacant spaces. Another study by (Koumetio Tekouabou et al.; 2020) presents prediction of availability rate at Birmingham parking locations using ensemble techniques. The features used are unique parking Id, date, time ,availability proportion, parking space, and occupancy. Overall, RFR showed a significant performance. However, these results are not completely reliable as the model is trained on a very small dataset of 30 days.

2.1 Challenges

Some of the key challenges observed by the researchers while working on the parking data are listed below:

- (Yang et al.; 2019), shows that most of the parking sensors operate on specific days and timings. Also, are often prone to technical delays and issues. Both these cases lead to misreading and missing data. Therefore the authors proceed with assumptions that such parking locations are not monitored in specific period and hence predictions in that period will not be feasible.
- As highlighted by (Saharan et al.; 2020) there is an issue of double parking in Seattle on-street parking. This causes occupancy count to be more than capacity. This challenge is addressed by the authors by considering the availability as 0 where the occupancy is greater than capacity.
- Another challenge is in selecting the best parameter combination for the machine learning algorithms. This is resolved by all the authors by using cross validation techniques.
- There is also a problem with diversity in scale of input features affecting the predictions. This is resolved by authors with the help of min-max normalization.

All of these challenges and their solutions observed form the literature are considered in this study.

2.2 Conclusion

From the systematic review of the literature, it can be perceived that in all these implementations the choice of methods and datasets is different. Here, most of the methodologies include only the temporal variables such as hour, day of the week, and weather conditions. Whereas, a few have considered additional spatial features such as traffic flow, pedestrian volume, and distance from nearby parking locations, which are identified as influential in parking availability prediction. Therefore, the researchers have suggested the inclusion of more exogenous spatial factors in future studies. Also, none of these studies have considered the impact of access to the financial centers and public transportation on parking availability, which will be considered in this study. Additionally, weather is identified to have an influence on the parking availability and hence is included in the presented study as well.

In the literature Neural Networks and Ensemble Tree-based models have shown better results and are most commonly used. Therefore RF and BPNN models are used in this study. In addition to that, most of the studies have used relatively smaller datasets of below 2 months and have predicted the availability in 30 min or hourly intervals. Therefore, the researchers have suggested consideration of larger datasets and prediction in 15 min intervals, which is addressed in this study.

Overall from the literature, it can be perceived that there is a great scope for enhancements and augmentations in this field. Therefore, all of the above-mentioned improvements are carefully composed in the presented study.

3 Methodology

The complete data mining procedure in the presented study of parking availability prediction follows a set of steps based on KDD methodology. All these steps are systematically presented in Figure 1.

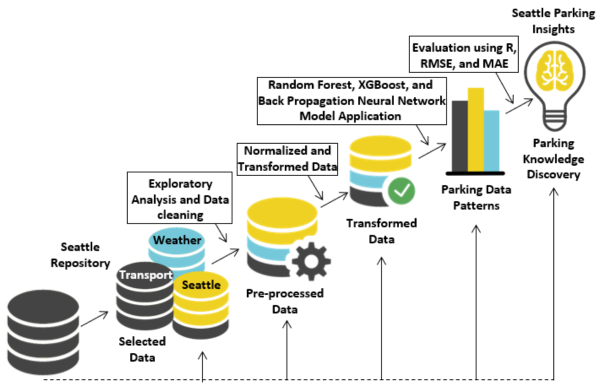

Figure 1: Parking availability prediction methodology

3.1 Data Acquisition

In the presented research the Seattle data repository is explored and data required for the study is acquired from a variety of resources, which are explained below

- Seattle City On-Street Parking Data
  At first the on-street parking availability data recorded by the sensors in the Seattle city is acquired from the Seattle Department of Transport API [1]. The minute level parking availability data of six months from July 2019 to Dec 2019 is extracted via API using Python. Around 49 unique parking road segments within 1 kilometer of the city centre are selected due to the restriction of API requests and computational complexity. Overall, the dataset consists of 43,63,631 records in total. The on-street parking road segments in the Seattle city are monitored from 8 am to 7pm and only from Monday to Saturday(excluding holidays). Therefore, the data in only this period is available.

- Seattle City Public Transport Stations
  The data about all the public transport stations in the Seattle city is acquired from the King County Transport Department website [2]. It contains four related text files such as stops, stop_times, trips and routes. When these files are combined

[1] https://data.seattle.gov/Transportation/2019-Paid-Parking-Occupancy-Year-to-date-/qktt-2bsy

[2] https://kingcounty.gov/depts/transportation/metro/travel-options/bus/app-center/developer-resources.aspx

together using respective unique identifies provide all the public transport stations in the Seattle city, their coordinates and modes. This contains around 7264 stations, which includes 50 rail stations(streetcar, rail, and trams), 7210 bus stops , and 4 water taxi(ferry) stations. In addition to this, the location coordinates of the city financial centre(Downtown Seattle) and Seattle international airport are obtained from the google maps[3].

- Seattle City Daily Weather Data
  The Seattle daily weather information from July 2019 to Dec 2019 is obtained from the National Ocean and Atmospheric Administration(NOAA) website API[4]. It consists of measures such as TMIN(Minimum temperature in F), TMAX(maximum temperature in F), PRCP(Precipitation in mm), and AWND(Wind speed in mph).

All the data is fetched using Python and then stored into the ubuntu instance running on OpenStack cloud for further processing.

3.2 Data Pre-processing and Transformation

Post the acquisition of datasets, a set of pre-processing steps are performed. The complex computation of the distances from the nearest public transport stations and city centre is performed using scala due to its parallel in memory computational ability. Whereas rest of the processing and model application is done using python. The pre-processing steps are explained in detail in the following sections:

3.2.1 Dealing with Missing Data

- Seattle City On-Street Parking Data
  Here, it is observed that the data is missing for only 31 days, which includes Sundays and holidays such as Independence Day, Labour Day, Veterans Day, Thanksgiving, and Christmas. As observed from the SDOT, the public on-street parking segments are not monitored on Sundays and holidays. Therefore, considering this as a limitation of the study the missing values are ignored.

- Seattle City Daily Weather Data
  It is observed only three days (30-Sept-2019, 5-Nov2019, and 11-Nov-2019) are missing from the six months daily data. Therefore, these records are imputed instead of ignoring them. The values are imputed with the help of interpolate function provided by Python, which follows a linear approach as shown in Figure 2.
  Both the parking and weather datasets are then combined based on 'Date' column.

3.2.2 Feature engineering

- Distance from the nearby public transport stations
  At first, the longitude and latitude of all the public transport stations in the Seattle city are acquired as described in the section 2. To identify the nearest public transport stations, at first the distances from all the public transport stations are

[3] `https://www.google.com/maps`
[4] `https://www.ncdc.noaa.gov/cdo-web/webservices/v2`

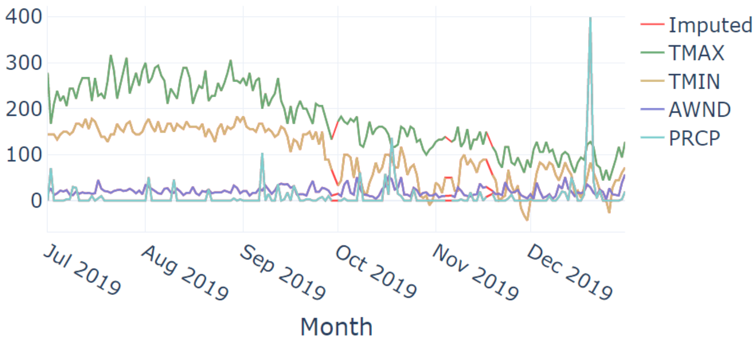

Figure 2: Imputed Missing Weather Data

calculated. From the acquired distances, the closest bus, rail(streetcar, rail, and trams) and water taxi stations are identified. Using Scala a function is written based on Haversine's distance algorithm (Winarno et al.; 2017) for this complex computation. This algorithm is explained in pseudocode [1] below. In the end, features such as NearByBus, NearByRail, and NearByFerry are created respectively.

- Distance from the Seattle city financial centre and the nearby international airport
  The distance of the parking locations from the closest international airport in the Seattle city and the city financial centre is calculated directly using the Haversine's formula[1] in Scala. Post this features called as DistCity and DistAir are created respectively.

Algorithm 1 Haversine Distance Computation

function HAVERSINEDIST($lon1, lat1, lon2, lat2$) {Longitudes and Latitudes of the two locations}
  $lon1 \leftarrow (lon1 * 3.14)/180$
  $lat1 \leftarrow (lat1 * 3.14)/180$
  $lon2 \leftarrow (lon2 * 3.14)/180$
  $lat2 \leftarrow (lat2 * 3.14)/180$ {Degree to Radian Transformation}

  $dist\_lon \leftarrow lon2 - lon1$
  $dist\_lat \leftarrow lat2 - lat1$
  $area \leftarrow sin(dist\_lat/2)^2 + cos(lat1) * cos(lat2) * sin(dist\_lon/2)^2$

  $central\_angle \leftarrow 2 * atan2(sqrt(area), sqrt(1 - area))$ {Central Angle Computation}
  $earth_radius \leftarrow 6371$
  $dist \leftarrow central\_angle * earth\_radius$ {Distance Between Two Points}
  return $dist$
end function=0

- Extraction of the features from the date column
  The features such as hour, min, and dow(day of the week) are extracted from the occupancydatetime column using regular expressions.

- Parking availability computation
  The Avail% target column, which shows the percentage availability in the road parking segments is computed as below. At first the availability is calculated using below formula:

$$Avail = ParkingSpace - ParkingOccupancy \quad (1)$$

As observed by (Saharan et al.; 2020), sometimes people double park, causing more vehicles than the actual capacity of the on-street parking. In such cases the availability is negative. To address this, it is assumed that the parking is full when the Avail is negative and hence it is replaced with 0. Post this the Avail(%) is calculated as below:

$$Avail(\%) = Avail * 100/ParkingSpace \tag{2}$$

3.2.3 Transformation

- As highlighted by (Saharan et al.; 2020), there is a need of prediction in 15-minute intervals. Therefore, the parking availability records at 15-minute intervals are only selected in this study.
- Further, the data is grouped by sourceelementkey, dow, hour, and min. As this records availability for each parking segment depending on the day of the week, hour, and minute.
- The extreme outliers in a parking dataset can affect the model performance. Hence they are removed using Z score, which removed 321 rows with Zscore over 3.

After all the processing and transformation conducted in above sections the data contains 12209 rows. The all the features in the final parking data are showed in the Figure below:

Hour	Hour of the day	**AWND**	Average Wind Speed
Min	0, 15, 30, and 45 minutes	**PRCP**	Precipitation
Day of The Week	Monday to Saturday	**NearByFerry**	Distance from ferry station
Parkingspacecount	Capacity of the parking	**NearByRail**	Distance from rail station
parkinglimit	Time limit of parking 2, 4, and 10 hour	**NearByBus**	Distance from bus station
sideofstreet	Side of the street such as SE, SW,W,NW,NE, and E	**DistCity**	Distance form city centre
TMIN	Minimum Temperature	**DistAir**	Distance from airport
TMAX	Maxium Temperature	**Avail%**	Percentage availability

Figure 3: All the features in parking data

3.2.4 Exploratory Analysis

In this section some of the key insights about the data are presented.

- Plot of Parking Availability(%) by Hour, Min and Day of the Week
  From the Figure 4.a it can be observed that there is a high availability in the morning at around 8 am. Post that, during the peak office hours the parking locations starts filling till 12pm. From 12 to 1 pm the availability remains stagnant post which it starts increasing till 4 pm followed by a down surge. Figure 4.b shows that the availability is low at 45 minutes as compared to 0 minute. Also, the Figure 4.c depicts that the occupancy of the on-street parking segments in the Seattle city is low on Mondays and gradually increases towards the end of the week. However, the effect of minute and day of the week on the availability is minimal as compared to the hour of the day.

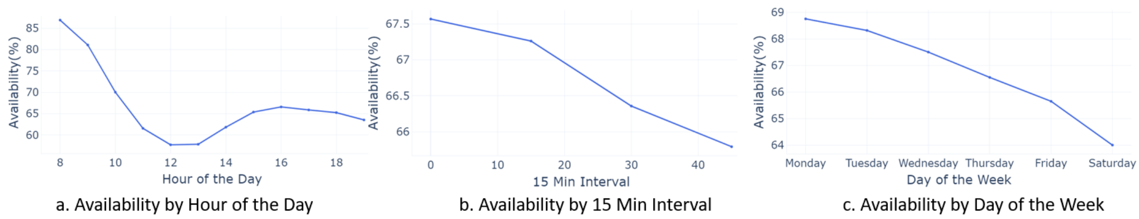

Figure 4: Parking Availability(%) by Hour, Min and Day of the Week

- Plot of Parking Availability(%) by Parking Limit and Side of the Street
  From Figure 5.a it can be observed that the parking locations with 4 hour parking limit are highly occupied as compared to the ones with 10 hour parking limit. Also, Figure 5.b depicts that the North West side parking locations are highly available as opposed to East side parking.

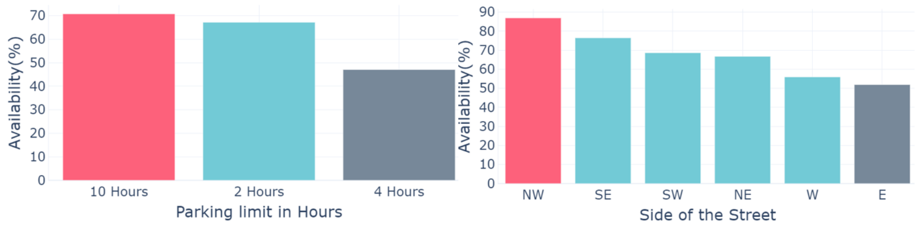

Figure 5: Parking Availability(%) by Side of Street and Parking Limit

- Correlation
  The Figure 6 presents the Spearman's correlation between the parking availability% and the transformed input features. The availability(%) is not normally distributed and hence Spearman's correlation is used instead of Pearson's. From the Figure 6 it can be observed that the distance from water taxi(ferry) stations has comparatively strong correlation(around 39%) followed by the distance from bus, rail and the city center. Also, it depicts that higher the distance from the ferry and rail stations higher the availability, whereas lower the distance from the bus stops and the city center higher the availability. In addition to this, the hour, precipitation, parking limit, and side of the street have an impact on the availability as well. Sides of the streets such as NW, SE and SW has weak positive correlation with availability, whereas West side has negative impact. Also the 2 hours parking locations are positively correlated wit availability, whereas 4 hour parking locations are observed to be negatively correlated.

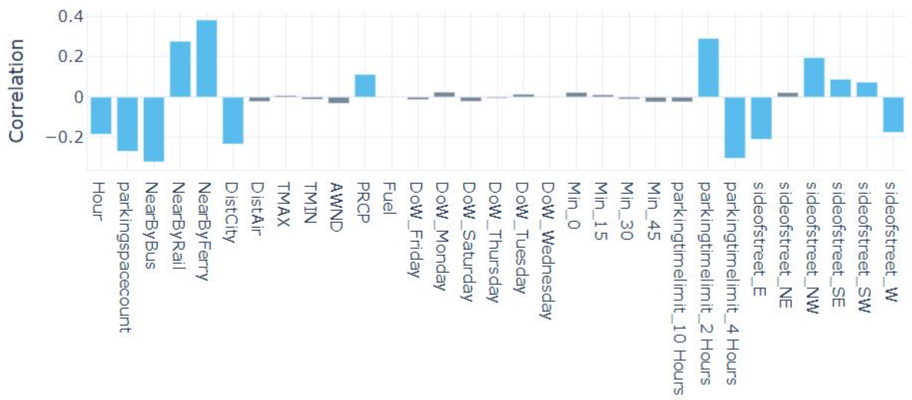

Figure 6: Correlation

3.3 One-Hot Encoding

Some of the features are categorical such as min(0, 15, 30, and 45), parkinglimit(2 hours, 4 hour, and 10 hours), dow(monday to saturday) and sideofthestreet(NW, SE, SW, NE, W, and E). However, the selected machine learning models cannot handle categorical input features directly. Therefore, One-Hot encoding is applied on the dataset, which transforms the categorical variables such as sideofstreet, min, and parkinglimit into dummy variables. This transforms each category in these variables into individual features in the form of 0's and 1's.

3.3.1 Normalization

The combined dataset contains 31 features(including dummy variables) each of a different scale. This diversity of scale in the input features often affects the performance of the machine learning models. To address this gap min-max scaling is used in the proposed study, which transforms all the features in a scale of 0 to 1. Sklearn's MinMaxScaler library[5] is used for this purpose, which works on the below formula:

$$xnew = ((x - xmin)/(xmax - xmin)) * (max - min) + min \tag{3}$$

Also, the Avail(%) target variable is transformed from the scale of 0 to 100 into a scale of 0 to 1 by dividing it by 100. E.g. 85.25% is transformed to 0.8525. This helped in treating all the variables in a same scale.

4 Design Specification

The overall architecture of the parking availability prediction system presented in this study is divided in three main phases, which are explained in detail below:

- Database Layer:
  At first, the Seattle parking related datasets of different formats such as CSV and API are acquired from different data sources using python and stored on the ubuntu instance residing on the OpenStack cloud. Post this the necessary merging, pre-processing, transformation, and feature engineering is carried out using both Scala

[5]https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

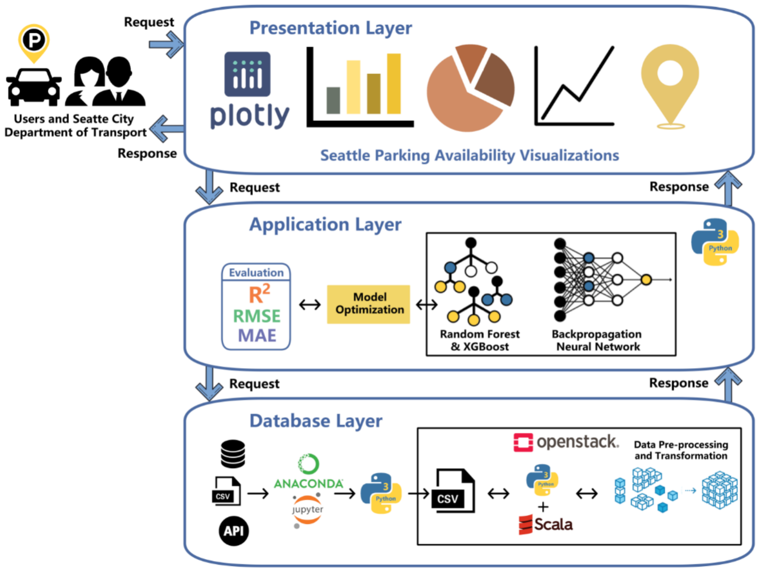

Figure 7: Design Specification

and Python. The reason for using Scala is it's speed and support for parallel processing. Whereas, Python is selected as it provides different libraries for data manipulation and machine learning. After this, the processed parking data is stored on the OpenStack instance in the CSV format.

- Application Layer:
  The processed data is then used for parking availability prediction. Machine learning models such as RF, XGBoost and BPNN are used for the prediction. The modeling and optimization of the presented models is carried out using Python Jupyter Notebook running on the OpenStack cloud. The results are evaluated using R^2, RMSE and MAE.

- Presentation Layer
  The results obtained from the model are visualized using the Plotly library available with Python. This will help the commuters in anticipating the parking availability and will also assist the Seattle Department Of Transport in the efficacious management of the parking demands.

5 Implementation

5.1 Data Preparation and Model Application

After the processing and normalization of the parking availability dataset around 12,209 rows of processed data is ready for the model application. The training capability of the model depends on the size of the data it is trained on. Therefore, the dataset is split into 75%(9156 rows) train and 25%(3053 rows) test datasets using random sampling to avoid bias. Post this the machine learning models presented in section 5.2 are applied on the train and test datasets. From the critical review of literature it is observed that considering the non-linearity in the parking dataset researchers have commonly used the ensemble tree based models and complex neural networks. In the researches conducted using spatio-temporal features RF, XGBoost and BPNN models have shown comparatively better performances. Therefore, these three models are implemented in this study. The performance of these models is enhanced using GridSearchCV with 10-folds. The results are explained in the Section 6

5.2 Machine Learning Models

5.2.1 Random Forest(RF)

Random Forest(RF) is a machine learning algorithm which works on the concept of bagging. It is an ensemble tree based model that combines performance of numerous weak tree based models to achieve an accurate performance(Ahmad et al.; 2018). Here, Sklearn's "RandomForestRegressor"[6] library is used for the same. In the case of RF there are some important parameters which need to be configured correctly to acquire the desired results. In the presented study, 'max_depth' is kept as "None", which allows the tree to grow till its completely pure. The 'max_features' are set as "auto" to automatically set it equal to the number of variables in the input. Whereas 'n_estimators', 'min_samples_leaf', and 'min_samples_split' parameters are identified using GridSearchCV optimization technique. The default values of these parameters are 200, 1 and 2 respectively. However, higher these values better the model performs. Therefore, different combinations are considered in the parameter grid as shown below:

- n_estimators : [500,1000]
- min_samples_leaf : [1,3,5]
- min_samples_split : [2,4,6,8]

There are 24 unique combinations out of which the best performer is identified by the GridSearchCV. Rest of the features are set as default. The results are explained in a detail in the Section 6. Below is the optimal parameter combination obtained by GridSearchCV.

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=None, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=1000, n_jobs=None, oob_score=False,
                      random_state=None, verbose=0, warm_start=False)
```

Figure 8: Optimal parameters of RF by GridSearchCV

5.2.2 Extreme Gradient Boosting(XGBoost)

Similar to the RF discussed in Section 5.2.1, the XGBoost model is an ensemble tree based technique. However, it works based on the boosting technique in contrast to the RF. Unlike RF, it is an iterative process, where the errors observed in the former learner are minimized in the next tree. This helps the model to improve its performance and avoid overfitting. It is implemented using XGBoost library[7]. In the case of XGBoost,

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

[7] https://xgboost.readthedocs.io/en/latest/parameter.html

there are a set of parameters which need to be effectively selected to obtain the best performance. Here, 'objective' is set as 'reg:linear' as it is a regression study. Whereas, the most important parameters such as 'max_depth' , 'n_estimators', 'colsample_bytree', and 'learning_rate' are optimized using GridSearchCV.

- n_estimators : [500,1000]
- max_depth : [8,16,2]
- colsample_bytree : [0.3,0.4,0.5,0.6]
- learning_rate : [0.01,0.02,0.03,0.04,0.05]

There are 120 unique combinations out of which the best performer is identified by the GridSearchCV. Rest of the features are set as default. The results are explained in a detail in the Section 6. Below is the optimal parameter combination obtained by GridSearchCV

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=0.6, gamma=0, gpu_id=-1,
             importance_type='gain', interaction_constraints='',
             learning_rate=0.300000012, learning_rate_init=0.01,
             max_delta_step=0, max_depth=10, min_child_weight=1, missing=nan,
             monotone_constraints='()', n_estimators=500, n_jobs=0,
             num_parallel_tree=1, objective='reg:squarederror', random_state=0,
             reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
             tree_method='exact', validate_parameters=1, verbosity=None)
```

Figure 9: Optimal parameters of XGBoost by GridSearchCV

5.2.3 Back Propagation Neural Network(BPNN)

The Back-Propagation Neural Network(BPNN) is a deep learning model which consists of three interconnected layers (Zhao and Zhang; 2020). The first layer is called as the input layer. Here, the number of neurons are identified by the number of variables passed as input to predict the parking availability(%).The last layer in the neural network is called as the output layer. The presented study includes a regression problem and therefore contains a single output neuron which defines the availability(%). Both these layers are interconnected with each other via one or more layers called as hidden layers. Here, the process flows from input to the output layer, where at each layer weights are assigned which contribute in the prediction of availability(%). In the case of BPNN this process is repeated several times, where these weights are altered to get the best performance (Zhao and Zhang; 2020). For this, the Sklearn's MLPRegressor library[8] is used. The BPNN model consists of several critical hyperparameters which need to configured precisely. In the presented study, "Relu" is used as an activation function as availability target in this study is in the range of 0 to 1 and relu gives output in the range of 0 to $f(x)$. The 'learning_rate' is kept as "adaptive" which keeps learning at a proportion provided during the configuration and if the score is not improving it stops. The "Adam" solver is used as it works well with data above 10,000 rows. Also, the epoch is set as 1000 as higher this value better the model is in learning the complex relations. On the other hand, parameters such as 'learning_rate_init', 'random_state' and 'hidden_layer' configurations are optimized using GridSearchCV. It is observed that a single hidden layer does not

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

provide good result hence two hidden layers are used. The parameter grid of these parameters used for the optimization is as below:

- learning_rate_init : [0.01,0.02,0.03,0.04,0.05]
- random_state : [0,1,2,3,4,5,6,7,8,9,10]
- hidden_layer : [(100,100),(100,110),(100,120),(100,130)]

There are 220 unique parameter combinations out of which the best parameter is selected. The results are explained in the detail in Section 6. Below is the optimal parameter combination obtained by GridSearchCV.

```
MLPRegressor(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
             beta_2=0.999, early_stopping=False, epsilon=1e-08,
             hidden_layer_sizes=(100, 110), learning_rate='adaptive',
             learning_rate_init=0.01, max_fun=15000, max_iter=1000,
             momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
             power_t=0.5, random_state=4, shuffle=True, solver='adam',
             tol=0.0001, validation_fraction=0.1, verbose=False,
             warm_start=False)
```

Figure 10: Optimal parameters of BPNN by GridSearchCV

5.3 Hyperparameter Tuning

Grid Search Cross Validation(GridSearchCV) is an API[9] or library provided by Sklearn that allows us to identify the optimal parameters for any type of machine learning model. Provided a set of parameters it can identify the ones that perform the best depending on the evaluation measure set during the computation. For each parameter combination from the multidimensional parameter array, it calculates the performance scores depending on the measure specified (Yanxu et al.; 2015). In addition to that, while doing so it uses k-fold cross validation technique which helps to identify best performer with different test and train combinations. In the presented 10-fold cross validation enabled GridSearchCV is applied on RF, XGBoost, and BPNN to improve their performances. This identifies the best parameter for the availability(%) prediction.

6 Evaluation

The performances of the machine learning models implemented as per the Section 5 in the prediction of the parking availability(%) are critically evaluated in this section. The Evaluation measures used in the study are mentioned below:

- Coefficient of Determination(R^2)
  It depicts the reliability or correctness of the model (Yu et al.; 2015).
- Residual Mean Square Error(RMSE)
  It depicts the error associated with the predicted values (Yu et al.; 2015).
- Mean Absolute Error(MAE)
  Specifies acuteness or error of the model (Yu et al.; 2015).

[9]https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html?highlight=gridsearch#sklearn.model_selection.GridSearchCV

6.1 Experiment with Random Forest

- **Hyperparameter Optimization**
  As specified in the Section 5.2.1, the parking availability(%) prediction using RF is performed using two different settings. At first, the RF is applied with its default settings. Whereas in the second experiment it is optimized with the GridSearchCV. The result of the study are presented in the Table 11. Here, it can be perceived that there is a minor improvement in the performance of RF using GridSearchCV with 10-fold. It acquired a higher R^2 of 97.30% with minimal error and overfitting.

Experiment	R Square		RMSE		MAE		Overfit
	Train	Test	Train	Test	Train	Test	
Default Prameters	0.9963	0.9728	0.0149	0.0415	0.0102	0.0288	2.35%
GridSearchCV(10-fold)	**0.9965**	**0.973**	**0.0145**	**0.0413**	**0.01**	**0.0287**	**2.35%**

Figure 11: Parameter tuning of Random Forest

- **Feature Importance**
  The Figure 12 shows the contribution of each feature in the prediction of availability(%) in RF. It depicts that the distance from rail station shows the highest variability in the parking availability followed by hour, distance from water taxi stations, and airport in case of RF. The rest of the features depict comparatively lower contributions.

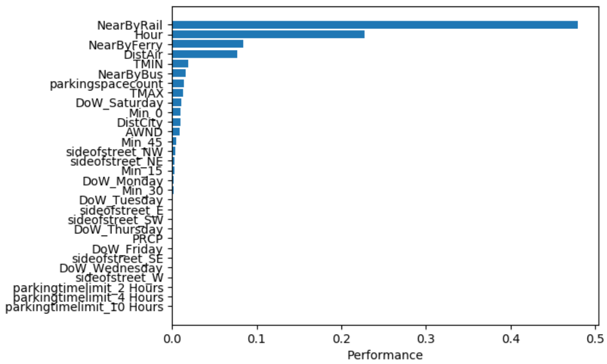

Figure 12: Feature Importance of Random Forest

6.2 Experiment with XGBoost

- **Hyperparameter Optimization**
  Similar to the RF, two experiments are carried out with XGBoost. It can be observed that the GridSearchCV optimization slightly improved the accuracy of the presented model. The optimized XGBoost performed better than RF with an R^2 of 97.65% and overfitting of 2.28%. It also acquired low RMSE and MAE as 0.03 and 0.02 respectively.

Experiment	R Square		RMSE		MAE		Overfit
	Train	Test	Train	Test	Train	Test	
Default Prameters	0.9857	0.9704	0.0297	0.0433	0.022	0.031	1.53%
GridSearchCV(10-fold)	**0.9993**	**0.9765**	**0.0064**	**0.0385**	**0.0046**	**0.0272**	**2.28%**

Figure 13: Parameter tuning of Random Forest

- **Feature Importance**
  From the Figure 14 it can be observed that unlike RF, in XGBoost the distance from airport shows higher variability in the parking availability followed by distance from water taxi station and rail stations. On the other hand, the minute categories explain the lowest variability in the availability count as compared to the hour.

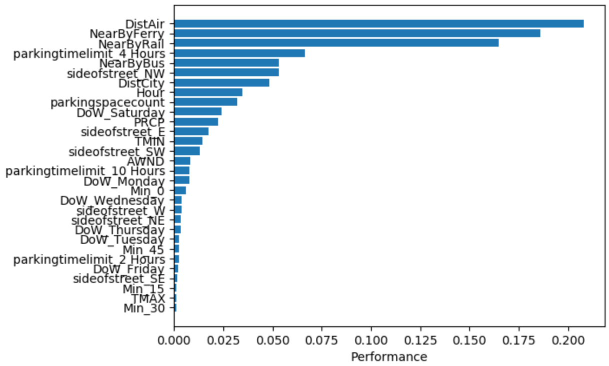

Figure 14: Feature Importance of XGBoost

6.3 Experiment with BPNN

- **Hyperparameter Optimization**
  Similar to RF and XGBoost the BPNN model is experimented with default and optimized hyperparameter combinations. It is observed that the GridSearchCV with 10-fold substantially improved the performance of the BPNN model by 8.1%. It also helped to reduce the overfitting of the model from 1.28% to 1.06% with minimal RMSE(0.0591) and MAE(0.0433). As compared to the RF and XGBoost, the BPNN model acquired the lowest overifitting.

Experiment	R Square		RMSE		MAE		Overfit
	Train	Test	Train	Test	Train	Test	
Default Prameters	0.8768	0.864	0.0871	0.0929	0.0648	0.0707	1.28%
GridSearchCV(10-fold)	**0.9556**	**0.945**	**0.0522**	**0.0591**	**0.0381**	**0.0433**	**1.06%**

Figure 15: Parameter tuning of BPNN

- **Feature Importance**
  The Figure 16 depicts the contribution of each feature in predicting the pakring availability in terms of the weights assigned. It can be observed that distance from the ferry is the most important features in BPNN model followed by Hour and the distance from city centre. However, unlike RF and XGBoost the weather features have lowest contribution in BPNN.

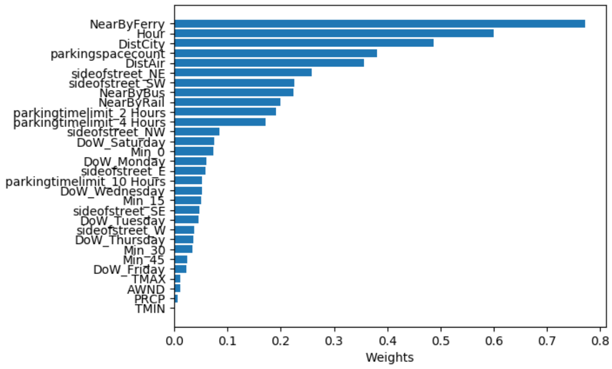

Figure 16: Feature Importance of BPNN

6.4 Discussion

This study presents prediction of parking availability in the Seattle city using spatio-temporal feature combination which is not considered in the literature. The machine learning models used in this study such as RF, XGBoost, and BPNN showed better results as compared to the previous studies. The GridSearchCV optimization used in this study helped in boosting the performance of the models significantly. It is observed that all the models achieved an accuracy above 94% and error(both RMSE and MAE) close to 0. Here, RF and XGBoost performed better than BPNN with an R^2 of 97.30% and 97.65% respectively. However, with respect to the R^2, RMSE, and MAE measures the XGBoost is identified to be the best performer. On the other hand, BPNN showed the lowest overfitting of 1.06%. However, it acquired around 2% lower R^2(94.50%) as compared to the XGBoost and RF. Overall, all of these models outperformed the previous researches in the Seattle city conducted by (Saharan et al.; 2020; Kim and Koshizuka; 2019), which considered limited factors such as the side of the street, type of the area/subarea, parking limit , block face, day, and hour. This could be due to the weak correlation between these factors and the availability. Both the XGBoost and RF opted in this study acquired a very low MAE around 0.02 as opposed to the same implemented by (Kim and Koshizuka; 2019) which resulted in MAE of 0.0592 and 0.039 respectively. On the other hand, RF and BPNN opted in this study acquired R^2 over 94% which is substantially greater than the RF and BPNN implementations by (Saharan et al.; 2020) which resulted in R^2 of 0.42 and 0.14 respectively. In addition to this, to identify the contribution of the features in the model performance, feature importance of all the models is analysed. It is observed that the presented novel features such as distance from public transport stations such as(Bus, Ferry, Rail and Airport) and city financial center, which are not considered in the literature have major contribution in improving the model performances.

Overall, this study has presented valuable contributions in the field of parking availability prediction. The accuracy of the models can help the consumers in effectively predicting the parking availability beforehand. Also, the novel feature set used in this study are found influential in the parking availability. Therefore the insights obtained in this study can help the SDOT in management of the parking systems.

7 Conclusion and Future Work

This study aims at investigating the extent to which the optimized machine learning techniques can assist in gauging the Seattle city parking availability using spatial features such as the distances from the public transport stations(Bus, Rail, Water Taxi, and Airport) and financial centre along with the temporal features such as day of the week, weather, hour,and minute. The Seattle city parking location specific factors such as side of the street and parking limit are considered as well. RF, XGBoost and BPNN models are implemented in this study with GridSearchCV optimization which managed to acquire better performance than the previous researches in the parking availability prediction. The XGBoost is identified to be the best performer in this study with an R^2 97.65%

From the exploratory analysis of the data it is observed that hour has a higher and negative impact on the availability(%). However, it is important to note that the variability of car occupancy in the intervals of 15 minutes is trivial. It further depicts that the distance from the water taxi(ferry) stations has comparatively highest positive correlation with the availability followed by the distance from the rail stations. This depicts that the commuters opt to park in a proximity to the rail and ferry stations causing high occupancy, whereas the distance from the airport has no correlation with the availability. On the other hand, parking locations closer to the bus stations and city financial center have a higher vacancy. Additionally, higher precipitation in Seattle is also identified to cause higher parking availability. This shows that on rainy days people would be avoiding private vehicles causing high availability. Considering the feature importance of the models all of these novel features followed by hour have shown major contribution in improving the model performances. On the other hand, features such as side of the street and parking limit have shown variable importance, whereas day of the week, weather, and minute show minor contributions. These valuable insights obtained from the study can help the Seattle Department of Transport in effectually planning the parking systems. The presented optimized and accurate models can help the commuters in gauging the parking availability in advance, which can assist them in planning their travel accordingly. Additionally, the novel insights of this study will also assist the fellow researchers in future explorations in the field of parking systems.

In the future work it will be intriguing to extend the presented study for parking recommendation system based on cloud, which can recommend available parking spaces based on the commuters choice of destination and time. Also, due to the big data size and absence of time only 49 parking locations are used in this study. Therefore in future it would be interesting to extend this study on all the parking segments across the Seattle city.

8 Acknowledgement

The author would like to appreciate Mr. Hicham Rifai for the valuable assistance towards the completion of this work. His expertise helped in honing both the technical and report writing fronts.

References

Ahmad, M. W., Reynolds, J. and Rezgui, Y. (2018). Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees., *Journal of Cleaner Production* **203**: 810 – 821.

Clayton, W., Ben-Elia, E., Parkhurst, G. and Ricci, M. (2014). Where to park? a behavioural comparison of bus park and ride and city centre car park usage in bath, uk., *Journal of Transport Geography* **36**: 124 – 133.

Errousso, H., Malhene, N., Benhadou, S. and Medromi, H. (2020). Predicting car park availability for a better delivery bay management., *Procedia Computer Science* **170**: 203 – 210.

Ghosal, S. S., Bani, A., Amrouss, A. and El Hallaoui, I. (2019). A deep learning approach to predict parking occupancy using cluster augmented learning method., *2019 International Conference on Data Mining Workshops (ICDMW), Data Mining Workshops (ICDMW), 2019 International Conference on* pp. 581 – 586.

Junkai, F., Qian, H. and Zhenzhou, T. (2018). Predicting vacant parking space availability: an svr method with fruit fly optimisation., *IET Intelligent Transport System* **12**(10): 1414.

Kim, K. and Koshizuka, N. (2019). Data-driven parking decisions: Proposal of parking availability prediction model., *2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT IoT and AI (HONET-ICT), Smart Cities: Improving Quality of Life Using ICT IoT and AI (HONET-ICT), 2019 IEEE 16th International Conference on* pp. 161 – 165.

Koumetio Tekouabou, S. C., Abdellaoui Alaoui, E. A., Cherif, W. and Silkan, H. (2020). Improving parking availability prediction in smart cities with iot and ensemble-based model., *Journal of King Saud University - Computer and Information Sciences* .

Li, J., Li, J. and Zhang, H. (2018). Deep learning based parking prediction on cloud platform., *2018 4th International Conference on Big Data Computing and Communications (BIGCOM), Big Data Computing and Communications (BIGCOM), 2018 4th International Conference on, BIGCOM* pp. 132 – 137.

Monteiro, F. V. and Ioannou, P. (2018). On-street parking prediction using real-time data., *2018 21st International Conference on Intelligent Transportation Systems (ITSC), Intelligent Transportation Systems (ITSC), 2018 21st International Conference on* pp. 2478 – 2483.

Origlia, A., Di Martino, S. and Attanasio, Y. (2019). On-line filtering of on-street parking data to improve availability predictions., *2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2019 6th International Conference on* pp. 1 – 7.

Rajabioun, T. and Ioannou, P. (2015). On-street and off-street parking availability prediction using multivariate spatiotemporal models., *IEEE Transactions on Intelligent Transportation Systems* **16**(5): 2913.

Richter, F., Di Martino, S. and Mattfeld, D. C. (2014). Temporal and spatial clustering for a parking prediction service., *2014 IEEE 26th International Conference on Tools with Artificial Intelligence, Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on* pp. 278 – 282.

Saharan, S., Kumar, N. and Bawa, S. (2020). An efficient smart parking pricing system for smart city environment: A machine-learning based approach., *Future Generation Computer Systems* **106**: 622 – 640.

Walaa, A., Sheng, W. and Wanlei, Z. (2017). On-street car parking prediction in smart city: A multi-source data analysis in sensor-cloud environment., *Security, Privacy, and Anonymity in Computation, Communication, and Storage : SpaCCS 2017 International Workshops, Guangzhou, China, December 12-15, 2017, Proceedings* p. 641.

Winarno, E., Hadikurniawati, W. and Rosso, R. N. (2017). Location based service for presence system using haversine method, pp. 1–4.

Yamin Siddiqui, S., Adnan Khan, M., Abbas, S. and Khan, F. (2020). Smart occupancy detection for road traffic parking using deep extreme learning machine., *Journal of King Saud University - Computer and Information Sciences* .

Yang, S., Ma, W., Pi, X. and Qian, S. (2019). A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources., *Transportation Research Part C* **107**: 248 – 265.

Yanxu, Z., Rajasegarar, S. and Leckie, C. (2015). Parking availability prediction for sensor-enabled car parks in smart cities., *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks Information Processing (ISSNIP)* p. 1.

Yu, F., Guo, J., Zhu, X. and Shi, G. (2015). Real time prediction of unoccupied parking space using time series model., *2015 International Conference on Transportation Information Safety (ICTIS)* p. 370.

Zhao, X., Chen, P., Jiao, J., Chen, X. and Bischak, C. (2019). How does 'park and ride' perform? an evaluation using longitudinal data., *Transport Policy* **74**: 15 – 23.

Zhao, Z. and Zhang, Y. (2020). A comparative study of parking occupancy prediction methods considering parking type and parking scale., *Journal of Advanced Transportation* pp. 1 – 12.