UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Marten Jaago

Predicting Demand for Smart Parking Systems

Bacherlor's Thesis (9 EAP)

Supervisor: Amnir Hadachi, PhD

Tartu 2019

Nõudluse Ennustamine Nutikatele Parkimissüsteemidele

Lühikokkuvõte:

Kiire areng infotehnoloogias on võimaldanud inimestel, kes ei ole antud ala eksperdid, luua täpseid masinõppe mudeleid, mis suudavad õppida olemasolevatest andmetest. Selliste mudelite rakendamine linnaplaneerimisel esinevatele probleemidele võib aidata luua paremaid elamistingimusi suurlinnades. Antud töö keskendub tehisnärvivõrgule, mis suudaks ennustada parkimisala hõivatust. Töö käigus välja töötatud mudeli integreerimine süsteemi, mis suunab inimesi parkimiskohtadele võiks anda väärtusliku toote, mis vähendaks liiklusummikuid ja parkimiskoha otsimiseks kuluvat aega.

Võtmesõnad:

Parkimine, masinõpe, linnaplaneerimine

CERCS: P170

Predicting Demand for Smart Parking Systems

Abstract:

The rapid development of information technologies has enabled to create accurate machine learning models that could learn from the existing data without the need of an expert in the field. Applying these models in the field of urban design problems could help to create better living conditions in large cities. This research focuses on creating a neural network that could predict parking area occupancy. Integrating the model to a parking guidance platform would produce a valuable product that could help to reduce the traffic congestions and the cruising time for a parking spot.

Keywords:

Parking, machine learning, urban design

CERCS: P170

Contents

1. Introduction

1.1 General View

With the number of vehicles on the road in Europe rising from 243 million to 257 million in the last 5 years [1] we are reaching a state where the parking systems have to be optimized for the growing amount of cars. It is estimated that 30% of vehicles on the road are spending an average of 7.8 minutes on cruising for a parking spot in the downtown areas of major cities [2] and therefore affects the urban areas with congestion and unnecessary emission of polluting gases [3] [4].

The strategies for solving this problem could mainly be divided into two categories: parking spot reservation and parking spot recommendation. Reserving parking spots often lead to a problem where the parking spots are always reserved with many of the drivers that reserved a spot being absent. Parking spot recommendation, however, applies a policy that provides users more information about the best parking lot near their designated destination. [5] This research focuses on developing a model that would help to recommend a parking spot.

Projects like LA Express Park and SFPark have tried to tackle this problem by implementing demand-based pricing models [4]. By informing the driver of a cheaper alternative nearby the destination the decision of where to park could be influenced. Other projects using parking guidance and information systems like EasyPark[1] aim to guide the user to an optimal parking spot based on the destination, estimated occupancy and other factors.

As a solution, this research uses artificial neural networks to predict parking area occupancy. When combined with a parking guidance service it could distribute the parking evenly throughout the areas in the system. This research focuses only on the on-street parking because usually gated parking or private area parking already have some kind of a management system applied which would regulate or inform users of the occupancy.

In cooperation with EasyPark and the city of Stavanger in Norway, the data was gathered about previous parking between the years of 2010 and 2019. Stavanger has taken an interest in the development of innovative parking systems to improve the urban area infrastructure.

[1] https://easypark.se/howItWorks/en

In addition, a demand-based pricing model will be developed based on the previous projects like LA Express Park and SFPark [4] which could also help to change the customer behavior so that the parking would be distributed evenly.

1.2 Objectives

The aim of this thesis is to develop two different models. One that could predict parking area occupancy on a given time and a second one for dynamical pricing.

The predictive model has to accurately predict parking area occupancy based on the data of previous parking in any given area. To develop a comprehensive model the data has to be analyzed and prepared accordingly.

For the demand-based pricing model, previous researches are to be investigated to see if the scenario fits our input data. Hypothetical prices will be generated to see if the model is competent enough to be tested in the field.

1.3 Limitation

The parking rates have not fluctuated enough in order to develop a model that could take into account the prices and predict parking area occupancy based on that as well. As the prices have been quite stable a new model has to be developed and field tested or the user behavior has to be simulated. This is beyond the purpose of this research.

Rate changes have to be discussed with the city in regards to the demand-based pricing model and limitations have to be set so that the prices would not reduce the parking revenue.

1.4 Contribution

The contribution of this research would be to reduce the economical footprint of big cities by improving the cruising time before finding a parking spot, reducing congestions and guiding users to an available parking spot.

Developing the models that could be integrated into the EasyPark's service would provide a wide platform that would make it possible to make the guidance and demand based pricing available on a large scale.

1.5 Roadmap

This thesis is divided into four chapters that are described as follows.

Chapter 2: Describes different state-of-the-art methods of machine learning and regression models.

Chapter 3: Lists the methods and steps used to achieve the goal of the thesis. It explains the neural network structure and the methods used for data preprocessing.

Chapter 4: Describes dataset and gives an overview of the analysis done during the research. It shares the results of the predictive model and describes the prices that the demand-based model could propose.

Chapter 5: Conclusion and future research perspectives are presented in this chapter.

2. State-of-the-art

2.1 Introduction

With the rapid urbanization [6] creating a demand for smart solutions to tackle problems like pollution and traffic, cities have been investing into the development of AI and machine learning in order to improve their livability. On the supply side there is the availability of devices and services producing data that could be utilized to improve the urban design of a city [7].

The application of this research is to develop models that could be used to improve the parking systems. This is done by using iterative models and machine learning methods like regression analysis and deep neural networks[2]. Deep artificial neural networks were chosen because they have proven to predict patterns in data better than simple regression models [8].

2.2 Parking Area Occupancy Prediction

Parking area occupancy prediction combined with the latest deep learning methods has proven itself more accurate than the estimating researches. The advantage that a long-term predictive model provides over a short-term estimating model is the practicality for parking guidance applications. Drivers can make better decisions when they know the probability of finding a vacant parking spot on the time of their arrival. In addition, it helps to reduce the harmful pollution gases that are produced while cruising for a parking spot. [9]

A neural network allows incorporating features besides the previous parking area occupancy like weather forecasts and traffic speed. That has significantly improved the predictive model's performance as proven by a group of researchers at Carnegie Mellon University. On a test set their model achieved high accuracy with a relatively low number of training epochs. [9]

2.3 Regression Analysis

General

As described by Geoffrey R. Urland and Kevin B. Raines the regression analysis is a form of data analysis technique that inspects associations between different input variables. With it, we can analyze the influence of several variables on a dependent variable. [10]

[2] https://www.techopedia.com/definition/32902/deep-neural-network

Linear regression

Linear regression is a type of regression analysis that can be expressed in the form of the following equation:

$$y = w_1 x_1 + w_2 x_2 + \cdots\ w_n x_n + b$$

Here y is the dependent real variable, x_1, x_2 and x_n are the input variables, w_1, w_2 and w_n are the regression coefficients (from now on referenced as weights) and b is the error (from now on referenced as bias) in the prediction. The model will estimate the weights and the bias by finding the regression lines that minimize the prediction error. Prediction error represents the difference between the dependent variable and the actual value through an error function. An example of an error function is the mean of the sums of the squared differences between the prediction and the actual value. [10]

2.4 Artificial Neural Networks

General

Artificial neural networks are parallel computational models consisting of interconnected processing units called neurons or perceptrons. An important feature of these networks is their ability to learn by examples rather than solving a problem by using programmed rules. This makes the neural networks applicable to fields where one does not have the required knowledge of the problem to be solved but has the training data available [11].

There are many different architectures of neural networks with the most popular ones being feed-forward, recurrent and long short-term memory neural networks [12]. This research will use a feed-forward neural network.

Feed-Forward Neural Network

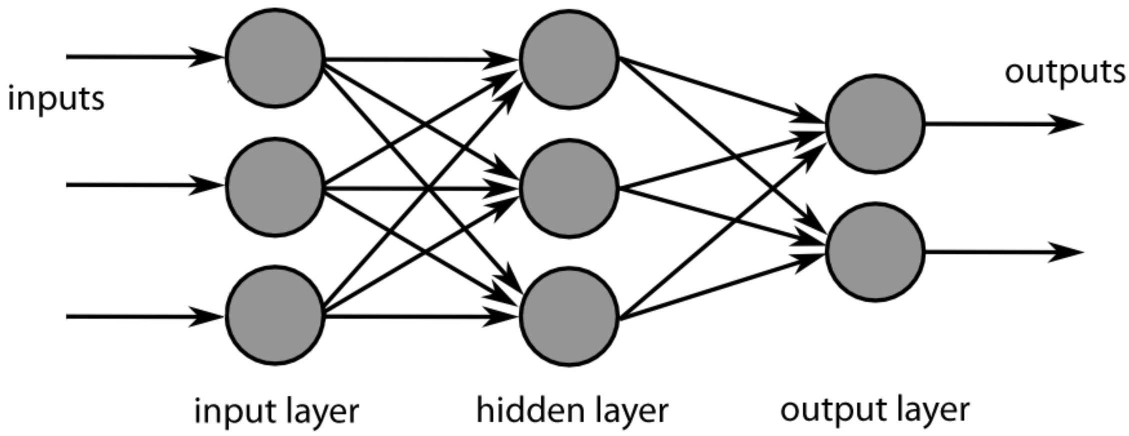

Figure 1. Example of a feed-forward neural network. [13]

Figure 1 describes an example of a feed-forward neural network. In a feed-forward neural network, neurons are arranged in layers where the first layer receives the inputs and the last one produces outputs. The hidden layers have no direct connection to the outside world [14].

Feed-forward neural networks are an extension of the previously described linear regression where the weights are represented by matrices or vectors instead of single variables. In addition, it adds non-linearity by applying non-linear activation functions to the outputs of a layer in the network. Therefore each output from a previous layer gets applied an activation function after which they are multiplied by a set of weights and summed up to *feed-forward* values to the neurons of the next layer [15].

As in the previously described linear regression, the feed-forward neural network tries to optimize the model by estimating the weights by applying a method named gradient descent iteratively for every layer. This iterative approach for estimating the weights in a neural network is called backpropagation. In each iteration, the network feeds the data forward and outputs a prediction which is then followed by backpropagation. During the backpropagation, the derivative of the error function with respect to the weights in each layer is subtracted from the respective weights. [15]

2.5 Conclusion

Combining artificial intelligence with urban design is something that every city with available resources should strive for. It could be used to improve the living quality for the residents in the city while reducing the ecological footprint.

A feed-forward neural network is just the next step from the linear regression model. As stated in the introduction part of this chapter they can produce more accurate predictions than a simple linear regression. This can be explained by the complexity of adding non-linear activation functions and multiple layers of weights to the model which makes it more applicable for different patterns in the data.

3. Methodology

3.1 Introduction

As stated in the first chapter this research aims to develop two models. One is the model that could accurately predict parking area occupancy and the other one being a demand-based pricing model.

For the predictive model, the data needs to be preprocessed in order to avoid problems like overfitting or even longer training times. Preprocessing the data adds a better ability of generalization to the model without losing necessary patterns and also optimizes the computations [16]. Neural network algorithms also involve hyperparameters which are variables set before optimizing the model's parameters [17].

The demand-based pricing model is inspired by the iterative approach of previous projects like SFPark and LA Express Park which applied a rate changing model to cities like Los Angeles and San Francisco [4].

3.2 Preprocessing

This research used two preprocessing methods to improve the quality of the model and the data. Normalizing is used to put the data in a common frame [16] which would speed up the learning process as it makes the gradient descent converge faster [18]. The second preprocessing method is Z-score which helps to get rid of outliers [19].

Z-Score

An outlier is a single sample in a set that deviates markedly from other samples in the set. Potential reasons for a sample to be an outlier could be either bad data, some random variation or it might be caused by some external event. [19] Removing the outliers will result in cleaner data. It also might cause the algorithm to work incorrectly or fail to learn correctly from the given data [16].

This research uses the Z-score method to detect and remove potential outliers. Z-score of an observation expresses in units how many standard deviations the sample is from the mean. The Z-score of observation could be expressed as follows:

$$Z_i = \frac{Y_i - \bar{Y}}{s}$$

With $\bar{Y}$ and *s* denoting the mean of the samples and the standard deviation respectively. [19] Z-score allows to set a threshold and identify the sample with values over the threshold as an outlier.

Normalization

Normalization or scaling is necessary because it helps to maintain the distribution of the input. Otherwise the layers in a neural network would have to constantly readapt to the new distribution when there is a change in the input's distribution. When the distribution of the input changes it is said to experience a covariate shift. When the input is normalized the distribution remains the same and the standard deviation decreases. Therefore the weights do not have to be readjusted when the input experiences a covariate shift. It also speeds up gradient descent by fitting each feature into the same range making them equally important. [18]

This research uses a normalization method called min-max normalization. Min-max normalization is a simple method to fit the values of the input to a predefined boundary. Let the values of the input be X. Then the min-max normalization technique could be described as following:

$$X_{normalized} = \left(\frac{X - X_{min}}{X_{max} - X_{min}}\right) * (D - C) + C$$

Where,

X_{max} is the maximum value of X

X_{min} is the minimum value of X

C and D are the lower and the upper limit of the predefined boundary respectively.

This technique would output the normalized values in the predefined boundary [20].

This research fits the data into the range [0,1] which is common. Therefore the formula could be expressed as:

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

3.3 Network Structure

Neural networks are designed by setting a number of variables called hyperparameters. Hyperparameters are features that define a neural network. There are many ways to find the most optimal parameters for your neural network. For example, they could be set by hand, selected by some algorithm or found by some hyper-learner. Hyperparameters include features like the loss function, number of epochs, number of hidden neurons, activation functions and more. [17]

The number of epochs determines how many times the feed-forward and backpropagation loop described in section 2.3 is running. One of the most common ways to determine the optimal number of epochs is the principle of early stopping which stops training once the performance of the network on a test data set stops increasing. This is done to prevent overfitting. [17] In this research the limit for the number of epochs were set manually because of the time and resource limits.

The number of hidden neurons is defined by the hidden layers and their size. Using a first hidden layer which is larger than the input tends to work better [17]. This research set the optimal number of hidden layers based on the results of manually testing multiple different numbers of layers with different sizes on a smaller test set to see which produced the best result.

Commonly used activation functions are the sigmoid, tanh, rectified linear unit and the hard tanh [17]. This research uses rectified linear units because they are proven to be more accurate than the widely used sigmoid function [21]. In addition, they showed the best results when manually testing different activation functions. The rectified linear unit's function is:

$$f(x) = \max(0, x)$$

Loss function defines the model's accuracy. A general loss function can be written as the squared average over the training set. This loss function is called the mean squared error. Mean squared error is computed as follows.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

Where n is the number of calculated predictions, y_i is the correct value of the predicted variable and $\hat{y}$ is the value predicted by the model.

3.4 Demand Based Pricing Models

Tom De Ruijter conducted research over the course of two and a half years based on the LA Express Park project for demand-based parking. The idea was to reduce the amount of cruising and improve driver satisfaction when searching for a parking spot. This research uses the data provided by EasyPark as a substitute. From the results of the demand-based pricing model, the data about the changes in the parking behavior could be obtained and used for future reference. [4]

3.5 Conclusion

This chapter introduces the methods used in this research in order to get to the objective. Normalizing the data and removing the outliers makes consuming the data and training the model quicker and more accurate. Setting several hyperparameters while testing different possibilities ensure a model that is capable to perform accurate results.

4. Results and Discussion

4.1 Introduction

The practical part of this thesis is divided into 5 categories which were done in the following order:

1) Getting the input data

2) Data evaluation and visualization

3) Preprocessing the data

4) Building and training the predictive model

5) Building the demand-based pricing model

Based on the visualization the research is trying to find areas where there are congestion problems which could be improved with the demand-based pricing model.

The parking data is divided into review periods so it could be analyzed. An example of a review period is the year 2018 divided into hourly averages. Workdays and weekends are analyzed separately as human behavior differs.

An area is considered congested when the average occupancy in a review period is over 90% of the areas maximum capacity. An area is considered underused when the occupancy in a time period is under 70%. [4] The maximum capacity for an area is the highest number of parking simultaneously active in the given time period. The number of active parking sessions in an hour is the number of parking that was active during the one hour period.

4.2 Input Data

The input data was taken from EasyPark's database. As a provider of parking services, they hold the parking data about the city of Stavanger dating back to 2011. Different types of parking in the system include gated parking, private area parking, electrical vehicle loading station parking, and on-street parking. As mentioned in the introduction this thesis focuses only on the on-street parking. In addition, they hold the geographical data for the parking areas in the city which was also used in this research.

The parking data was exported from the database. The exported parking data includes the fields beginning date, end date, the id of the area and the type of parking. The geographical data for the areas was fetched through an API provided by EasyPark. For every area, there was a set of geographical coordinates in the format of GeoJson[3].

[3] http://geojson.org/

Data was exported between the first occurrence of parking in Stavanger in the EasyPark system which was in November of 2010 until the year 2019. Parking with areas that were inactive or areas that have been deleted for now were filtered out. Between the given years there have been roughly 700 000 parking. Among these, there are 11 unique on-street parking areas.

4.3 Data Evaluation and Visualization

In order to understand the problem better the data was visualized spatially and temporally. Spatial visualization was done to have an overview of how the areas are divided and from the temporal visualization, it was possible to analyze the patterns in human parking behavior. This part of the research is trying to find out whether the problem of overuse exists in the city of Stavanger. In addition, this section tries to find if there are patterns in the data that the neural network could learn.

Spatial Analysis

Geospatial visualization was done using Python with packages like GeoPandas[4] and Folium[5]. All parking areas were projected on the city of Stavanger in the world map. A parking area is visualized on the map as a rectangle or a set of rectangles all of the same color. Different areas have different colors. Some additional information about the area is displayed on a pop-up after clicking on any area.

[4] http://geopandas.org/
[5] https://python-visualization.github.io/folium/

Figure 2. All areas displayed on the map.

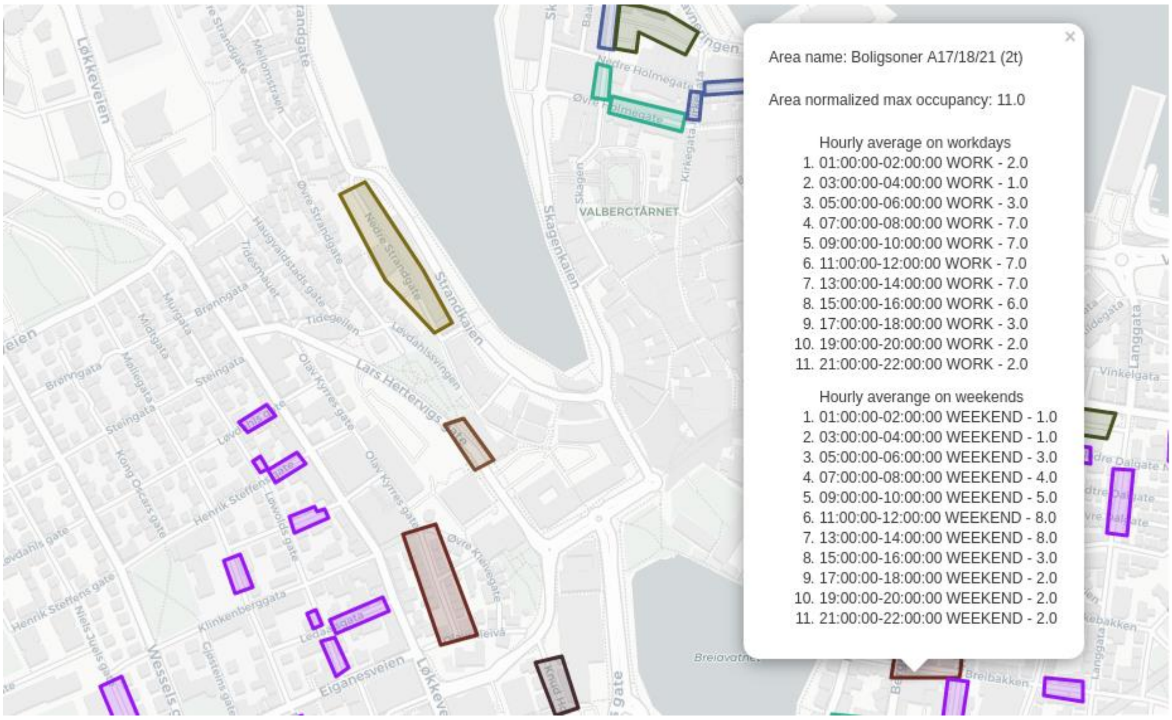

Figure 3. Pop-up with additional information about the area.

The additional information on the pop-up is based on the year 2018. Averaging it over all the data would produce misleading information as the customer’s behavior has changed from the year 2011. A year's worth of data is still informative enough to give an overview of the situation.

During the visualization, a problem was discovered. The distance between two parts of the same area could be too long for it to be beneficial to predict the area's occupancy. As visible in figure 2 some parts of the purple area are very distant so predicting the occupancy area-wise requires improvements.

Taking the year 2018 as a review period none of the areas showed signs of overuse meaning that no areas were found where the average occupancy in an hourly time period was over 90%.

When looking at the monthly review period in the year 2018 there were two problematic areas as seen on the map below.

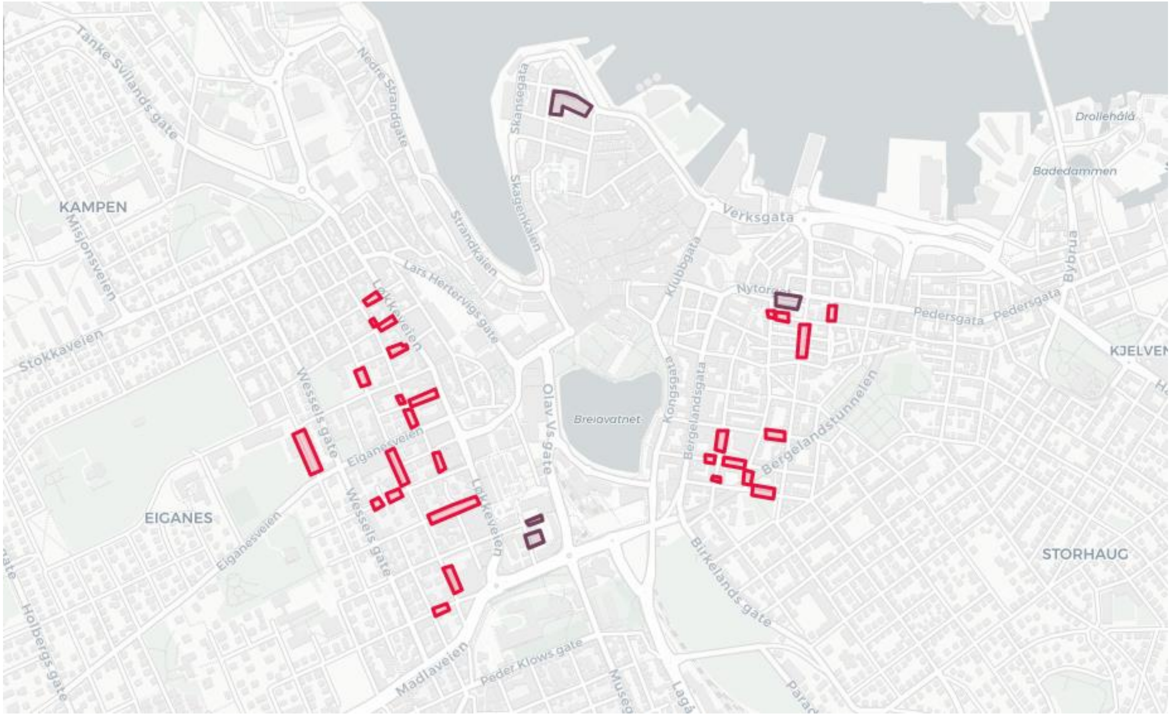

Figure 4. Areas that experience monthly overuse.

Because of the sparse distribution of the areas, it is hard to specify what geographic areas are the most problematic ones.

Weekly review period in the year 2018 revealed that 8 out of the 11 areas experience overuse on a weekly basis. The map below represents areas that experience weekly overuse.

Figure 5. Areas that experience weekly overuse.

That seems to indicate that there are problems that could be solved on either with a demand-based pricing model or with an intelligent parking guidance platform.

Temporal Analysis

The temporal analysis was done using mainly Pyplot[6] from the Matplotlib[7] library and seaborn[8]. The temporal analysis was done using the year 2018 which is the most recent one and describes customer behavior the best.

Firstly the days were divided into workdays and weekends to see if there is are some occurring patterns. If the days were to divided into weekdays the graphs for the days from Monday to Friday would look similar and graphs for Saturday and Sunday would look similar as well thus the days were divided by the type of day.

In the graphs below the x-axis represent the hour of the day and y-axis represents how much parking have there been in that type of day during a given hour.

[6] https://matplotlib.org/api/pyplot_api.html
[7] https://matplotlib.org/
[8] https://seaborn.pydata.org/

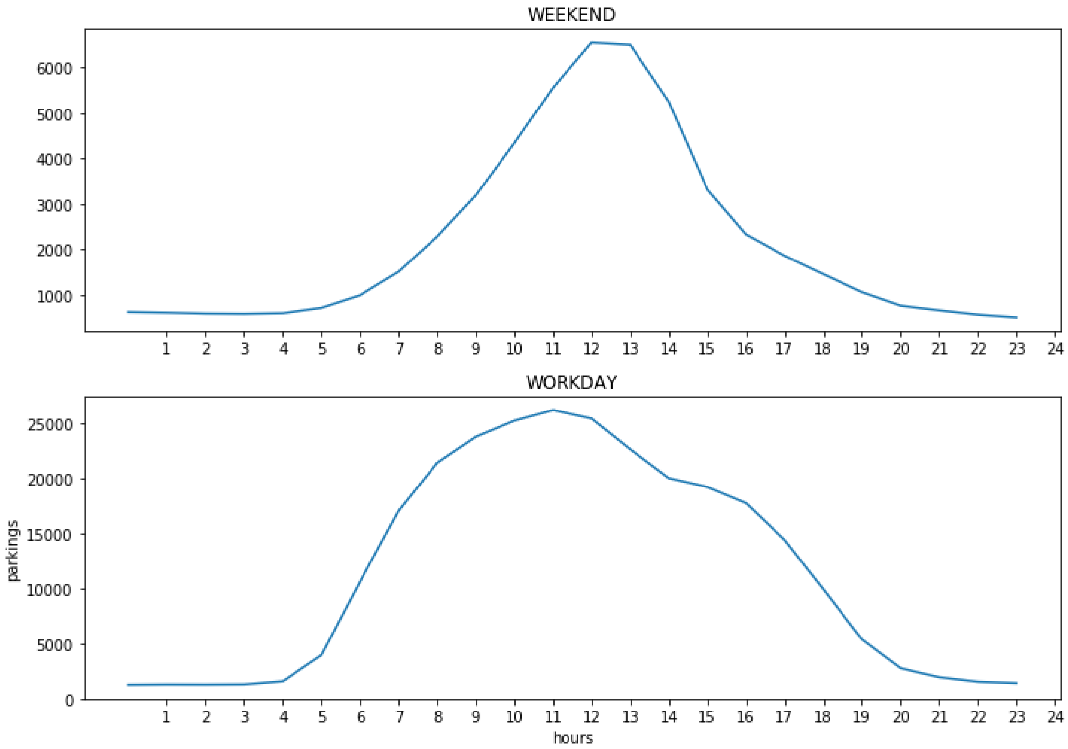

Figure 6. Parking by the type of day.

In addition, the problematic areas described in the geographical visualization section were investigated. To get a better overview of the extent of the overuse in these problematic areas monthly and weekly review periods graphs were plotted. The graphs describe how many hours have the problematic areas been overused on a timely basis. Weekly review periods graph was plotted as a line graph opposed to the bar graph because the frequency of overuse is higher than for the monthly review period.

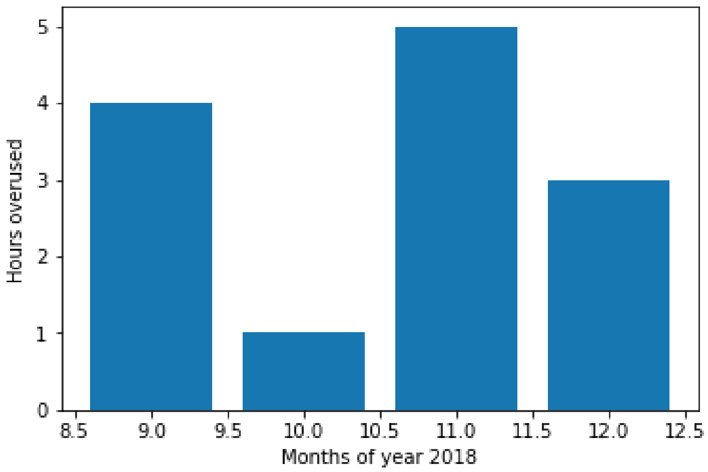

Figure 7. Hours of overuse for a monthly review period.

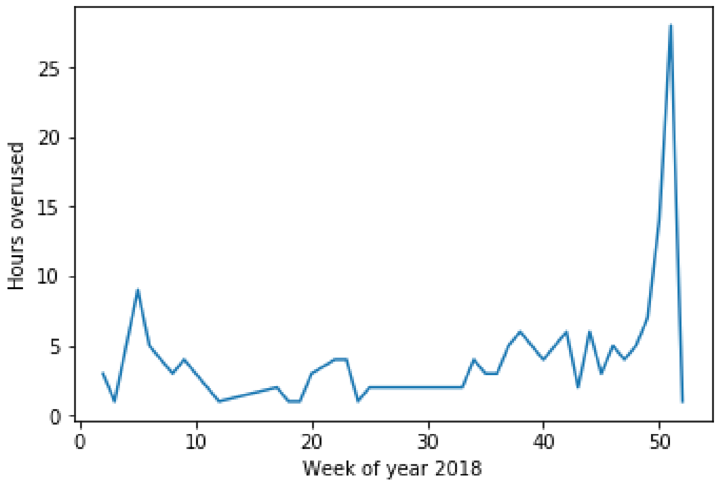

Figure 8. Hours of overuse for a weekly review period.

As visible in figure 6 there are clearly some patterns. Seems like on the workdays people start parking a lot earlier which makes sense considering the fact that people go to work which normally starts somewhere around 8 in the morning. By that time the parking has almost peaked and the number of parking does not rise a lot above that. The number of parking starts to fall somewhere around 12 which could is possibly caused by lunch breaks and leaving after that. Parking count is relatively stable around hours 14 and 17 after which it starts to fall again which is most probably caused by the end of the workday so people leave the city and the paid parking areas.

The amount of overuse on a monthly period shows that the months September, October, November and December experience overuse. Averaging it over a monthly review period leads to believe that the problem is not that serious as the number of hours of overuse never goes above 5 hours.

The number of hours of overuse on a weekly review period seems alarming towards the end of the year where there are over 25 hours of overuse per week on average. This number is bigger than on a monthly basis due to the fact that the weekly overuse was experienced by 8 areas opposed to the 2 areas, therefore, the hours of overuse distribute on more areas.

Conclusion

This confirms that there are patterns which could be predicted by the predictive model. Also as the figures 7 and 8 show, there are time periods in areas where the demand-based pricing model could have some effect on the overuse. Although because the overuse is not a big problem the demand-based pricing model would have small effects on the prices for the given areas.

4.4 Preprocessing

Preprocessing is done by first formatting the data to numerical features. This is followed by filtering out the outliers, fitting the numerical values to a bounded range and then splitting the data to training and test sets.

Formatting

As described in section 4.2 the data includes the fields start date, end date, area id and the type about a parking session. The data was grouped into hourly periods so that every parking that has an overlay with a certain hour of the day is considered active parking during that hour. It also provides a certain level of generalization that is useful for the model without losing the necessary features.

The predictive model used features extracted from the temporal data. The input features of the training samples were the following:

- Year, values from 2011 to 2019
- The month of the year, values 1 to 12
- Hour of the day, values 0 to 23
- Is workday, the binary value

The predicted variable was the number of active parking sessions in an hour.

Normalization

Normalization is performed in two phases. First, the outliers are removed by the Z-score method described in section 3.2. That is followed by a min-max normalization to fit the numerical values in the boundary [0, 1].

The Z-score was applied to every area separately because different areas have different distributions. The Z-score was applied only on the dependent variable which was the number of active parking sessions in an hour. Other values were not applied to the Z-score method because it was temporal data. Values with the Z-score over 3,5 [19] were filtered out. This filtered out less than half of a percent worth of data which also indicates to quite an even distribution of parking during the year because of the small number of outliers.

Min-max normalization was applied separately for every area as well to get a more accurate scaling. The scales were saved using a package named pickle[9] so they could be reused and new samples could be scaled the same.

Train and Test Data Splitting

Train and test data were also split separately for each area so that training and test set would contain the same amount of data relatively to the absolute amount of data per area. The training set contains 80% of the data and the test set contains 20% of the data. The data was shuffled before it was split to feed the model data about different time periods.

4.5 Predictive Model

The model that would predict parking area occupancy was made in the form of a neural network. For easier implementation, this research used TensorFlow[10] for creating the neural network and managing the weights. The neural network implemented for this thesis was a fully connected sequential feed-forward neural network with 6 layers with 4 being hidden layers and 2 layers for input and output. The best results were achieved with a learning rate of 0.0001.

Layers from 1 to 5 used the rectified linear unit activation function as their neurons output whereas the last output layer just multiplied the weights and summed the product to produce a real-valued output. The visual representation of the neural network is described in figure 9.

[9] https://docs.python.org/3/library/pickle.html
[10] https://www.tensorflow.org/

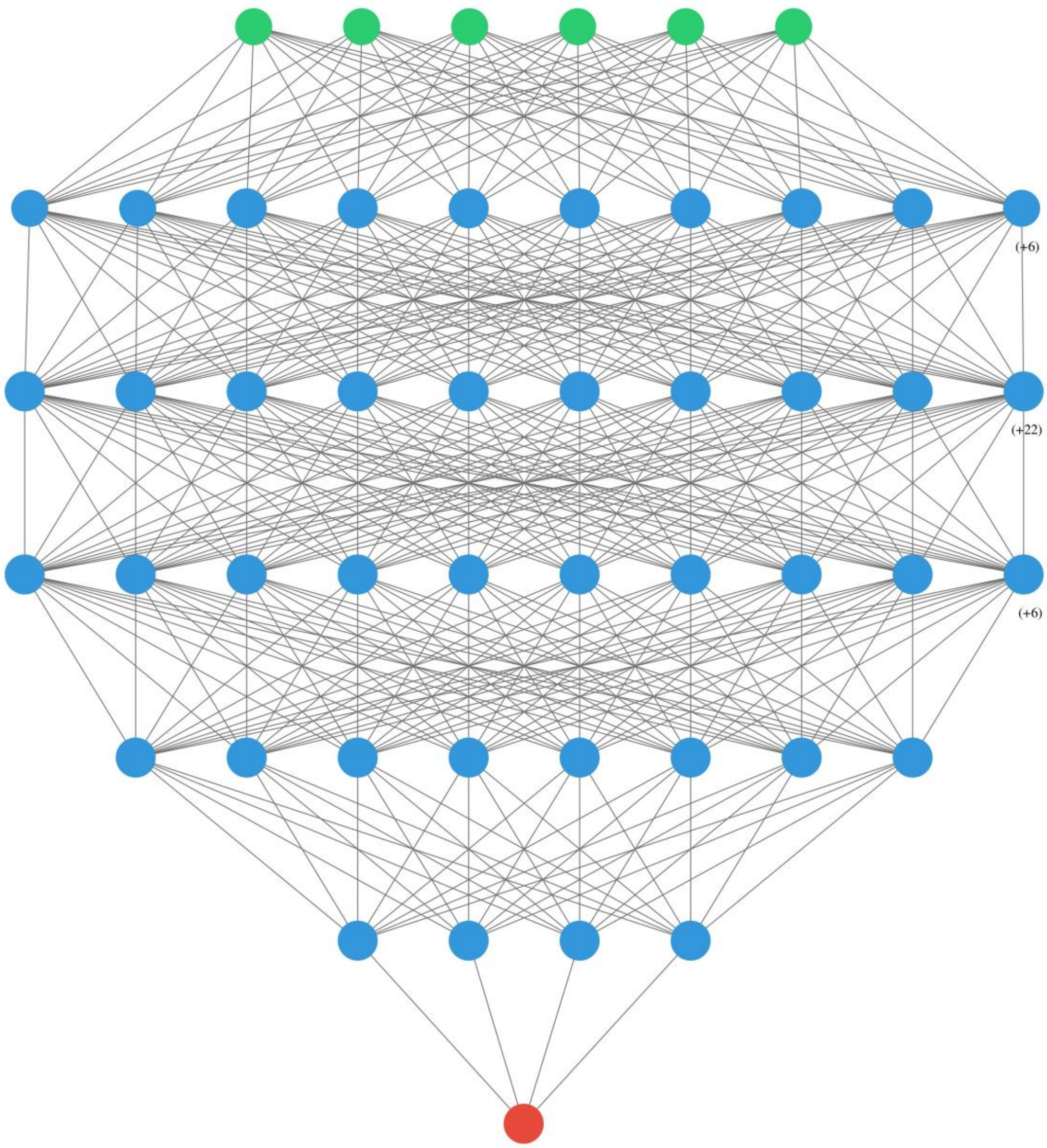

Figure 9. The visual representation of the neural network.

The model was trained for 300 epochs for each area separately. The weights were saved to separate folders for different areas after the training so weights could be reused to produce predictions on samples. After every 50 epochs, the weights and the cost function outputs for test and training set were saved to keep track of overtraining and to persist the optimal weights.

4.6 Demand Based Pricing Model

The demand-based pricing model was implemented using the following iterative model from the research conducted by Tom De Ruijter [4].

Algorithm 1 The LA Express rate change iteration

for each block b **do**

- Compute the *congestion index* $I_c^{(b)}$ as the fraction of operating hours in the review period that b is congested (occupancy $> 90\%$).
- Similarly compute the *underuse index* $I_u^{(b)}$ (occupancy $< 70\%$).
- Define the *congestion-underuse balance* as
$$B_{cu}^{(b)} = I_c^{(b)} - I_u^{(b)}.$$

for each block b with $B_{cu}^{(b)} > 1/3$ (congestion dominant problem) **do** Increase the rate by one step in the ladder.

for each block b with $B_{cu}^{(b)} < -1/3$ (congestion dominant problem) **do** Decrease the rate by one step in the ladder.

Pricing ladder is using 0.5€ steps when the congestion-underuse balance exceeds the defined thresholds. The street blocks in that algorithm are replaced with parking areas in the city of Stavanger.

The initial prices were taken from the data provided by EasyPark. The prices are only valid in the operating hours defined by the city officials. The congestion-underuse balance was calculated separately for each area. The model could only be applied for one iteration because there is no data available about the changes in human behavior after a price change.

4.7 Results

Predictive Model

The predictive model trained successfully. After 300 epochs the cost was low for all areas and it was still dropping on certain areas in the test set. The training was stopped because of time consumption. The example of a loss function is visualized in the graph below.

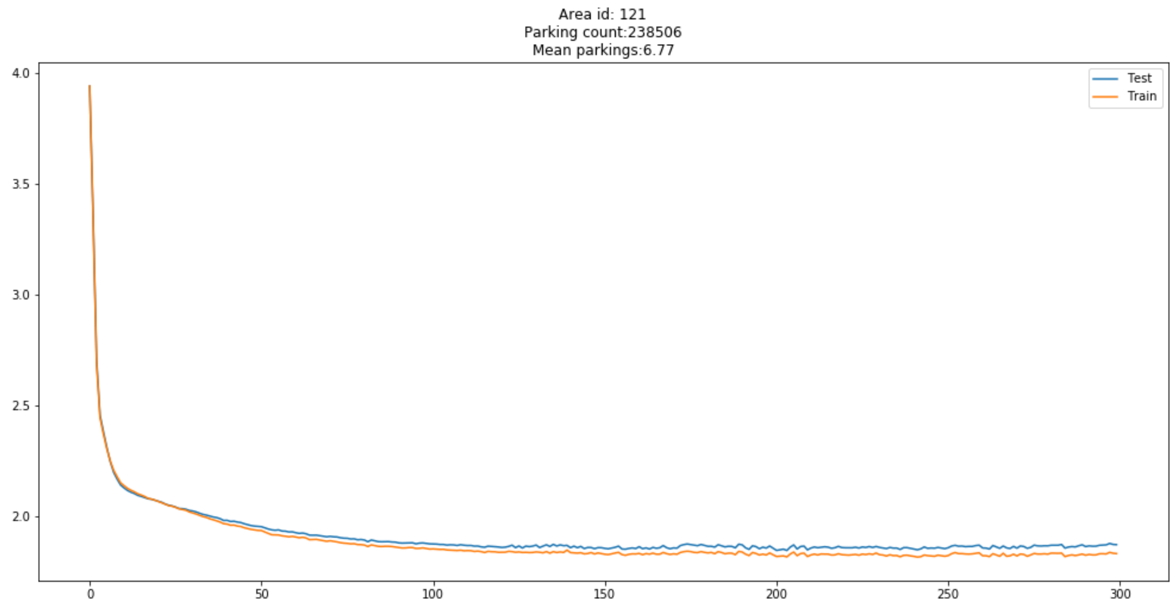

Figure 11. The loss function for the area with the most parking.

Figure 11 describes the mean of the loss function which was the mean of the difference between the predicted values and the correct values in the given epoch. As visible, the value of the function drops below 2 which is enough to predict the probability whether an area has a vacant spot or not. The loss drops until the 250th epoch after which it is rising a bit which may indicate overfitting.

The test set losses for every area after every 50th epoch are visualized in the table below.

	30556	122	115	3057	40339	40338	3058	114	119	121	19278
0	1.380844	0.238797	1.516967	0.944407	7.352557	5.291502	0.634488	4.235522	2.330485	3.938908	1.061364
50	0.904195	0.212148	1.087696	0.634947	0.450817	0.992164	0.530908	2.069912	1.327280	1.955740	0.672582
100	0.904564	0.212646	1.061800	0.631471	0.449550	0.979907	0.528086	2.025903	1.305671	1.878101	0.666755
150	0.901425	0.214165	1.056669	0.632823	0.448414	0.974312	0.529371	1.997206	1.294694	1.858095	0.665692
200	0.901225	0.213013	1.050063	0.633978	0.447843	0.973472	0.528085	1.979936	1.289289	1.848926	0.660855
250	0.898132	0.215681	1.055503	0.632096	0.448572	0.972780	0.528021	1.971557	1.287164	1.861028	0.659429
299	0.899873	0.214825	1.060276	0.631808	0.450365	0.973979	0.527758	1.953491	1.285694	1.874905	0.660527

Figure 10. Loss function table for all areas

It seems that the values have been converged and might be starting to overfit as the loss function for the test set is already rising in some areas. In areas where the value of the loss function is not rising in the last row of the table the difference between the previous row is marginal and these could be considered to be converged as well.

Demand Based Pricing Model

The congestion underuse balance in the yearly review period for 2018 was below the threshold for every area, therefore, the model had no effect on the prices.

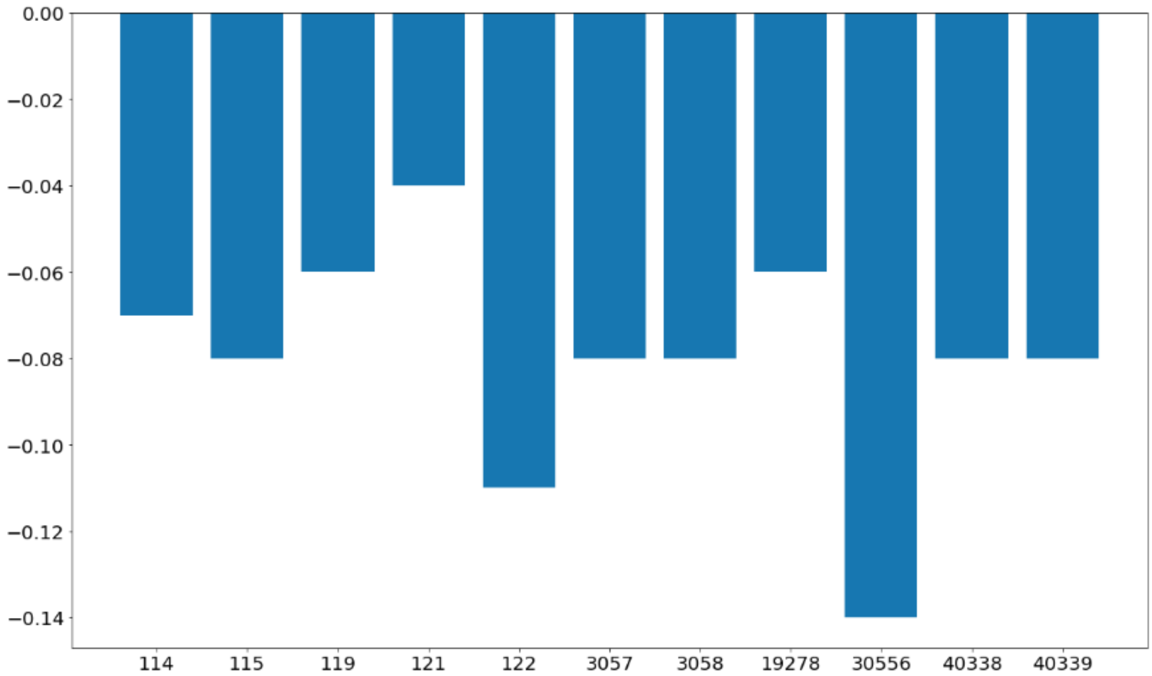

Figure 12. Congestion-underuse balance for 2018 review period.

The thresholds for the monthly periods in the year 2018 were all below average as well so no change could be applied to the prices. A weekly analysis was not conducted because changing the price weekly would be too frequent and the users would not have enough time to change their behavior.

The prices remained the same and they will not be presented in this research as they do not add any extra value and are business critical variables.

4.8 Conclusion

Although the demand-based pricing model did not have any effect on the current prices it supports the observation made in chapter 4.3 that the overuse is not that big of a problem in that region. The predictive model's accuracy proves that there are patterns in the data which could be used to predict the number of active parking in a parking area at a certain point in time.

5. Conclusion

As the rate of urbanization rises so does the problems related to the urban design. Luckily some these of these problems could be solved by integrating the latest technologies with innovative ideas. The numerous services and devices collecting data daily provide the opportunity to implement smart solutions on problems that have not been solved before with machine learning and data science.

The predictive model developed during this research could be used to shorten the time for cruising for a parking spot by integrating it with a parking guidance app or provide the data to the user in any other understandable way. This could alleviate many problems. For example, it would reduce the emission of polluting gases, reduce traffic congestions, improve the driving experience for the city residents and tourists and many more.

The demand-based pricing model should be tested by applying it to a region where there parking related problems under supervision. If it does not prove to be useful then the data could be used to analyze human behavior after a price change.

5.1 Future Perspective

The predictive model could be integrated with a parking guidance app. It provides predictions accurate enough that a user could be led to a parking area with a vacant spot. It could also be used to give an overview of the distribution of parking in the city and based on that a heatmap could be produced that would give the user a probability of where it could find the nearest parking spot at a given time in the future.

In addition, the predictive model could be improved by taking more data into account. For example, the weather data, the traffic speed [9] and the price rate could be added to the list of input variables which might make the model more accurate.

The demand-based model should be field tested in an area where there are problems with cruising for a parking spot. After the field testing the data should be analyzed and another model should be improved so that it could take into account the changes in the user behavior after a price change.

6. Bibliography

[1] European Automobile Manufacturers' Association, "www.acae.be," European Automobile Manufacturers' Association, Europe, 2018.

[2] T. R. R. S. Richard Arnott, Alleviating Urban Traffic Congestion, Massachusetts: CESifo Book Series, 2005.

[3] P. Hossam El-Din I. S. Ahmed, “Car Parking Problem in Urban Areas, Causes and Solutions,” p. 1, 2017.

[4] T. De Ruijter, A big data view of on-street parking, Nijmegen: Radboud University, 2015.

[5] S.-Y. C. C.-Y. C. G. C. Jiazao Lin, “SPA: Smart Parking Algorithm Based on Driver Behavior and Parking Traffic Predictions,” *IEEE Access,* vol. 7, pp. 34275 - 34288, 2019.

[6] J. Griffiths, “CNN,” 15 10 2018. [Online]. Available: https://edition.cnn.com/2018/05/16/world/world-population-cities-un-intl/index.html.

[7] T. V. Ark, “Forbes,” 26 6 2018. [Online]. Available: https://www.forbes.com/sites/tomvanderark/2018/06/26/how-cities-are-getting-smart-using-artificial-intelligence/#691b5f373803.

[8] M. Y. H. P. Reza Gharoie Ahangar, "The Comparison of Methods Artificial Neural Network with Linear Regression Using Specific Variables for Prediction Stock Price in Tehran Stock Exchange," *International Journal of Computer Science and Information Security,* vol. 7, no. 2, 2010.

[9] W. M. X. P. S. Q. Shuguan Yang, “A deep learning approach to real-time parking occupancy prediction in spatio-temporal networks incorporating multiple spatio-temporal data sources,” p. 21, 2019.

[10] G. R. U. &. K. B. Raines, “Regression Analysis,” in *Encyclopedia of Survey Research Methods*, Thousand Oaks, Sage Publications, Inc., 2011, pp. 711-712.

[11] F. o. A. N. Networks, Mohamad H. Hassoun, MIT Press, 1995.

[12] A. Tch, “Towards Data Science,” 4 8 2017. [Online]. Available: https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464.

[13] W. Feng, “The University of Tennessee,” [Online]. Available: http://web.utk.edu/~wfeng1/spark/_images/fnn.png.

[14] J. Pang, “Stanford University,” 2000. [Online]. Available: https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Architecture/feedforward.html.

[15] J. Rocca, “Towards Data Science,” 8 12 2018. [Online]. Available: https://towardsdatascience.com/a-gentle-journey-from-linear-regression-to-neural-networks-68881590760e.

[16] D. Chicco, “Ten quick tips for machine learning in computational biology,” *BioData Mining,* vol. 35, p. 10, 2017.

[17] C. Raffel, “Colin Raffel Wiki,” 17 12 2015. [Online]. Available: http://colinraffel.com/wiki/neural_network_hyperparameters.

[18] C. S. Sergey Ioffe, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *arXiv,* p. 10, 2011.

[19] NIST/SEMATECH, e-Handbook of Statistical Methods, NIST/SEMATECH, 2003.

[20] K. K. S. S.Gopal Krishna Patro, “Normalization: A Preprocessing Stage,” *arXiv,* p. 4, 2015.

[21] D. Pedamonti, “Comparison of non-linear activation functions for deep neural networks on MNIST classification task,” *arXiv,* p. 5, 2018.

Appendix

I. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, **Marten Jaago**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
   Predicting Demand for Smart Parking Systems,
   mille juhendaja on Amnir Hadachi, PhD
   reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commonsi litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Tartu, **5/14/2019**