Long Short-Term Memory to predict parking space in Denmark

Alkharif Sarah-M2016426

GitHub site: sarah-Abdulaziz

Introduction

The smart parking is in smart city of IoT environment aims to facilitate users with accurate information regarding available parking spots. The prediction of available parking spots in next time steps for parking using deep learning can facilitate improved accuracy of occupancy status, efficient traffic management and better utilization of parking lots.

Datasets

We collected datasets for our project from citypulse website. A data-stream with parking data provided from the city of Aarhus. There are a total of 8 parking lots providing information over a period of 6 months (55.264 data points in total). From May 22, 2014 to November 4 2014.
When we visualized our dataset by take the average occupancy per hour, we noted that from morning until mid-day all the parking lots are almost occupied.

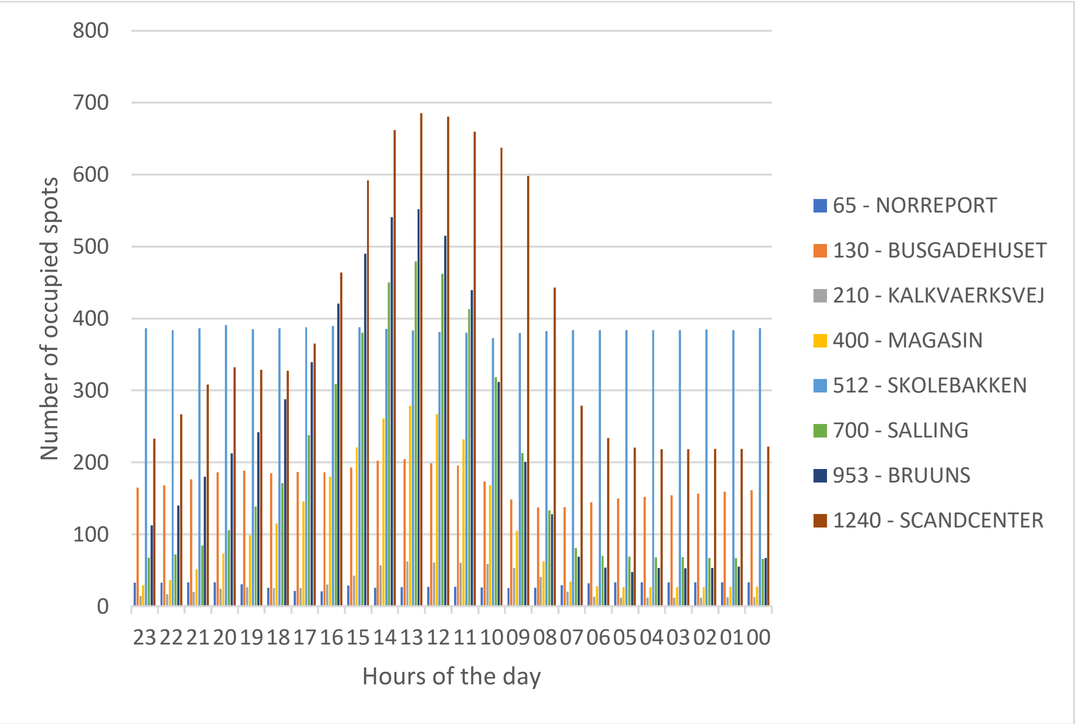


Figure 1. Occupancy of each parking lot for each hour of the day

Related work

In 2017, we apply statistical models for Time Series Prediction to predicted spot price from Amazon Web Services, using popular statistical models such as moving average (MA), auto-regressive (AR), auto-regressive integrated moving average (ARIMA), and seasonal ARIMA (SARIMA), our data set of spot price that content timestamp, spot price, availability zone, and instance type, we fetch 11 months from March. 2106 to Feb. 2017, for 18 zones, and we select the instance in General, GPU, Compute-, Memory-, and Storage-optimized types that are m4.2xlarge, g2.2xlarge, c3.2xlarge, r3.2xlarge, and i2.2xlarge.

We evaluate naive, seasonal naive, mean, seasonal ARIMA, linear regression, and Prophet algorithms. The linear regression and Prophet was the worse than other algorithms.

Figure 4 shows the test error of different algorithms. For seasonal Arima and mean, we select the best performing configuration values. Each algorithm is executed in all 18 availability zones.

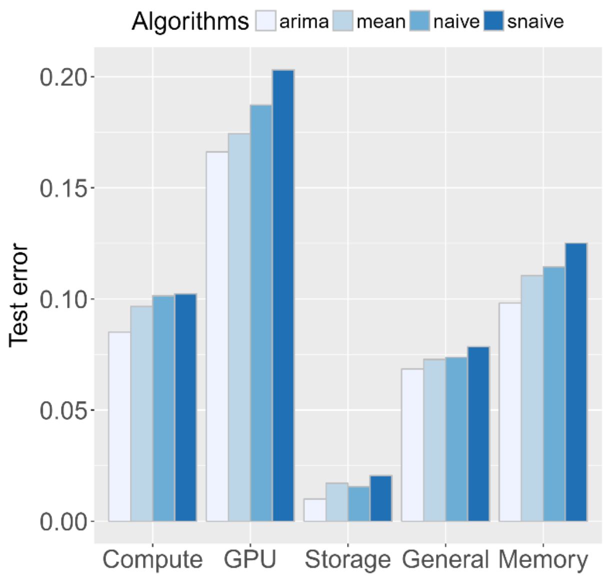


Figure2 . EC2 instance (-optimized)

In 2018, we try to predicted spot price from Amazon Web Services, by apply deep learning algorithms, to forecast spot market price by using Long Short-term Memory (LSTM) algorithm, after we apply cross validation technique for our data set, which help our model to learn deeply. We measure the performance with root-mean-square error (RMSE), and find that, for our 10 regions in our dataset, the error is decreasing more with the cross-validation and get the result much faster. We apply our data set to other models to compare their performance; we found that our model works more accurately.

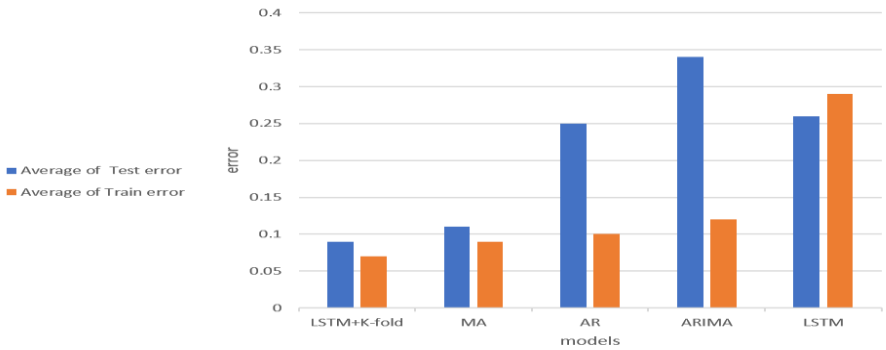


Figure3 . the Average Result for 3 months data set

Based on our result, we try to find availability zones that price does not become over on-demand price and less changing price over time, which help users to choose the most stable availability zones.

We try to work in real time by importing Amazon spot instance prices daily. With the data, we run our model to forecast for next 24 hours, and give users the availability zones whose prices do not become over a preset on-demand price and are less changing over time, which helps the users to choose the most stable availability zones. This application is called as Amazon cloud price prediction (ACPP), We publish our result in http://167.99.77.9/.

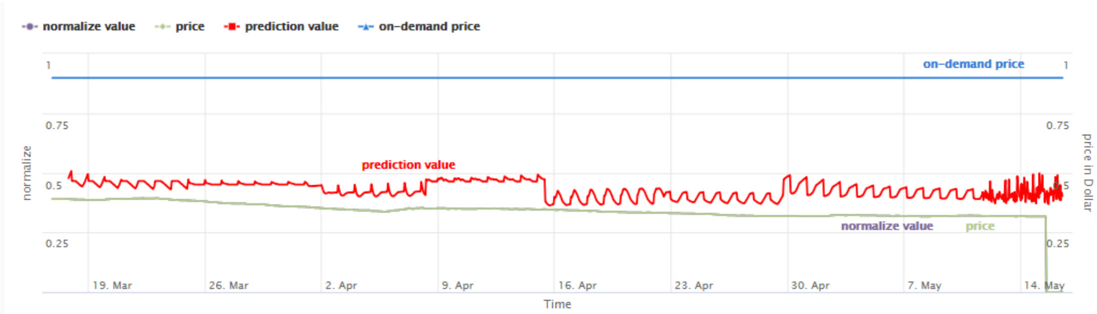


Figure 4 chart that show prediction for ACPP application with actual price

Method

Time series analysis algorithms try to predict data that change over time like spot weather forecast. In this is project we are trying to use sequence models in Neural Network algorithms that takes datasets as input X and learning from them in profoundly to get result Y as output by using this general formula:

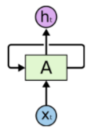
$$y = \delta(WX + B)$$

$$\delta = f(x) = \frac{1}{1 + z^{-x}}$$

where the activation function change based on data that we work with, w; weight and b; bias as parameters, we initialize these parameters randomly, the benefit of Neural Network is backward propagation stage where updating these parameters by compute the cost function and minimize the error using Gradient Descent algorithm.

$$cost\ function = \frac{1}{m}\sum L(y, \hat{y})$$

with applying Long Short-Term Memory network which is a special kind of Recurrent Neural Network that works with loop for everywhere to mimic human brains thinking by learning in practicing the previous



Recurrent Neural Networks have loops.

the problems appear when we deal with big data because the network will forget some relevant information that shown at the beginning of data, for this reason, Long Short-Term Memory networks (LSTM) introduced by Hochreiter & Schmidhuber (1997), and after we apply LSTM we will compare the prediction result with Statistical models for Time Series Prediction such as moving average (MA), auto-regressive (AR); that help us to measure the performance for our model.

Project time line