PARKING AVAILABILITY FORECASTING MODEL

A THESIS IN

Computer Science

Presented to the Faculty of the University
Of Missouri – Kansas City in partial fulfillment
Of the requirements for the degree

MASTER OF SCIENCE

By

MANOHAR BOORLU

BS, University of Missouri-Kansas City – Kansas City, Missouri, USA 2019

Kansas City, Missouri

2019

PARKING AVAILABILITY FORECASTING MODEL

Manohar Boorlu, Candidate for the Master of Science Degree

University of Missouri-Kansas City, 2019

ABSTRACT

Parking is increasingly an issue in the world today especially in large and growing cities with contemporary urban mobility. The effort spent in searching for available parking spots results in significant loss of resources such as time, and fuel, as well as environmental pollution. Parking Availability can be influenced by many factors such as time of day, day of week, location, nearby events, weather and traffic conditions. Driven by the idea of predicting parking availability to help drivers plan ahead of time, we contribute a *Parking Availability Forecasting Model*, which uses a time series analysis and machine-learning algorithms to predict the number of available parking spots at a certain location on a desired date and time. The forecasting model is trained on historical parking data from the cities of Kansas City, US and Melbourne, Australia. This paper also compares the accuracy of different time-series forecasting models, and how each of them fits our use-case scenario. Multivariate data analysis together with temperature and weather summary are used to cross-validate our forecasting model.

Keywords: Time Series Analysis, Forecasting model, Predictive modeling, Parking prediction.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Computing and Engineering, have examined a thesis titled "Parking Availability Forecasting Model" presented by Manohar Boorlu, candidate for the Master of Science degree, and hereby certify that in their opinion, it is worthy of acceptance.

Supervisory Committee

Mohammad Amin Kuhail, Ph. D., Committee Chair

Department of Computer Science Electrical Engineering

Vijay Kumar, Ph. D.

Department of Computer Science Electrical Engineering

Sejun Song, Ph. D.

Department of Computer Science Electrical Engineering

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF ILLUSTRATIONS

ACKNOWLEDGEMENTS

I am especially indebted to my advisor Dr. Mohammad Amin Kuhail, Professor in the Department of Computer Science and Engineering at University of Missouri-Kansas City, who have been supportive of my career goals and who worked actively to provide me with the protected academic time and support to pursue those goals. I am grateful to all of those with whom I have had the pleasure to work during this and other related projects. Each of the members of my Thesis Committee has provided me extensive personal and professional guidance and taught me a great deal about both scientific research and life in general. The University of Missouri-Kansas City provided me with a great platform to showcase my skills and interest in the form of research and thesis work.

Nobody has been more important to me in the pursuit of this project than the members of my family and to make them proud. I want to thank my parents; whose love and guidance are with me in whatever I pursue. They are my ultimate role models.

CHAPTER 1

INTRODUCTION

1.1 Motivation

Parking is one of the most frequent activities that motorists do on a daily basis. Parking can be an issue when there is high demand for parking, and very low supply (e.g. a few parking spots) to fulfil the demand. This problem is particularly pressing in cities with high population density. Since 2009, the United States has become home to the second largest passenger vehicle market in the world after China. Every year on average, around 7.5 million vehicles were sold to customers in the last 20 years in United States [17]. Around 70% of Americans prefer to drive to work [18].

Due to the vast usage of vehicles in the Unites States, drivers experience parking spots shortage mainly in populated areas such as city centers, shopping districts, and touristic areas. A significant portion of traffic is generated by drivers looking for parking spots in busy areas. According to Le Fauconnier and Gantlet [20] it takes around 12 minutes on average to find a free parking space. This also negatively impacts the traffic flow of other drivers since seekers of parking spots often slow down the traffic for other drivers. To fulfil the high parking demand, there are private parking garages in cities where drivers do not have easy access to free or affordable parking. Private garage parking is based on the time duration that the car is being. New York City has the highest short-term parking rate in the world with an average of $32.97 for 2 hours [19].

1.2 Proposed Solution

There have been various attempts for parking availability forecasting. Some models rely on data provided by users (crowd sourcing). The data can be explicitly reported by users or implicitly by smartphone usage. Such models can be beneficial. However, a substantial number of user base is needed to generate meaningful predictions. Other parking availability models rely heavily on sensors to report real-time parking. Such models are deployed in some cities around the world such as San Francisco by companies like SFPark. While promising, the infrastructure cost to establish these models is substantially high.

To address the issue of forecasting parking availability, we propose a Parking Availability Forecasting Model which aims at predicting the availability of number of parking spots at a given location, date and time. The model analyzes historical parking data of the city centers of Kansas City, US and Melbourne, Australia. The historical parking raw data lists the number of cars parked in five-minute intervals for about a year. The processed data is trained with various machine learning-models such as ARIMA, LSTM (RNN), and Prophet to predict the number of available parking spots through time series forecasting methods. The algorithms used variables such as weather, and day of week as our variables that could potentially have an effect on the forecasting. We have evaluated the accuracy of our model, and the results seem promising. We were able to predict the parking availability for 1 week with a mean absolute error of 1.6 vehicles for Melbourne Dataset and 0.78 vehicles for Kansas City Dataset using Recurrent Neural Networks Long Short-Term Memory architecture.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Overview

Massive volume of data is being created everyday throughout the world. We are generating around 2.5 million Terabytes of data each day and it is increasing exponentially. It is noted that 90% of the worlds data was created in the last 2 years. Such a huge volume of data is called as Big Data and it is becoming increasingly hard to store and process that data. This is why the importance of Big Data is rapidly increasing considering the four V's of Big Data i.e. Volume, Velocity, Variety and Veracity. The demand for Big Data Analytics and the need for storing and processing that information through new ways has increased.

The combination of Big Data and Internet of Things (IoT) is becoming popular and creating smarter applications for consumers. One of the main sources of data for Big Data and IoT are sensors. Sensors are of various types and of various sizes that variety collect data. The data from sensors are being used in many industries worldwide. One type is the sensors which monitor a specific assigned task over the time and collect the data. Such data is called as time-series data.

Time-series data obtained from sensors consists of data related to specific thing collected over the course of a time period indexed in time order. Generally, time-series data is successive and has equally spaced points in time. Time-series data is often used to analyze, and gain insights related to where the sensor is taking information from. This is called as time-series analysis. Many industries worldwide use time-series analysis on their data to improve their businesses.

Time series is a series of data points mostly in sequence taken at successive points in time. Time series prediction is an important component of machine learning because prediction problems usually contain a time component. Forecasting a time series data is very useful in many use cases such as business, economics and decision making. There have been many studies to find the best time series model for prediction, but the performance of a model depends deeply on the nature, type, seasonality, and complexity of the time series data. Predicting parking availability for longer times with accuracy has been the problem with parking availability prediction models.

The related work can be divided into parking availability models and time-series forecasting machine learning algorithms.

2.2 Parking-Availability Systems

Several parking-availability models rely on data reported by users. Users may explicitly report free parking spaces, or the system infers that by a phone-based application. Users can view the available parking spaces and reserve them by the application. Examples of such applications are ParkoPedia, ParkMünchen, Parkonaut, and ParkTAG. These systems have low operation and construction cost. However, they need a large user base to be successful.

Other systems such as SFPark and Park Assist generate real-time parking availability data by means of sensors. Such sensors detect whether a parking space is occupied by a vehicle, and then reports it to a database. While such systems are promising, it is highly costly to establish the infrastructure needed for reliable data.

2.3 Time-Series Forecasting Machine Learning Algorithms

In order to overcome the above issues and predict parking beforehand than showing real-time values, machine learning and statistical models can be used. In 2015, Adriana Horelu et al. [1] used the Hidden Markov Model with discrete emission probabilities, as opposed to continuous emission probabilities by Yingjian Zhang [2]. They later compared the HMM to Recurrent Neural Networks with Long Short-Term Memory in which RNNs with LSTM provided excellent accuracy. In [3], Zheng et al. used Regression tree, Support vector regression, and Neural Network on a different Melbourne dataset and concluded that Regression Tree performed the best. In [4], Yu et al. compared ARIMA model and back propagation neural network to predict unoccupied parking space with a dataset containing parking event in 15-minute bins and concluded that ARIMA model has better prediction results. Chen in [5] aggregated the clustered parking spots and performed ARIMA, Linear Regression, Support Vector Regression and Feed Forward Neural Network and concluded that Neural Networks performs best with longest training time.

CHAPTER 3

DATASET

3.1 Data Description

We worked with two parking datasets taken from two sources. The first dataset is from open data portal of the City of Melbourne in Australia [8] in csv format. This data was captured by in-ground parking sensors for every parking spot. The dataset has a total of around 36 million parking events throughout the year of 2017. To simplify our analysis and to match out processing hardware, we have narrowed the dataset down to the area of banks in Queen Street, Melbourne. This area contains around 780,000 parking events in 2017.

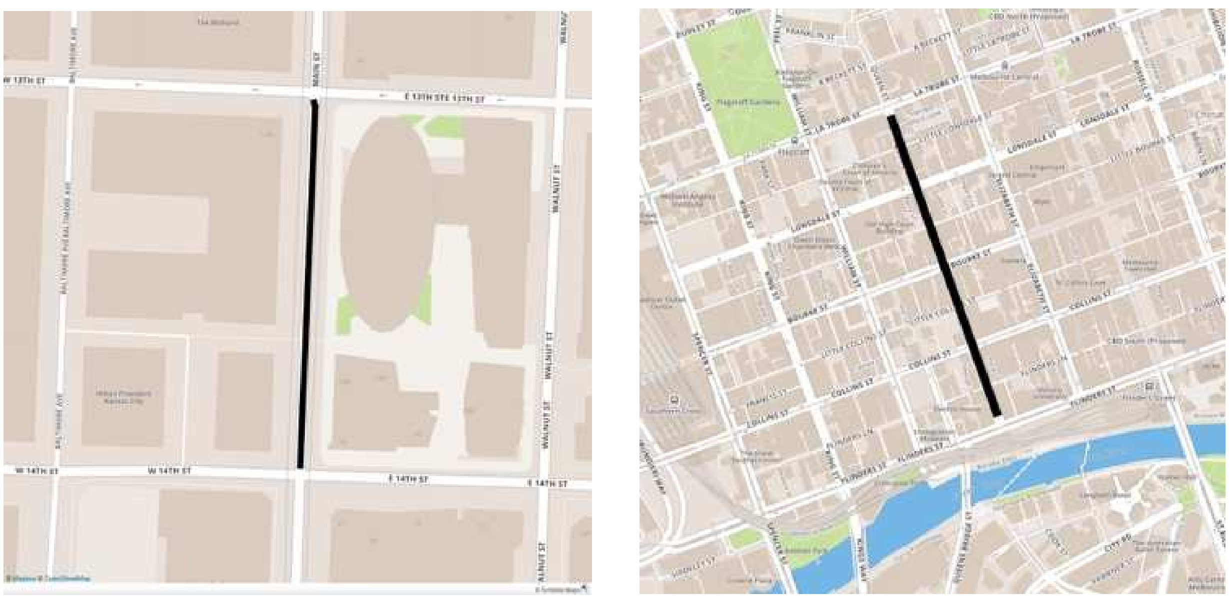


Figure 1: Street Map Showing Parking Sensor Locations

The data attributes include arrival time and departure time of a parking event of the vehicle, sensor id, area name, street name, duration of event length in seconds, street id and some other additional columns which are not necessary for our requirement. The second dataset is from Main Street, Kansas City, United States. The data contains 275,000 parking events for 8 months in 2018. The data is captured by traffic camera sensors by Xaqt Inc. [10] in csv format. This dataset attributes include arrival time and departure time of a parking event of the vehicle, sensor id, latitude and longitude coordinates of the vehicle, geobounds of the vehicle with polygon coordinates.

We have traced out a few of the issues from each of the two datasets using exploratory data analysis and found out that each of them has one major issue with their data collection from respective sensors:

- For the Melbourne data (MELB), there are a few sudden increments in parking events exactly at a few time points such as 12:00 AM, 7:30 AM, 6:30 PM etc. as illustrated in Fig. 2. This might be due to the activation of some in-ground parking sensors as each parking space has an individual sensor. This could be an issue when the forecasting model takes on the abnormality of the data into consideration.

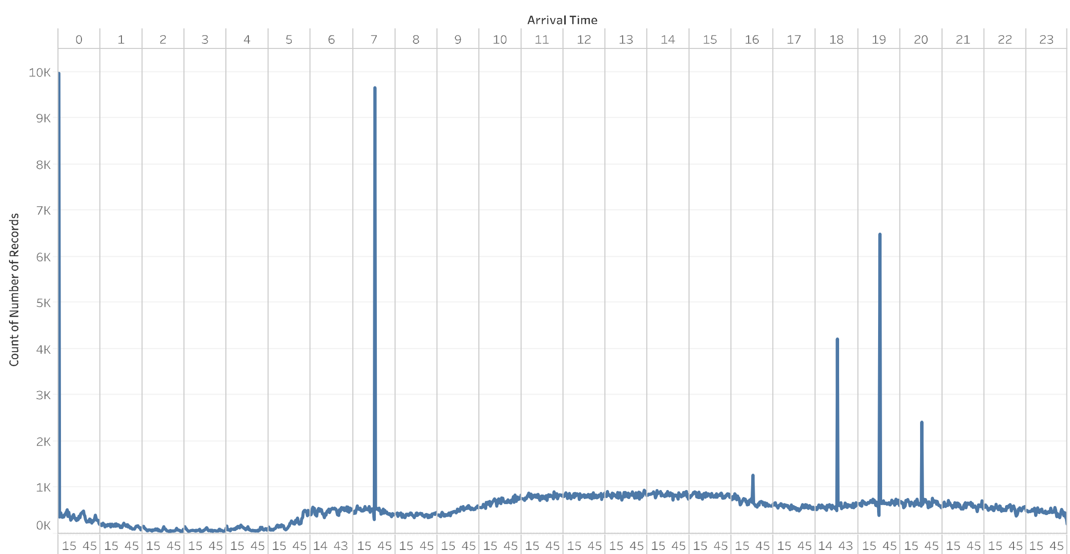
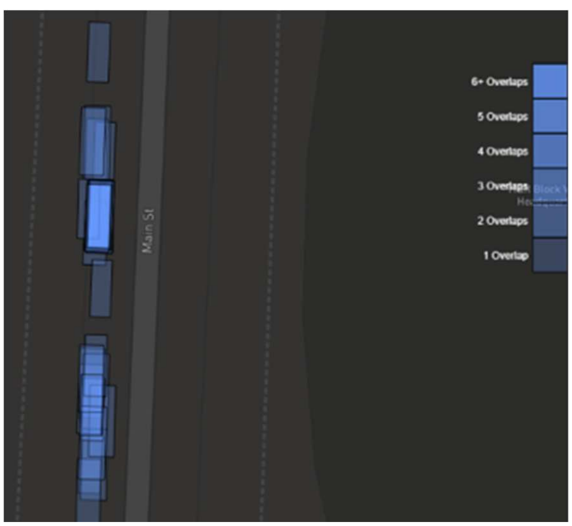


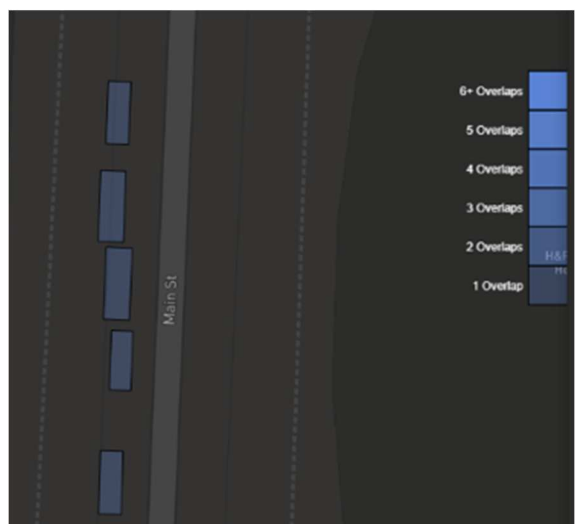
Figure 2: Dirty data of MELB dataset at few time points in a day

To address the above issue for the Melbourne dataset, we smoothened the data by truncating the extra number of parking events at those particular time which exceed the mean amount of parking events per 5 minutes i.e. around 800 parking events.

- For the Kansas City data (KC), as the data was captured from a camera-based sensor in an aerial view. There were various overlapping parking events at the same time and place. For example, several parking events had multiple cars parked at the same place in same time period. To address this issue, we plotted the vehicles using geo bound coordinates given with each parking event and truncated every event which is spatially and temporally overlapped within 1-minute intervals. This removed all the overlapped data i.e. around 30% of data and resulted in a clean dataset as illustrated in Fig. 3.



(a) Before cleaning



(b) After cleaning

Figure 3: Cleaning the parking events with spaciotemporal data

To transform the raw data with parking events to a time-series data, we created 5-minute intervals to calculate the total number of cars that were parked within that time range to get a univariate time-series data with 5-minute intervals. Time-series transformation has been coded with Pandas and NumPy [22] in Python extensively for this transformation.

Weather Summary and Temperature data have been retrieved using Dark Sky API [16] using location coordinates and added to the time-series data. The API retrieved weather data is one-hour interval data, so we added each hour data to 12 bin divisions for five-minute interval time-series data as seen in Table III.

TABLE I: Resulted Time-Series Dataset of KC

Time	No. of Vehicles	Temperature	Weather Summary
8/10/18 11:50	14	60	Clear
8/10/18 11:55	15	60	Clear
8/10/18 12:00	17	62	Mostly Cloudy

CHAPTER 4

PARKING AVAILABILITY FORECASTING MODEL

4.1 Methodology

Our system feeds all the data from different sources into a Data Pre-processor. The pre-processor performs the data transformation from raw data from sensors to time-series data and adds the weather and temperature elements to the dataset. The obtained multivariate dataset is then processed by different models we choose to use and outputs the predicted parking availability data to the user.

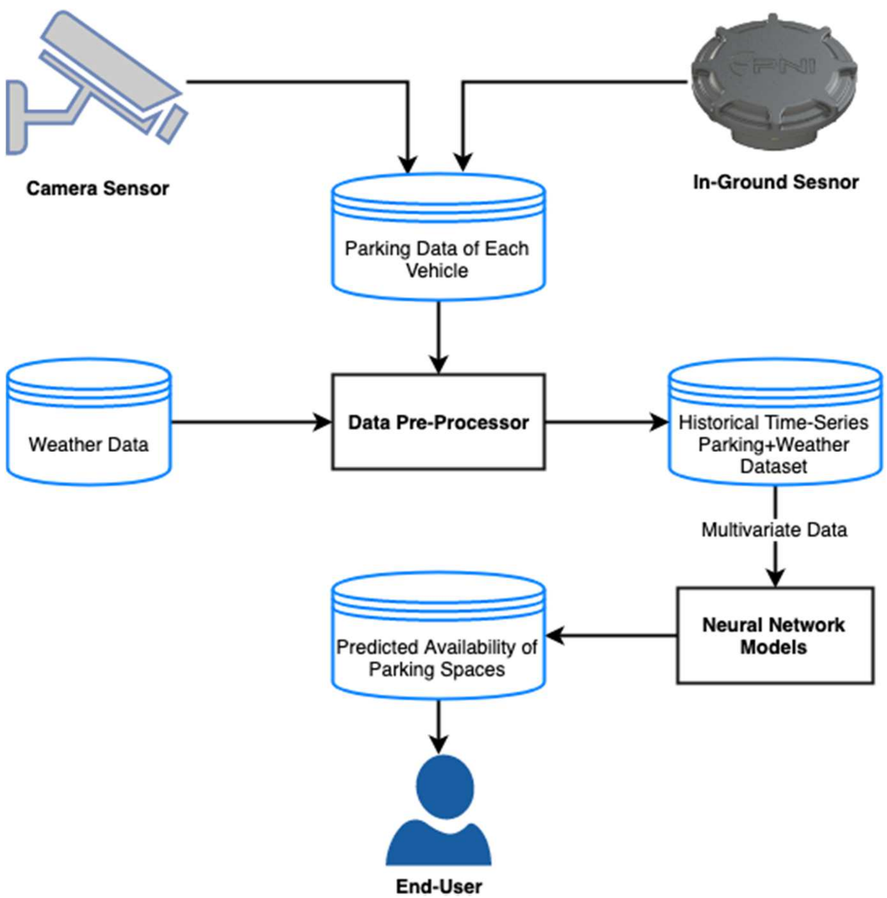


Figure 4: Architecture of Parking Availability Forecasting Model

Our aim is to predict the number of available parking spots at a required location, date and time by finding the model with highest performance and prediction accuracy. We have predicted the parking availability for 7 days i.e. 2016 bins of 5-minute interval time-series data. There are many time-series models used to forecast and predict the future of time series data. Since the time-series data is not in an increasing trend or a linear, we were careful to choose our models which can catch the non-linearity and patterns in the historical data and forecast accordingly. We have considered these 3 models because SARIMA gave out the best values when using the traditional regression models in the papers mentioned in related work and also because it considers the seasonality and patterns in the time-series data which is a rare quality in regression models. LSTM architecture outperforms other related models in neural networks as it has the ability to recognize the patterns and sequences within a temporal data and can learn long sequences of observations. Prophet The seasonality of the data is important to be smoothened and captured by all the models to get the best prediction accuracy. After carefully considering and exploratory data analysis, we have considered the following models.

4.2 The Seasonal ARIMA Model

The Seasonal Auto-Regressive Integrated Moving Average (ARIMA) method models the next step in the sequence as a linear function of the differenced observations, errors, differenced seasonal observations, and seasonal errors at prior time steps. It combines the ARIMA (also known as Box-Jenkins) model with the ability to perform the same autoregression, differencing, and moving average modeling at the seasonal level. The main parameters of the model to tweak are the maximum orders of the differencing, auto-regressive components, and the moving average components [14].

The notation for the model involves specifying the order for the AR(p), I(d), and MA(q) models as parameters to an ARIMA function and AR(P), I(D), MA(Q) and m parameters at the seasonal level, e.g. SARIMA(p, d, q)(P, D, Q)m where “m” is the number of time steps in each season (the seasonal period). A SARIMA model can be used to develop AR, MA, ARMA and ARIMA models. The method is suitable for univariate time series with trend and/or seasonal components [6].

$$\text{ARIMA} \quad \underbrace{(p,d,q)}_{\substack{\uparrow \\ \left(\begin{array}{l}\text{Non-seasonal part}\\ \text{of the model}\end{array}\right)}} \quad \underbrace{(P,D,Q)_m}_{\substack{\uparrow \\ \left(\begin{array}{l}\text{Seasonal part}\\ \text{of the model}\end{array}\right)}}$$

Figure 5: Explanation of Seasonal ARIMA Parameters

The main issue with using regression models to forecast the time-series data is the stationarity of the data. More often than not, if the data observed is not stationary, it will result in an inaccurate prediction. Hence, in order to make the best out of the regression models like ARIMA, we have to make the data stationary using methods such as Rolling Mean and Rolling Standard Deviation. The stationary data will make sure that the statistical properties of the data such as mean, variance, autocorrelation, etc. are all constant over time [23]. We can see from the Fig. 6 that the Rolling Std is between values 0 and 1. The Rolling Mean is along with the original data.

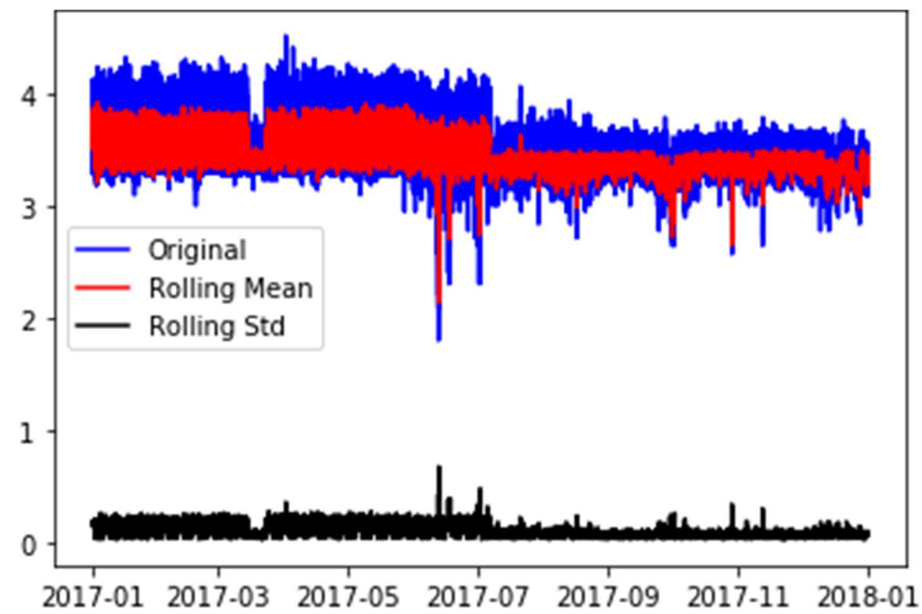


Figure 6: Rolling Mean & Standard Deviation for MELB Data

For calculating the parameters of the Seasonal ARIMA, we have used a function called auto.arima from pmdarima library [21]. It automatically tests the parameters from a pre-defined set of parameters and gives out the best performing parameters for the dataset considering AIC (Akaike Information Criterion) value.

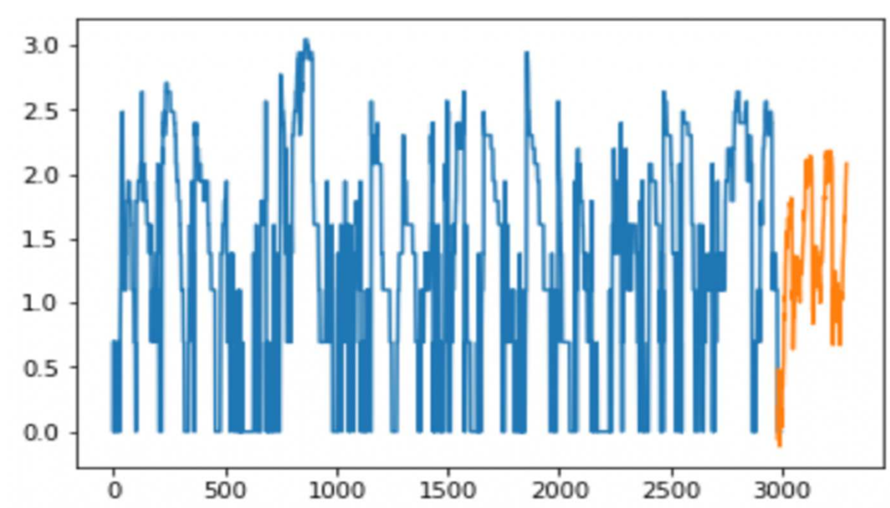


Figure 7: Forecasted values for a week KC data using SARIMA

4.3 The Long Short-Term Memory Model (RNN)

Recurrent neural networks are a type of artificial neural networks designed to recognize the patterns and sequences in data that lie within a temporal dimension [15]. Long Shot-Term Memory (LSTM) is a prominent variation of recurrent neural network has the promise of learning long sequences of observations within that linear temporal data. The purpose of the recurrent nets is to precisely classify the linear temporal input and for that it relies on the backpropagation of error and the gradient descent. But the naive recurrent neural networks suffer with the problems of vanishing gradient and exploding gradient which makes them useless for long range time series forecasting with many temporal points. So, LSTM's are designed explicitly to overcome this long-term dependency issue by introducing a memory unit which carries over the time steps from previous units.

Instead of using a grid search with pre-defined hyperparameters, we experimented trial and error with different hyperparameters like activation function, learning rate, optimizer, loss function, number of nodes, epochs etc. to learn the behavior of the model and found out the best hyperparameters that gives out best accuracy using trial and error method and later validated them using a grid search. The results show that sequential model with tanh activation function, adagrad optimizer learning rate of 0.2 with 100 nodes and 20 epochs gave good results out of the pre-defined set of hyperparameters.

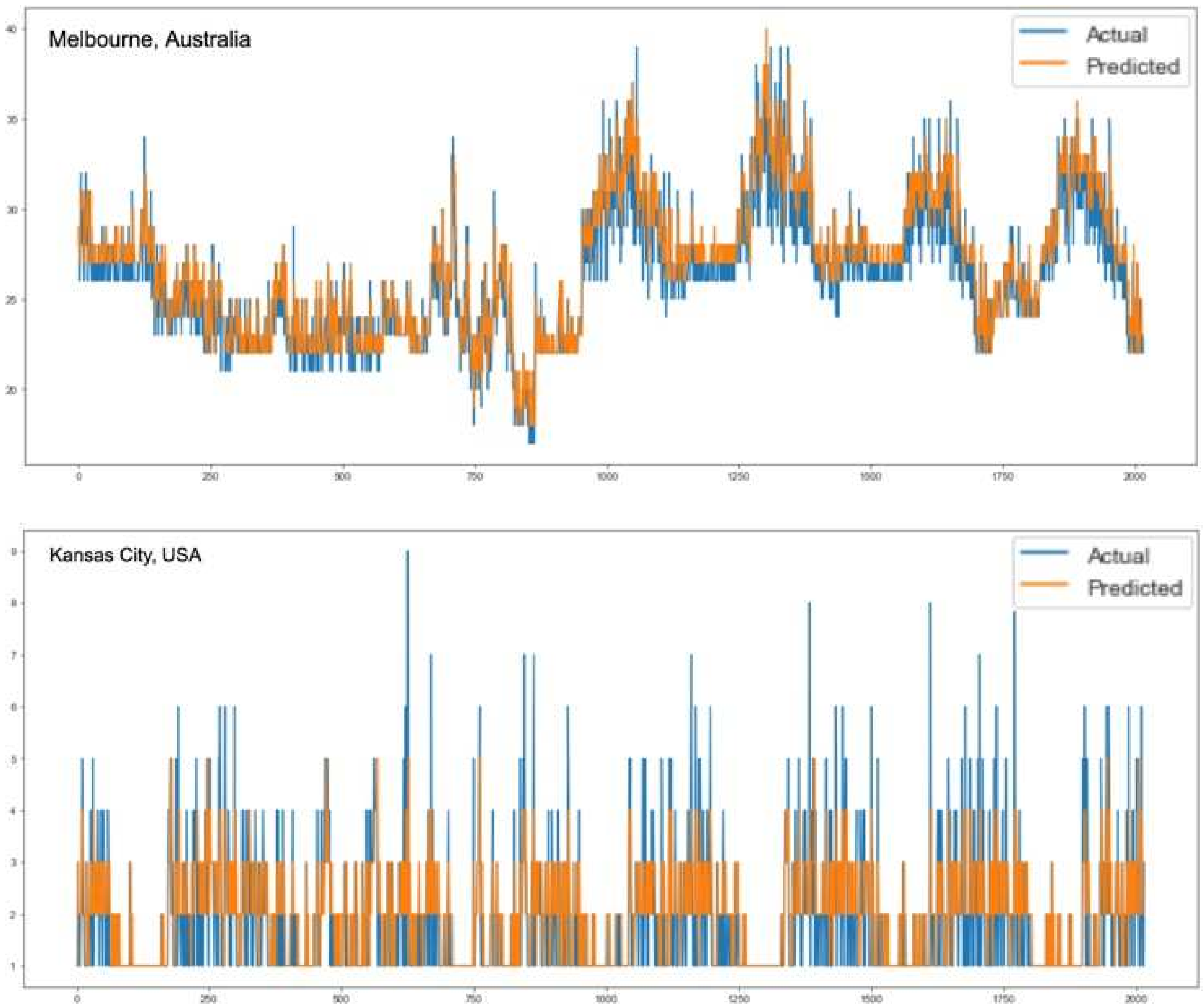


Figure 8: Prediction for MELB and KC datasets for a week using LSTM

4.4 The Prophet

Prophet [13] is an open source forecasting tool developed and open sourced by the Facebook's Core Data Science team in Facebook research. At its core, prophet is an additive regressive model built with four main components of a piecewise linear or logistic growth curve trend. Prophet automatically detects changes in trends by selecting changepoints from the data, a yearly seasonal component modeled using Fourier series, a weekly seasonal component using dummy variables, a user-provided list of important holidays. Since our data is processed in 5-

minute intervals, this model would be a good fit. To be able to predict the next value in a time series, we first train a model for that time series and then use that model to predict the next observed state. Prophet can be used as it is where it detects its own parameters and seasonality according to the data or can use human-interpretable parameters to tweak and improve the forecast accuracy accordingly if the data is known in and out [14].

Prophet is the simplest model of all we have used and also the fastest. This can be used by anyone who does not need the best accuracy without going through too much optimization of models using various hyperparameters, though it performs well comparing many of its competing models in accuracy and processing time. Fig. 9 shows the plot of actual vs predicted values and we can see that predictions are seasonal.

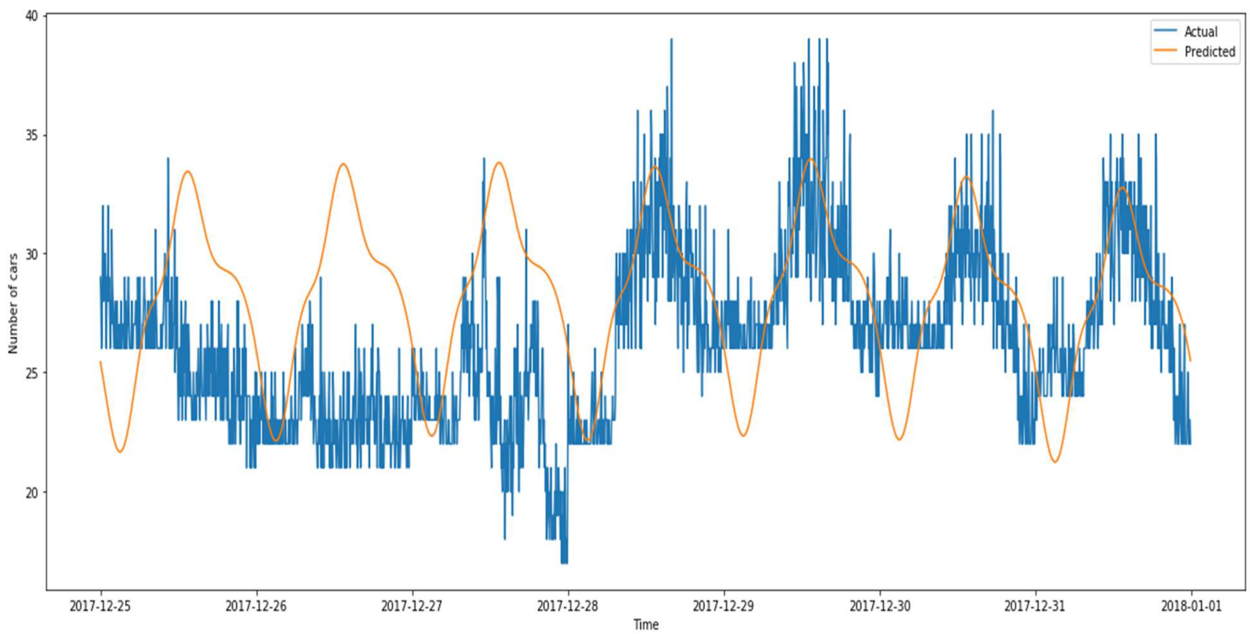


Figure 9: Prediction for MELB dataset for a week using Prophet

4.5 Correlations with Weather Summary

Weather is one of the factors that affect the driving conditions and determines whether or not we should use the vehicle for transportation as well as the ability to park in an open street. From the experimented correlations among the number of vehicles parked and the weather summary, we found some interesting correlations that makes sense when compared to real-life human thinking like many people does not want to park in an open spot and run to destination in a heavy rain.

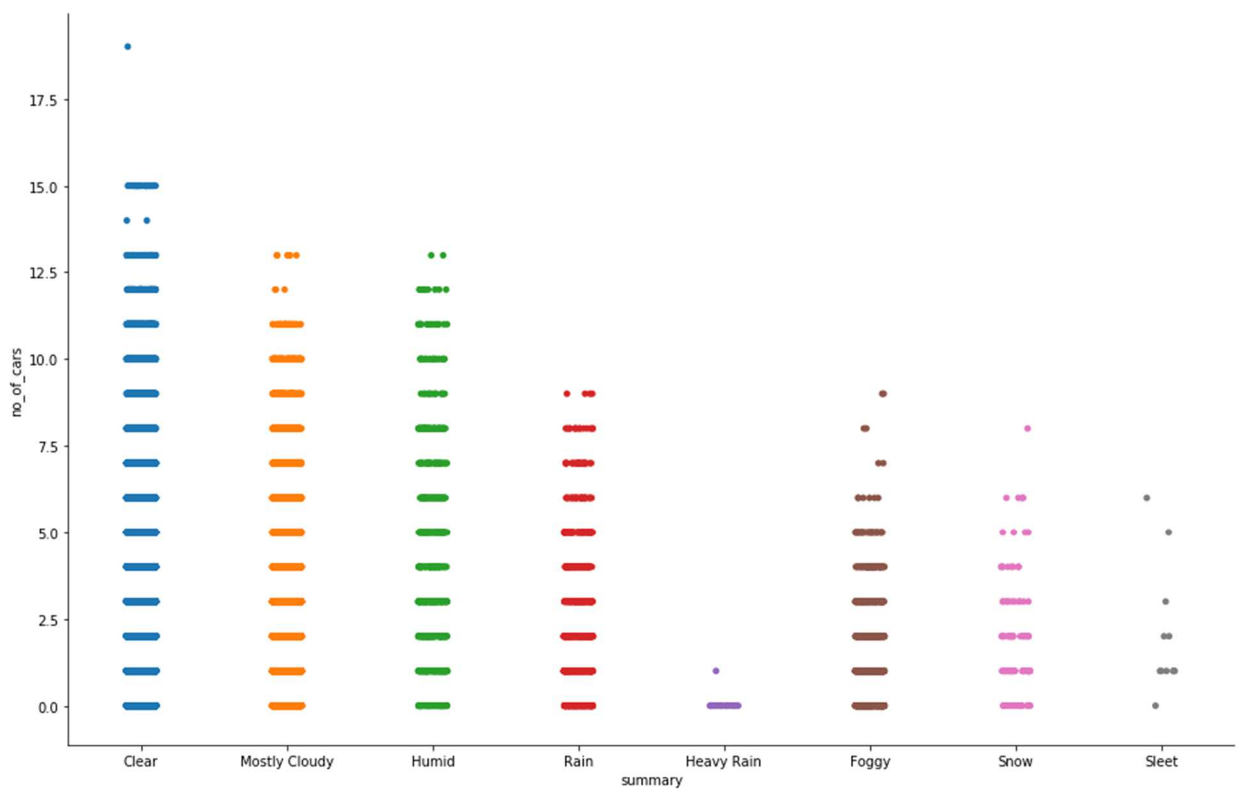


Figure 10: No. of vehicles parked on-street correlating with weather summary

From the above plot, we have noticed that there are positive correlations towards the weather conditions from normal to extreme. Normal weather conditions like Clear, Cloudy, Humid etc. has almost no effect on the parking occupancy. But we found lowered parking occupancy when there are extreme situations like Heavy Rain and Sleet which can affect the driving conditions. The moderate weather conditions like Rain, Fog etc. has moderate effect on the parking occupancy as illustrated in Fig. 10.

CHAPTER 5

IMPLEMENTATION AND RESULTS

Unlike other datasets, k-fold cross validation cannot be used for time-series data because of the temporal components inherent in the data. So, we use walk-forward validation on time series data. We have trained the above data with events in temporal order with the specified models and used the training, testing, and validation split datasets for each model. We have performed the entire calculation on a system with Anaconda environment in a jupyter notebook using python libraries like sklearn, statsmodels, Keras with TensorFlow backend etc.

To test the accuracy of the model, we have used MAE (Mean Absolute Error) which would be the closest to real-world comparison metric to see how many vehicles were off when predicting the availability of parking spots.

TABLE II. Performance of Models

	SARIMA		**Prophet**		**LSTM**	
	MELB	KC	MELB	KC	MELB	KC
MAE	1.83	0.85	3.4	0.9	1.6	0.78

Considering MAE, the predictions were off by around an average of 1.6 vehicles using LSTM for both the datasets while predicting the availability for almost 90 parking spots which is a good prediction accuracy to forecast the availability of parking spots. SARIMA took the longest to process the forecast followed by LSTM and the quickest was Prophet.

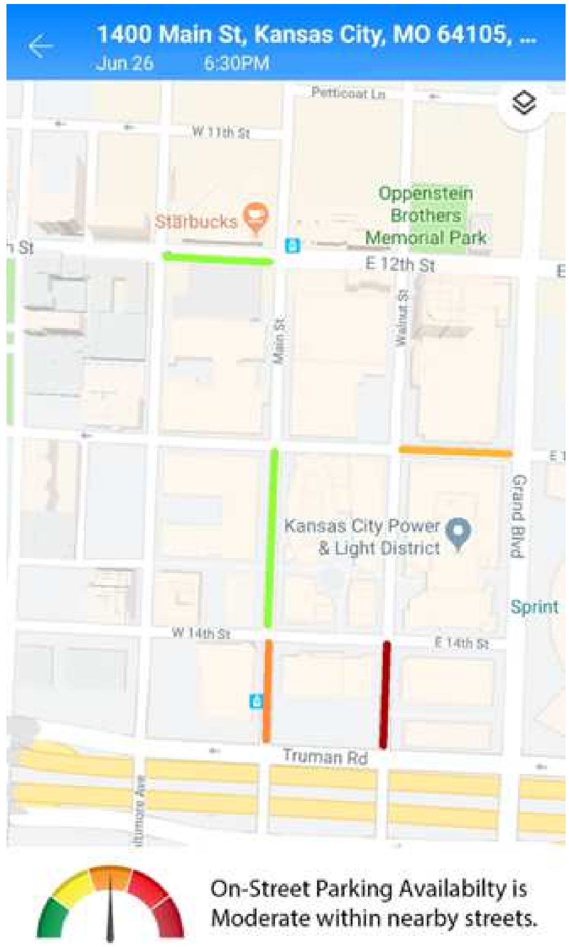


Figure 11: Prototype of Mobile Application

The end-user could potentially use this to plan his travel and parking up to a week ahead to find open parking spots especially in high population density areas where it is usually hard to find a parking spot as illustrated in Fig. 11.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, we introduced a forecasting model for predicting the availability of parking spots based on the historical data. As future work, the above best performing model can be used with real-time streaming data from the parking sensors and predicting the availability spots for the next few timestamps. We can integrate reinforcement learning to the above model and can also be implemented in this project to feed the system with correct data from sensors to the predicted values and improve the accuracy. We can also combine the weather and temperature data to make it multivariate time series data and see the effects of the weather on the parking availability.

REFERENCES

[1] A. Horelu, C. Leordeanu, E. Apostol, D. Huru, M. Mocanu, and V. Cristea, “Forecasting techniques for time series from sensor data,” *2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 2015.

[2] Y. Zhang, “Prediction of financial time series with hidden Markov Models,” Master’s thesis, Applied Sciences: School of Computing Science, Simon Fraser University, 2004.

[3] Y. Zheng, S. Rajasegarar, and C. Leckie, “Parking availability prediction for sensor-enabled car parks in smart cities,” *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pp. 1–6, 2015.

[4] F. Yu, J. Guo, X. Zhu, and G. Shi, "Real time prediction of unoccupied parking space using time series model", *Transportation Information and Safety (ICTIS) 2015 International Conference on*, pp. 370-374, 2015.

[5] X. Chen, *Parking occupancy prediction and pattern analysis*. Dept. of Computer Science, Stanford University, Semantic Scholar, 2014.

[6] J. Brownlee, "11 classical time series forecasting methods in Python (Cheat Sheet)", *Machine Learning Mastery*, 2019. [Online]. Available: https://machinelearningmastery.com/time-series-forecasting-methods-in-python-cheat-sheet [Accessed: 20- May- 2019].

[7] P. Baggenstoss, "A modified Baum-Welch algorithm for hidden Markov models with multiple observation spaces", *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 4, pp. 411-416, 2001. Available: 10.1109/89.917686 [Accessed 3 Mar. 2019].

[8] Data.melbourne.vic.gov.au. (2018). *On-Street Car Parking Sensor Data 2017*. [online] Available at: https://data.melbourne.vic.gov.au/Transport-Movement/On-street-Car-Parking-Sensor-Data-2017/u9sa-j86i [Accessed 10 Mar. 2019].

[9] G. Singh, "Methods to improve time series forecast (including ARIMA, Holt's winter)", *Analytics Vidhya*, 2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/02/time-series-forecasting-methods/. [Accessed: 08- May- 2019].

[10] Xaqt Inc. 2019, “Kansas City Main Street Car Parking Data” https://www.xaqt.com

[11] J. Brownlee, "Time series forecasting with the long short-term memory network in Python", *Machine Learning Mastery*, 2017. [Online]. Available: https://machinelearningmastery.com/time-series-forecasting-long-short-term-memory-network-python/. [Accessed: 11- Apr- 2019].

[12] S. Taylor and B. Letham, "Prophet: Forecasting at scale - Facebook Research", *Facebook Research*, 2017. [Online]. Available: https://research.fb.com/prophet-forecasting-at-scale/. [Accessed: 05- Apr- 2019].

[13] "Forecasting at Scale", *Prophet*, 2019. [Online]. Available: https://facebook.github.io/prophet/. [Accessed: 23- Apr- 2019].

[14] A. Choudhary, "Generate quick and accurate time series forecasts using Facebook’s Prophet (with Python & R codes)", *Analytics Vidya*, 2018. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/05/generate-accurate-forecasts-facebook-prophet-python-r/. [Accessed: 21- Apr- 2019].

[15] "A Beginner's Guide to LSTMs and Recurrent Neural Networks", *Skymind*. [Online]. Available: https://skymind.ai/wiki/lstm. [Accessed: 15- May- 2019].

[16] Skymind. "A Beginner's Guide to LSTMs and Recurrent Neural Networks." [Online]. Available: https://skymind.ai/wiki/lstm. [Accessed: 15- May- 2019].

[17] "Number of cars in U.S. | Statista", *Statista*, 2019. [Online]. Available: https://www.statista.com/statistics/183505/number-of-vehicles-in-the-united-states-since-1990/. [Accessed: 11- Jan- 2019].

[18] U.S. Department of Transportation. Highlights of the 2001 National Household Travel Survey, 2003. [Online]. Available: https://www.workingcarsforworkingfamilies.org/images/files/usdot_statistics_%20nhts_highlights_rpt.pdf. [Accessed: 30-July-2019].

[19] Parkopedia. Global Parking Index 2017. [Online]. Available: https://www.parkopedia.com/static/reports/global_parking_index2017-parkopedia.pdf. [Accessed; 30-July-2019].

[20] E. Gantelet and A. Lefauconnier, "The time looking for a parking space: Stragtegies, associated nuisances and stakes of parking management in France", in Proceedings of the European Transport Conference, Strasbourg, France, 2006.

[21] "pmdarima: ARIMA estimators for Python — pmdarima 1.2.1 documentation", *Alkaline-ml.com*. [Online]. Available: https://www.alkaline-ml.com/pmdarima/. [Accessed: 11- Mar-2019].

[22] T. Oliphant, *Guide to NumPy*. Austin, TX: Continuum Press, 2015.

[23] "Stationarity and differencing of time series data", *People.duke.edu*, 2019. [Online]. Available: http://people.duke.edu/~rnau/411diff.htm. [Accessed: 16- Feb- 2019].

VITA

Manohar Boorlu completed his Bachelor's degree in Computer Science from University of Missouri-Kansas City (UMKC). Manohar Boorlu started his Master's in Computer Science emphasizing in Data Science at the same University of Missouri-Kansas City (UMKC) in August 2017. While being a student at UMKC, he interned at 3 different companies as a Software Engineer, Data Analyst and Data Scientist respectively for a period of 3 months each. Upon successful completion of all his degree requirements, he plans to work as a Data Scientist to help companies take data-driven decisions for better success.