SciPro: Applications of Artificial Intelligence

Deep Learning for Short-term bike-sharing demand prediction: LSTM and CNN-LSTM

Submitted by:

Tri Dung Huynh
Matr.-Nr.: 235326
Email: tri.huynh@st.ovgu.de

&

Md Asaduzzaman
Matr.-Nr.: 226255
Email: md.asaduzzaman@st.ovgu.de

Supervisor:

Jun.-Prof. Dr. Kai Heinrich

Abstract

The bike imbalance issue may lead to decreased service reliability, user discontent, and decreased attraction and user commitment in the bike-sharing program, failing to achieve sustainable transportation system implementation expectations.”- which may require accurate forecasting of convenient demand. These studies have shown how weather conditions can affect how many bikes are demanded at bike-sharing stations at times of the day and weeks. It will be easier for operating agencies to rebalance bike-sharing systems in a timely and efficient manner if we can predict the short-term demand for bike-sharing. We use Helsinki city historical trip data from 2016 to 2020, along with extra inputs such as weather conditions, demand-related engineered features, and temporal variables related to demand. As opposed to statistical methods, deep learning methods can automatically learn the relationship between inputs and outputs, requiring fewer assumptions and achieving higher accuracy. Since their spatiotemporal functional observations are similar, the CNN and LSTM need to be segmented into time frames like hourly, daily, weekly, and monthly, in order to model the Helsinki bike-sharing station data effectively. We provide quantitative results show that accurate short-term demand (CNN-LSTM_336) was noticeable between the Models.

Key Words: *Bike sharing system; deep learning model; ARIMA model; Convolutional Neural Network (CNN); Long Short-Term Memory Network (LSTM)*

Introduction

To meet the growing need for public transport in cities, more and more transportation solutions were proposed and implemented, notably convenient and environmentally friendly solutions. The bike-sharing system is one of the most attractive solutions for all-size cities or urban areas. The main premise of implementing bike-sharing systems is to promote sustainable mobility in urban areas. They offer convenient and easy-to-use services for residents for short-distance trips. Moreover, they can improve first/last mile connection to other travel modes, reducing traffic congestion and energy consumption, and decreasing the environmental acts of daily travel. Furthermore, communities that organize bike-sharing programs increase physical activity and encourage remarkable health benefits to the users. In more congested areas, bike-sharing has a positive economic impact by saving time for finding parking and more time for patronizing the local stores.

However, the success of the model lies not only in infrastructure investment but also in convenience. Users can always find bikes near their starting position when they need them. Therefore, an essential question in planning and designing bike-sharing services is to support the user's travel demand by allocating bikes at the stations in an efficient and reliable manner which may require accurate short-time demand prediction. Some proposed models, from statistical to advanced deep learning models, were used to predict the short-time demand for bike-sharing systems. Notwithstanding, there is still no consensus on which model achieves the highest accuracy. In our study, we will re-implement several forecasting models, including ARIMA, LSTM, and CNN-LSTM for Helsinki City bike data from 2016 to 2020 and compare their accuracy.

Literature review

Along with the rapid development of technology in recent decades, many electronic devices and sensors have been created, giving us the ability to collect more data from our daily behaviour. Spatiotemporal data (STD) contain both space and time information. Spatiotemporal data has become a broad domain in ecology and environmental management, public safety, transportation, earth science, epidemiology, and climatology to discover patterns and knowledge (Jiang et al., 2019). ST data can be categorized into the following types: event, trajectory, point reference, raster, and video. In which event STD contains information on the type of event, time, and location where the event happens. Trajectory data contains the time, location, and moving sequence of the subjects, such as bike-sharing data. Point reference data contains a set of moving reference points in a specific space and time, like weather balloons record temperature and humidity. Raster data is similar to point reference data but with a fixed location. Video data is a sequence of images with the pixel in each image considered as spatial information (Wang et al., 2020).

In recent years, more and more researchers have used predictive learning methods for STD and discovered the great potential for application (Atluri et al., 2018). Graves and Schmidhuber (2009); Mikolov et al. (2010) use recurrent neural networks to help classify words or sentences in human speech recognition problems. At the same time, deep convolutional neural networks (CNN) were used with spatial maps as input features to predict the output at a time step of the ST raster in computer visioning (Krizhevsky et al. 2012; LeCun and Bengio 1995). Bahadori et al. (2014), Yu et al. (2015), and Zhou et al. (2013) reduced the model complexity using the spatial and temporal dependencies among the input features in tensor learning-based approaches.

Trajectory STD was used widely in research about human behavior and solving traffic problems (Kong et al., 2018). In early research, Castro et al. (2013) studied traffic dynamics using large-scale taxi pickup and drop-off data. The data includes essential information about each trip, such as the time and the GPS data of pickup and drop-off locations. Caulfiled et al. (2017) use the logistic regression model to examine the usage trends of bike-sharing in a small city by patterns such as frequency of usage, temperature, distance traveled, and time. A deep learning model Spatio-temporal graph neural network (STGCN) was used to predict traffic congestion using the data collected from Bluetooth sensors of passing cars in the study by Cunnov et al. (2021). However, both standard model ARMA and simply averaging outperforms the proposed model in this setup. A combination of k-nearest neighbor and LSTM was used to predict the traffic flow and achieve promising results compared with autoregressive integrated moving average (ARIMA), support vector regression (SVR), wavelet neural network (WNN), deep belief networks combined with support vector regression (DBN-SVR), and LSTM models in Luo et al. (2019) works. Specifically, for short-term prediction of bike-sharing demand, Ma et al. (2022) propose spatial-temporal graph attentional long short-term memory (STGA-LSTM), and the model received higher accuracy than the baseline models. At the same time, Mehdizadeh et al. (2022) apply the hybrid model of convolutional neural network (CNN) and long short-term memory (LSTM) to predict pickup demand for shared bikes in Montreal.

Methodology

3.1. ARIMA

Auto Regressive Integrated Moving Average (ARIMA) is a generalized model that combines Autoregressive (AR) process and moving average (MA) processes and builds a composite model of the time series. As the acronym indicates, ARIMA has the following parameters (p), (d), and (q). (p) describes the dependencies between an observation and a number of lagged observations; (d) is the differences of observations at a different time; (q) shows the dependency between observations and the residual error terms when a moving average model is used for the lagged observations (Siami-Namini et al., 2018).

3.2. Long Short-Term Memory (LSTM)

Recurrent Neural Networks are networks of recurrent connections; however, they have the limitation for lookback time of generally ten timesteps (Staudemeyer et al., 2019; Mozer, 1991). These restrictions occur due to vanishing or exploding problems of insufficient recurrent backpropagation (Graves, 2012).

LSTM is a particular recurrent neural network capable of remembering the values from earlier stages and forgetting unnecessary trends in each network cell (Ojo et al., 2019). LSTM can learn over 1000 discrete-time steps by allowing constant error carousels through special cells in layers (Hochreiter et al., 1997). The fundamental model consists of three layers: an input layer, a hidden layer, and an output layer. These multiple hidden layers act as a Deep Recurrent Neural Network. The hidden layer of an LSTM network contains memory cells that contain three gates responsible for updates to its cell state. These three gates are an input gate, an output gate, and a forget gate. Figure 1 visualizes the connection of these gates in a single LSTM cell. The math equations for input gate, output gate, forget gate, and cell state at tth time interval, respectively.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

$$f_t = \sigma\left(w_f[h_{t-1}, x_t] + b_f\right)$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_0)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tanh(w_c[h_{t-1}, x_t] + b_c)$$

In which i_t stands for the input gate, f_t stands for the forget gate, o_t stands for the output gate, c_t stands for the cell state at the t^{th} time interval, w stands for the weight for respective gate neurons, h_{t-1} stands for the output of the previous LSTM block at time t−1, x_t y stands for the input matrix at the t^{th} time interval, and b stands for biases for respective gates. ⊗ denotes element-wise multiplication with the exact dimensions. σ and *tanh* are the hyperbolic tangent activation functions (Mehdizadeh et al., 2022).

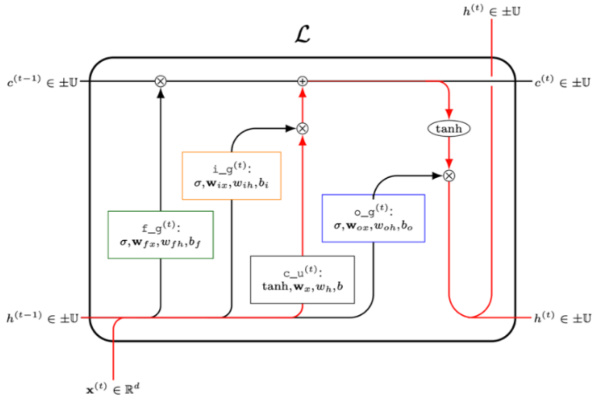

$h^{(t)} \in \pm\mathbb{U}$
$\mathcal{L}$
$c^{(t-1)} \in \pm\mathbb{U}$
$c^{(t)} \in \pm\mathbb{U}$
tanh
i_g$^{(t)}$: $\sigma, \mathbf{w}_{ix}, w_{ih}, b_i$
f_g$^{(t)}$: $\sigma, \mathbf{w}_{fx}, w_{fh}, b_f$
o_g$^{(t)}$: $\sigma, \mathbf{w}_{ox}, w_{oh}, b_o$
c_u$^{(t)}$: $\tanh, \mathbf{w}_x, w_h, b$
$h^{(t-1)} \in \pm\mathbb{U}$
$h^{(t)} \in \pm\mathbb{U}$
$\mathbf{x}^{(t)} \in \mathbb{R}^d$

Figure 1. A graphical illustration of a single LSTM cell. The red arrows show the recurrent neural unit of two tanh activation functions. (Elsworth et al., 2020)

3.3. Convolution neural network (CNN)

A convolutional neural network is one of the most used algorithms in the deep learning computing paradigm. It can manage a large amount of data and be able to discover meaningful features automatically without relationship instruction. CNNs are modeled after the structures of human and animal brains, similar to a conventional neural network. In CNN, the weight-sharing features increase the generalization and avoid overfitting by lessening the number of trainable parameters. (Alzubaidi et al., 2021)

The convolutional layer is the most critical constituent in CNN architecture. It contains a group of kernels (convolutional filters). In each kernel, the parameter weights are stored in a grid of discrete numbers and will be adjusted in the learning process. These kernels help determine the main features contributing to the model's outcome.

Pooling layer: The utmost function of the pooling layer is to generate and maintain the subset of input features. These processes will reduce the dimensions of extensive feature maps and keep prominent features in each step of the pooling process. There are several pooling strategies that can be used in different pooling levels. These techniques include global average pooling (GAP), global max pooling, global average pooling, gated pooling, and average pooling. The max, min, and GAP pooling techniques in figure 2 are the most well-known and widely used pooling techniques.

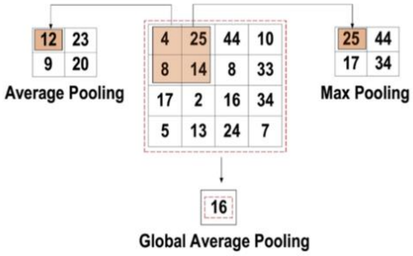


Figure 2. Three types of pooling operations (Alzubaidi et al., 2021)

Algorithms and Experiments

In this section, we discuss the different configurations investigated for comparative studies.

4.1. Dataset Description

Bike-sharing Data

The bike-sharing dataset was provided by the local traffic agency Helsinki Region Transport (HRT), the operator of the bike-sharing system in Helsinki. City Bike Finland is a local company maintaining and rebalancing the shared bikes in the city. The dataset contains 12.138.008 entries with 14 attributes such as departure time, return time, departure id, departure name, distance, duration, avg speed, departure latitude, departure longitude, return latitude, return longitude, air temperature of Helsinki bike-sharing system, and so on. Each entry is a record of pickup information in the whole system from 02.5.2016 to 01.11.2020. We clean the data by removing the outliner based on the trip's distance (smaller 50m and over 10km), duration, and temperature at pickup time. The remaining data is 11.278.850 entries from 347 stations in Helsinki.

Weather Data

Visual Crossing Weather provided the Weather Dataset used in this work. Historical hourly weather data were extracted from Weather Underground. Temperature, wind speed, humidity, precipitation, and overall weather conditions were the most common factors for predicting bike-sharing demand. The data includes a single row of header information indicating the information in the column. The total number of records in the raw data was 39443 rows and 18 columns. (Visual crossing weather, 2020) The dataset includes features such as ‘Location’ and ‘Address’ which are diminished because the weather information covers the area of Helsinki. The features ‘Snow Depth’, ‘Wind dir’, and ‘Heat Index’ kept more than 50% of missing data or null data and are removed from the dataset, as the existing data is inaugurated to calculate any valid approximations numbers to replace the missing values. (Jessica Quach, 2008) ‘Wind Gust’, containing 30% missing values, is erased from the dataset while cleaning the dataset. The feature ‘conditions’, or weather types, also contain any of the values- clear, partially cloudy, rain, overcast, or ‘rain, partially cloudy’. The values of the feature are categorical and cannot be measured in a numerical way. Additionally, the values of this feature can be accounted for by the features ‘icon’ and ‘Soler radiation’ (Jessica Quach, 2008). In summary, ten features are accumulated for consideration as weather components for further preprocessing. Those are datetime, max-min and average temperature, dew, humidity, precipitation, windspeed, cloud cover, visibility, and solar energy.

4.2. Data Pre-processing

Therefore, the first step was selecting & filtering some of the common data problems are missing data, duplicate rows, and column types. Both data sets were checked for these problems, with a slight variation between the two data sets given in their respective subsection. From the original bike-sharing dataset, we calculate the hourly number of pickups at each station. The timestamp is used as an index for bike-sharing and weather data, which helps to concatenate the dataset for prediction models. Hourly aggregation comes in a uniform format after processing for both datasets.

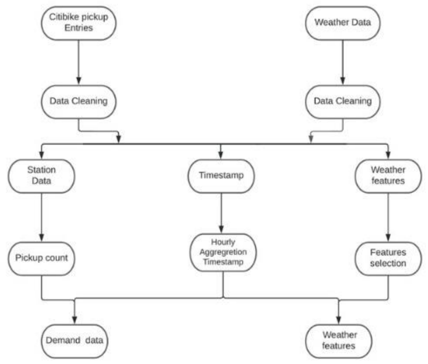


Figure 3. Data Preprocessing Framework

In the weather features, “Temperature”, “Maximum Temperature,” and “Minimum Temperature” are strongly correlated. It also strongly correlates with “Average number of Pickup”. We remove the features for maximum and minimum temperature and keep the average temperature “Temperature.” The reason for keeping the feature “Temperature(temp)” seems to have the highest correlation with “Avg number of pickups” followed by the feature “dew.” As a result of data preparation, a dataset is produced that includes 39443 observations and eight features, where each observation holds a record for a day. This dataset has been cleaned and processed into one dataset ready for further analysis.

4.3 Data Structure Design

Hourly aggregation demand of each station was processed into a demand data matrix as presented in the matrix (1). In which, D_t^c denotes the number of pickup bikes at station c and t^{th} time interval. We have 347 stations and 39.442 historical time intervals in our demand data. In weather matrix (2), W_t^f denotes the weather condition f at t^{th} time interval. From the analysis results of weather data obtained from Helsinki, we select seven crucial weather characteristics to use as weather features to train the prediction models. Chosen weather features include temperature, dew point, humidity, precipitation, wind speed, cloud cover, visibility, and solar energy. Additionally, the time interval for weather data was matched with time intervals of demand data.

$$\text{Demand data } = \begin{bmatrix} D_0^0 & D_0^1 & \cdots & D_0^c \\ D_1^0 & D_1^1 & \cdots & D_1^c \\ D_2^0 & D_2^1 & \cdots & D_2^c \\ \dots & \cdots & \cdots & \dots \\ D_{T-1}^0 & D_{T-1}^1 & \dots & D_{T-1}^c \\ D_T^0 & D_T^1 & \cdots & D_T^c \end{bmatrix} \quad (1)$$

$$\text{Weather Data } = \begin{bmatrix} W_0^0 & W_0^1 & \dots & W_0^f \\ W_1^0 & W_1^1 & \dots & W_1^f \\ W_2^0 & W_2^1 & \dots & W_2^f \\ \dots & \dots & \dots & \dots \\ W_{T-1}^0 & W_{T-1}^1 & \dots & W_{T-1}^f \\ W_T^0 & W_T^1 & \dots & W_T^f \end{bmatrix} \quad (2)$$

Input Data

To build the input data for the prediction models, we take the pickup demand at each station and stack it with shifted data for historical features. The number of historical demand features was decided by the number of "lags". Each lag represents 1 hour. We tested the performance of models with lags value: 12, 24 (1 day), 48 (2 days), 96 (3 days), 120 (5 days), 168 (1 week), 336 (2 weeks), 504 (3 weeks), 672 (4 weeks). For the weather features, we take the selected features at t-1, t, and forecasting weather conditions for the next 3 hours: t+1, t+2, and t+3. The feature builder processes are illustrated in figure 5.

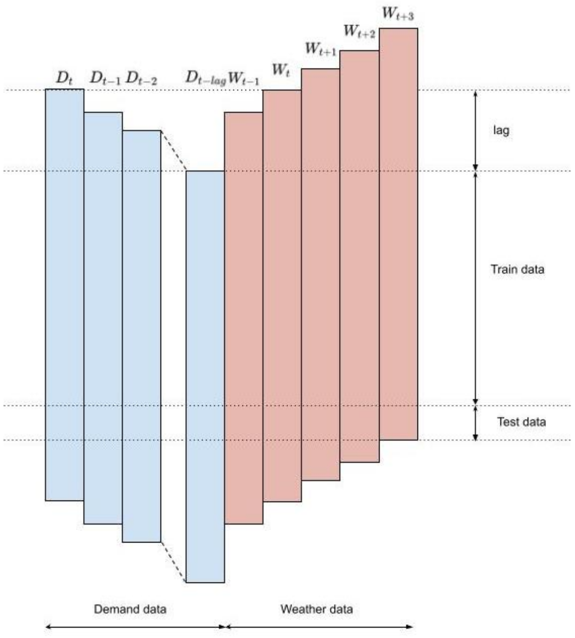


Figure 4. A graphical illustration of feature builders for training and testing data

In addition, during the data exploration, we discovered significant differences in the timing of the first pickup at stations. We, therefore, anticipate that this happens due to the growth and expansion of the bike sharing system. Therefore, we start timing from the first pickup and divide the collected data into 80% for training and 20% for testing. Mathematically, the input data for prediction models is presented in the matrix (3).

$$\text{Input } = \begin{bmatrix} D_t^c & D_{t-1}^c & \dots & D_{t-lag}^c & W_{t-1}^1 & \dots & W_{t-1}^f & W_t^1 & \dots & W_t^f & \dots & W_{t+3}^1 & \dots & W_{t+3}^f \\ D_{t+1}^c & D_t^c & \dots & D_{t-lag+1}^c & W_t^1 & \dots & W_t^f & W_{t+1}^1 & \dots & W_{t+1}^f & \dots & W_{t+4}^1 & \dots & W_{t+4}^f \\ D_{t+2}^c & D_{t+1}^c & \dots & D_{t-lag+2}^c & W_{t+1}^1 & \dots & W_{t+1}^f & W_{t+2}^1 & \dots & W_{t+2}^f & \dots & W_{t+5}^1 & \dots & W_{t+5}^f \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ D_{T-1}^c & D_{T-2}^c & \dots & D_{T-lag-1}^c & W_{T-2}^1 & \dots & W_{T-2}^f & W_{T-1}^1 & \dots & W_{T-1}^f & \dots & W_{T+2}^1 & \dots & W_{T+2}^f \\ D_T^c & D_{T-1}^c & \dots & D_{T-lag}^c & W_{T-1}^1 & \dots & W_{T-1}^f & W_T^1 & \dots & W_T^f & \dots & W_{T+3}^1 & \dots & W_{T+3}^f \end{bmatrix} \quad (3)$$

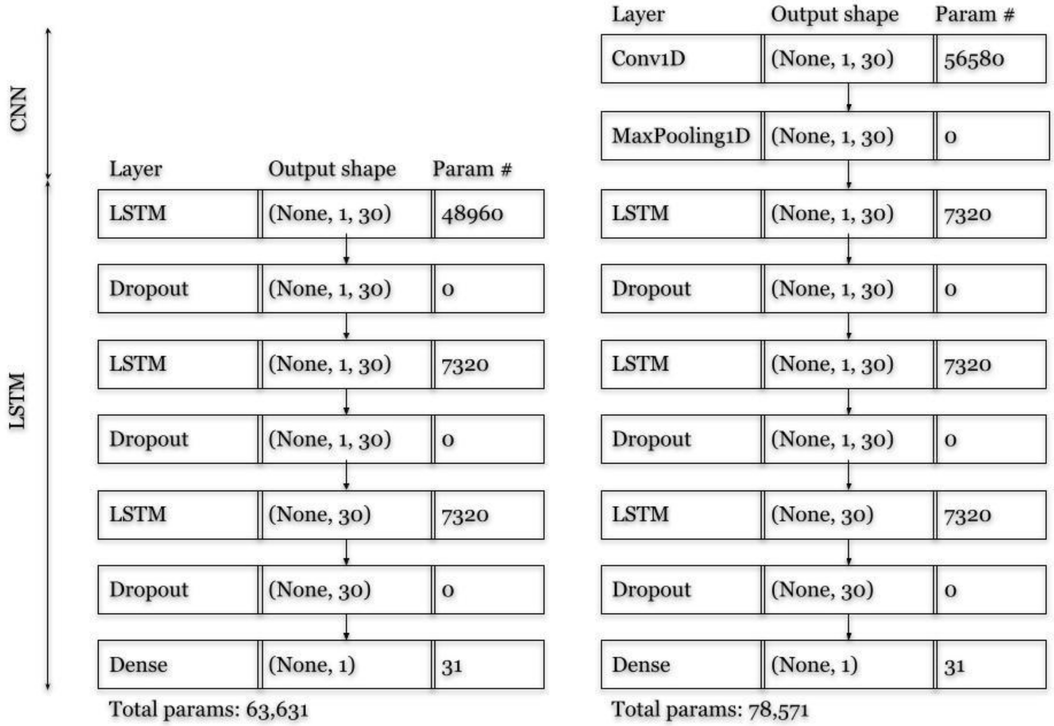
Output Data

The crucial factor for the bike-sharing system's success is the bike availability at the station. Howbeit, we need vehicles to relocate bikes from low demand points to high demand points. It is, therefore, necessary to predict the demand at each station ahead. Hence, we will focus on determining which model predicts the most accurately and stably for short-term demand at the station in the time window of two hours. We decided to choose 2 hours as we believe this is the time it takes for the system to be able to arrange vehicles between stations. The target result is the total pickups of the next 2 hours at the respective station. The output array can be presented as:

$$y^c = [D^c_{t+1} + D^c_{t+2}, \quad D^c_{t+2} + D^c_{t+3}, \quad \dots, \quad D^c_T + D_{T+1}, \quad D^c_{T+1} + D^c_{T+2}]$$

4.4 Deep Learning Prediction Models: LSTM and CNN-LSTM

Our project set up different model configurations to investigate the performance of LSTM and CNN-LSTM to predict the short-term pickup demand for each bike-sharing station. Figure 5 illustrates the layer structure of LSTM and CNN-LSTM. With consideration of 2 weeks time interval, the machine must train 78.551 and 63.631 parameters for CNN-LSTM and LSTM models, respectively. In the LSTM structure, we mixed the conventional LSTM layers and dropout layers to prevent the overfitting problem on training data. We stacked the 1D convolution layer to create a convolutional kernel and a Maxpooling 1D with an LSTM structure to form the hybrid CNN-LSTM model. We change the number of lag intervals and the prediction of weather data to experiment with the relationship of these factors to the prediction accuracy of the models.



Layer	Output shape	Param #
LSTM	(None, 1, 30)	48960
Dropout	(None, 1, 30)	0
LSTM	(None, 1, 30)	7320
Dropout	(None, 1, 30)	0
LSTM	(None, 30)	7320
Dropout	(None, 30)	0
Dense	(None, 1)	31

Total params: 63,631

Layer	Output shape	Param #
Conv1D	(None, 1, 30)	56580
MaxPooling1D	(None, 1, 30)	0
LSTM	(None, 1, 30)	7320
Dropout	(None, 1, 30)	0
LSTM	(None, 1, 30)	7320
Dropout	(None, 1, 30)	0
LSTM	(None, 30)	7320
Dropout	(None, 30)	0
Dense	(None, 1)	31

Total params: 78,571

Figure 5. The layer structure of LSTM and CNN-LSTM models with the number of parameters for 336 hours of lags (2 weeks)

Experimental results and discussion

5.1. Descriptive Analysis

The descriptive analysis of demand and weather data is consecutively drawn by several figures while visualizing the dataset.

Demand data

The definitions of variables and data summary are listed in Tables. HSL Bike corporation counted bicycles from May 2016 to November 2020. this organization gathered 80% of the bicycle demand counts from May to October. The monthly (2017) bicycle count data figure is presented in Figure 9. More bicycle counts are observed during peak months, followed by summertime. To clarify, the average counts in 2017, May, June, July, August, September, and October are 10.12, 10.58, 9.59, 9.58, 7.59, and 5.06. Bicycle counts, as expected, reach their peak in July and their trough in January. As shown in the Figure, the yearly demand for bike sharing in Helsinki city is much more than the previous year. In 2016, the average pickup demand per year was max. 10.54, which is gradually increasing in 2017 is 43.17, in 2018 is 67.80, in 2019 is 86.99 and finally in 2020 is 73.74, slightly decreasing due to the pandemic Covid attacked by worldwide.

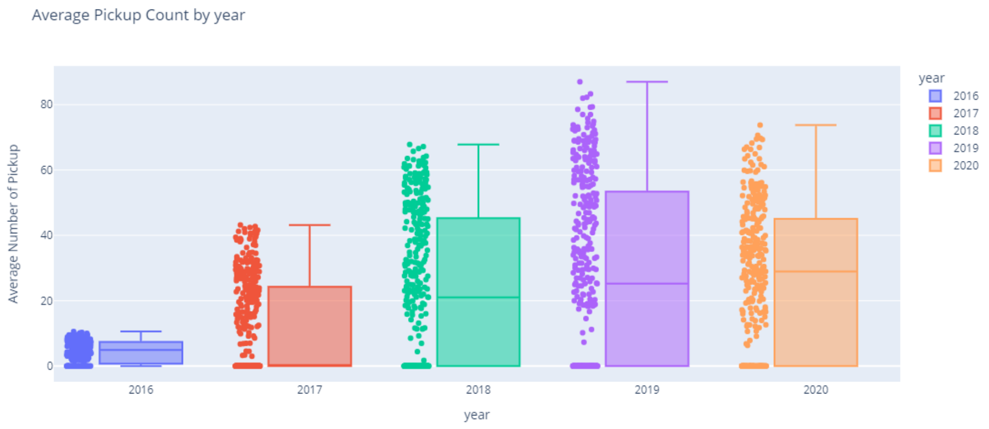


Figure 6. Average Pickup Count by Year

We take a box-and-whisker plot to describe the mean and max demand of 347 stations in Helsinki city. Figure 7 illustrates the distribution of the mean and max demand of bike-sharing stations. The upper whisker, upper hinge, median, lower hinge, and lower whisker are respectively 3.234,1.567,0.772,0.389, and 0.110. The interquartile range (IQR) is measured by the difference between the upper and lower quartiles, which are here 1.178. There is a favorable skewed distribution because the whisker and half-box are longer on the right side of the median than on the left. The value looked beyond the inner face denotes Outliers. All are seeming here mild outliers. Station Itämerentori shows the max outlier range with pointing 7.794.

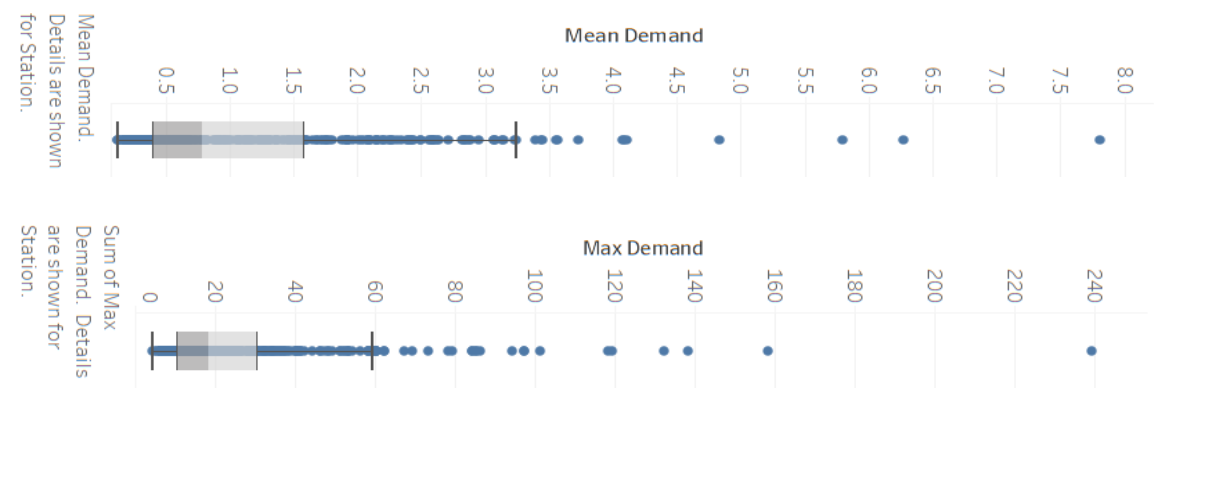


Figure 7. Mean and Maximum demand of all stations

The maximum and minimum demand noted in this analysis at station Kaivopuisto is 239, and Itäkeskus Metrovarikko is 4, respectively. The upper whisker, upper hinge, median, lower hinge, and lower whisker are respectively 59,30,18,10, and 4. The interquartile range (IQR) is measured by the difference between the upper and lower quartiles, which is 20. Station Kaivopuisto showed the highest outlier in this analysis.

Weather data

From the statistical point of view, weather data, in contrast to average weather pickup demand, have a positive correlation. The average temperature records a slight decrease in 2017 (around -18.5°C), contrarily to the rainfall and relative humidity that display in noticeable increase, the intra-seasonal variation goes in concordance with the Visual crossing report. Where consecutively 2016,2018,2019and 2020 show a higher temperature than the year 2017. Such lousy weather is decreasing the demand for bike picking in Helsinki, are mostly had some bad weather. The figure shows the maximum and minimum temperature with median, Quartile 1, and Quartile 3. the figure shows an excellent overview of how the demand was created in the past year and how it affects the demand due to concur weather. Maximum weather was recorded in 2018(27.7-degree Celcius.),2019(26.1-degree Cel.), and 2020(25.15-degree Cel.) with a median of 7, 6.55, and 9.95, respectively. Most ranges have an asymmetric distribution since the median is not in the middle. It is worth to state the higher the temperature, the chances of pickup bike demand increasing and vice versa. The relative humidity varies according to the precipitation rate. Most ranges have an asymmetric distribution since the median is not in the middle. Relative humidity in winter of 2018 and 2019 is close to the box limit, so 50% is close to the value of Q3(See appendix). Windspeed and Soler energy are also important factors in bike sharing demand.

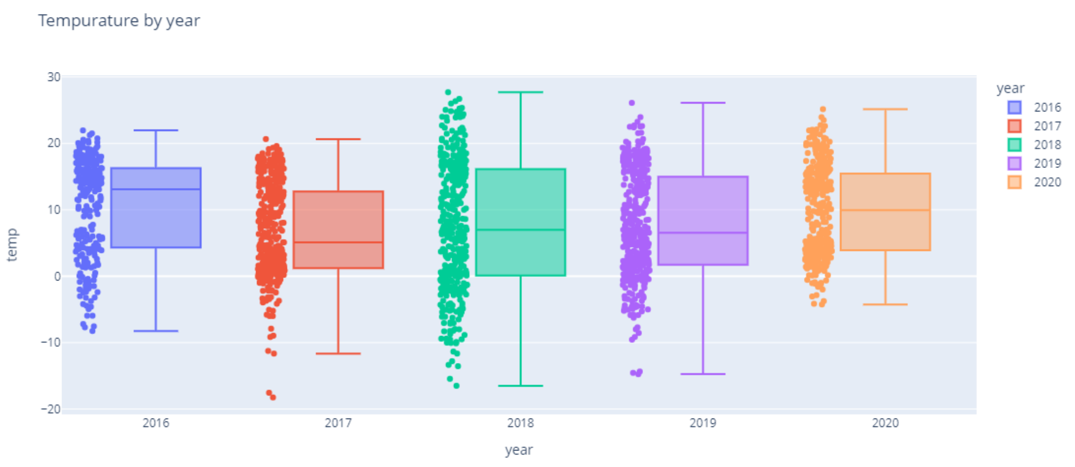


Figure 8. Temperature Count by year

Aggregation of Datasets

Separately, both datasets have been processed and cleaned. By using the timestamp as a baseline, we converted station-wise pickup data into an average number of pickups. In order to determine which of the weather features have the biggest impact on bike usage, we examine the correlation coefficients between the weather features in the dataset, as shown in Figure 8. The performance of models is enhanced when features are strongly correlated with one another. In Figure 8 indicates a strong positive correlation among features, 0 indicates no correlation between two features, and -1.0 indicates a strong negative correlation between two features (Jessica Quach, 2008).

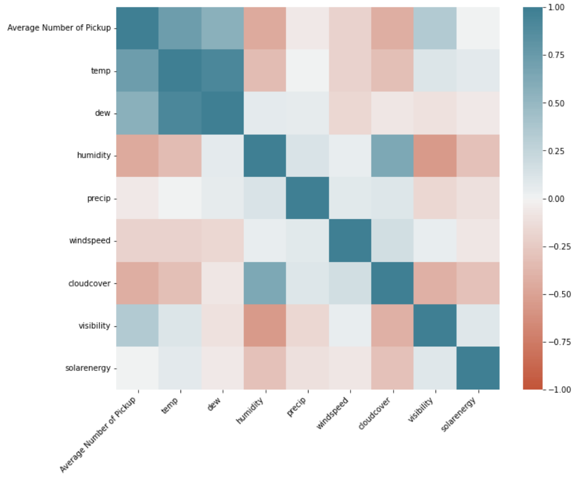


Figure 9. Merged data set Features Correlation

As shown in Figure, the pickup duration distribution is upbringing in the middle, with the mode being around by month. In contrast, the average pickup count is affected by temperature for several years. The highest density can be observed between 4 to 9 months. Closer inspection of the pick duration trend by months, Figure 10 shows that there has been a steady decline from October to march each year, a gradual increase reaching its peak in April, then it drops again from October each year. The trip duration variability is most likely attributed to weather conditions (Temp, windspeed), as bike users tend to cycle more in warm conditions. Figure 10 confirms it; the same pattern can be seen, the gradual decline from January to February, then the continual growth reaching a peak in September.

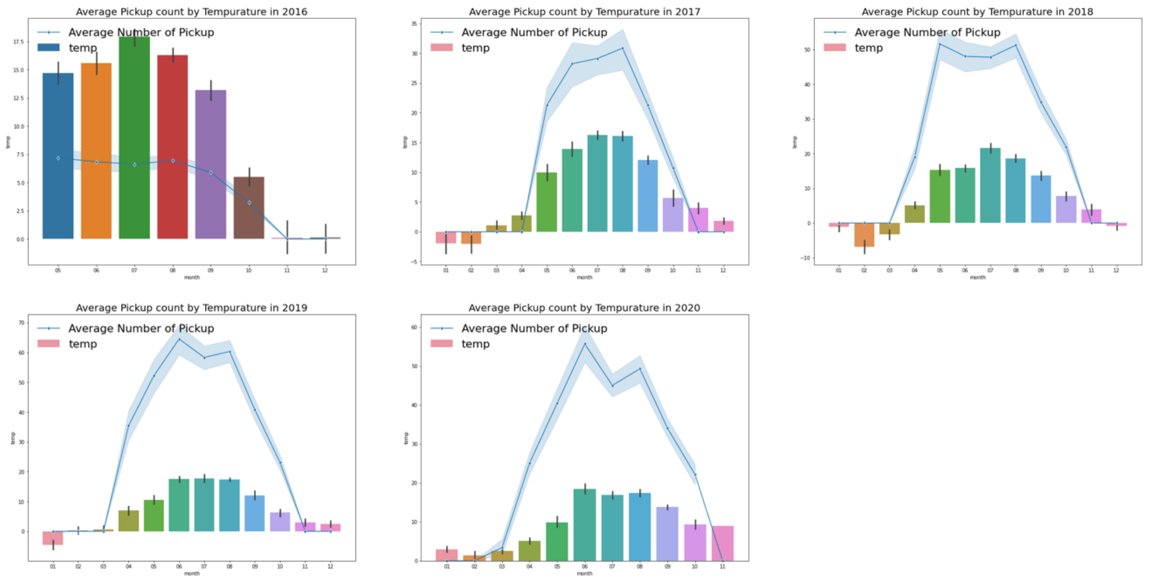


Figure 10. Average Pickup Count in contrast to temperature by month & year

5.2 Stationary

This paper requires the analysis of both datasets from a statistical point of view. With this view, it was checked out that both datasets are stationary or non-stationary to implement the ARIMA model. It is checked by Plotting the ACF and PACF for both data sets. Also, use the Augmented Dickey-Fuller (ADF) test to check if both datasets are stationary or not. According to the ADF test, the time series is stationary. We can reject the hypothesis if the p-value is less than the significance level (0.05), thus indicating that the time series is stationary the result comes out with all 347-station data being stationary and the weather data being stationary. As in our case, it is unable to build ARIMA model for this research of predicting bike sharing demand. However, there is another forecasting model, ARMA, which is better fit because the datasets are stationary.

$$ARIMA\ (p, d, q) = ARMA\ (p, q) + \ preliminary\ differencing\ procedure\ (d)$$

An ARIMA (p, d, q) model is essentially an ARMA (p, q) model with roots in the d units. From its formula, it is easy to see:

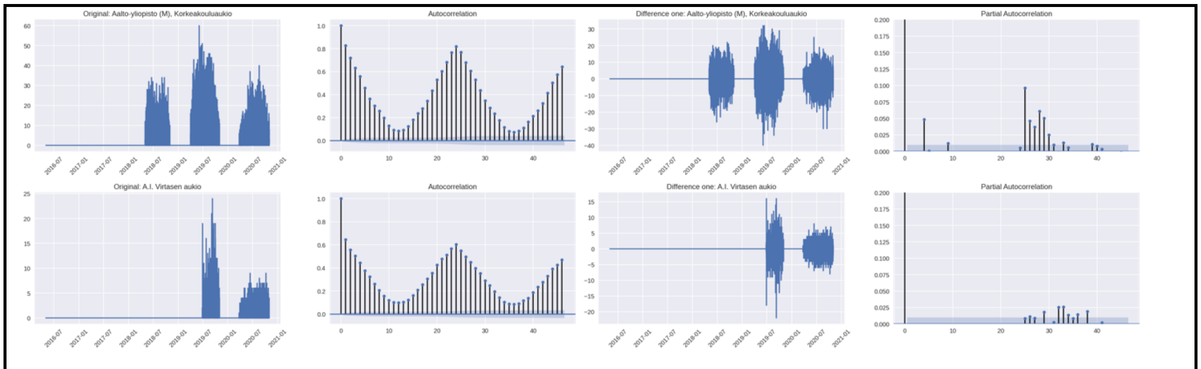


Figure 11. ACF & PACF of station different stations

For the ARMA method, ACF and PACF clearly showed that the curves differ at each station in Figure 11. Eventually, for 347 stations, the method can be generalized for any models but at the cost of increased computational time. So, we put it in cue and turn our thought from the traditional statistical model to the deep learning model. However, the model (CNN&LSTM) order estimation accuracy is much better than it is for most of the aforementioned methods (ARMA & ARIMA) (Lei Ji, 2019). So, we ultimately chose the model CNN-LSTM and LSTM for further process.

5.3. Time Lags with LSTM and CNN-LSTM in Prediction Accuracy

Metric selection

We use mean absolute error (MAE) to measure the differences between actual and prediction values. Root mean squared error (RMSE) represents the root of the average squared difference between the original and predicted values. We also use it to measure the differences between actual and prediction values but amplify the effect of significant differences. However, our model not only helps to predict the demand of a particular station but of many stations and the demand of different stations are completely distinct. The median of average pickups at each station per hour of 347 stations is 0.772. Therefore, even small MAE and RMSE are not enough to conclude the performance of models. Meanwhile, R-squared represents the proportion of the variance for dependent variables. It could help explain the strength of the relationship between actual and prediction values. For mentioned reasons, we use R-squared as the primary metric to measure the accuracy of the models.

LSTM vs. CNN-LSTM

Due to the limited computing resources, we could only run our model for 200 stations. Therefore, the results from this part will apply to only 200 stations. We ran 20 models on Colab Pro+ of Google with 51GB of RAM. Each model needs 8 to 14 hours to complete the training and evaluate prediction results. Table 1 reports our testing results on average R-squared, average MAE, and average RMSE of 200 training stations. Comparing the corresponding results of the same time lag between LSTM and CNN-LSTM models, we see that 8 out of 10 models of CNN-LSTM give better average R-squared results. Although, for the time of 504 hours (3 weeks) and 672 hours (4 weeks), CNN-LSTM models perform lower average r-squared, these two models still give better RSME (smaller) values for CNN-LSTM. On this basis, we conclude that the hybrid CNN-LSTM model outperforms the conventional LSTM model in predicting short-term demand in the bike-sharing system for our setup. An explanation for this result could be that the CNN layer provides more robustness to capture parameters of training features for the conventional model.

Models	Average R-Squared	Average MAE	Average RMSE
LSTM 6	0.260	1.652	2.623
LSTM 12	0.281	1.638	2.561
LSTM 24	0.369	1.448	2.344
LSTM 48	0.381	1.429	2.328
LSTM 96	0.391	1.421	2.315
LSTM 120	0.395	1.412	2.303
LSTM 168	0.421	1.352	2.237
LSTM 336	0.425	1.348	2.230
LSTM 504	0.427	1.360	2.243
LSTM 672	0.416	1.379	2.281
CNN-LSTM 6	0.281	1.617	2.568
CNN-LSTM 12	0.272	1.632	2.559
CNN-LSTM 24	0.372	1.448	2.342
CNN-LSTM 48	0.388	1.419	2.306

CNN-LSTM 96	0.392	1.399	2.283
CNN-LSTM 120	0.401	1.390	2.273
CNN-LSTM 168	0.429	1.323	2.186
CNN-LSTM 336	0.429	1.322	2.182
CNN-LSTM 504	0.416	1.335	2.204
CNN-LSTM 672	0.411	1.339	2.220

Table 1. Prediction model testing results for 200 stations.
The green box indicates for highest average r-squared, the orange box indicates for lowest average MAE, and blue box indicates for lowest RMSE.

Time lags

Figure 12 details the distribution of the R-squared value at each station in each prediction model. Based on table 1 and figure 12, we can observe that both the mean and median of all stations increase for both LSTM and CNN-LSTM as we increase the value of the lag time interval from 6 hours to 336 hours. This observation implies that the models will predict more accurately as we increase the time interval considered in the model's features. However, when the optimal value is reached, the model's performance will start to decrease. This can be explained that when the model reaches the optimal lag, increasing the lag value will increase the confounding information to the model. Based on the results obtained from the experiment, the optimal value for the historical demand period is 336 hours or two weeks.

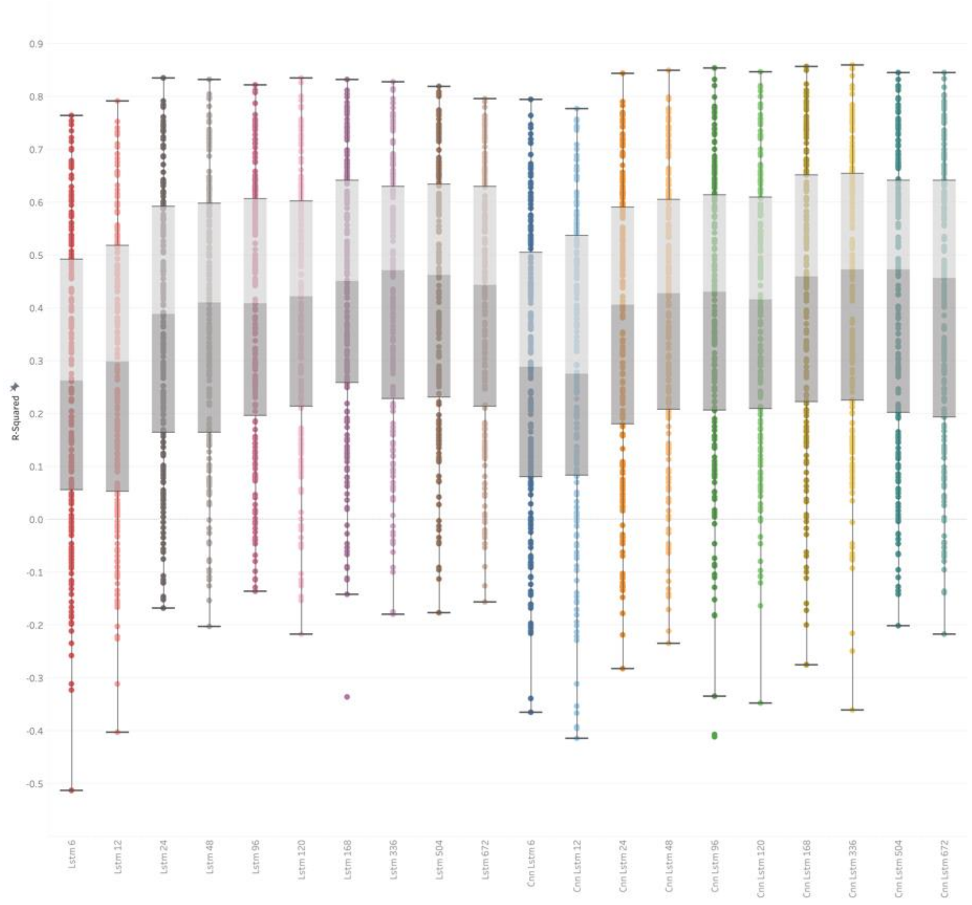


Figure 12. Boxplot distribution of R-Squared results for 200 stations

5.4. Average Pickups Demand in Prediction Accuracy

For the purpose of clustering stations, we use box-and-whisker to identify the distribution of the average demand of 200 stations. The upper whisker, upper hinge, median, lower hinge, and lower whisker are 3.435, 1.632, 0.875, 0.390, and 0.11, respectively. We use the upper and lower hinge values to divide stations into high demand, medium, and low demand groups. Hence 50% of stations will be in medium demand and 25% high and 25% on low demand.

We calculate the average of their mean demand for each demand group and compare it with the RMSE received from training our models for each hour. Figure 13 visualizes the results for CNN-LSTM 336 model. For high-demand stations, the average RMSE per hour is smaller than the average of mean demand by 0.685. These two values are roughly the same for the medium-demand group. However, in the low demand, the average of mean demand is only 54% of the value of the average RMSE. This finding reveals that the model's accuracy decreases according to the demand at each station. Thence, the model predicts more accurately for high-demand stations. We did the same calculations with all the remaining models and received the same result.

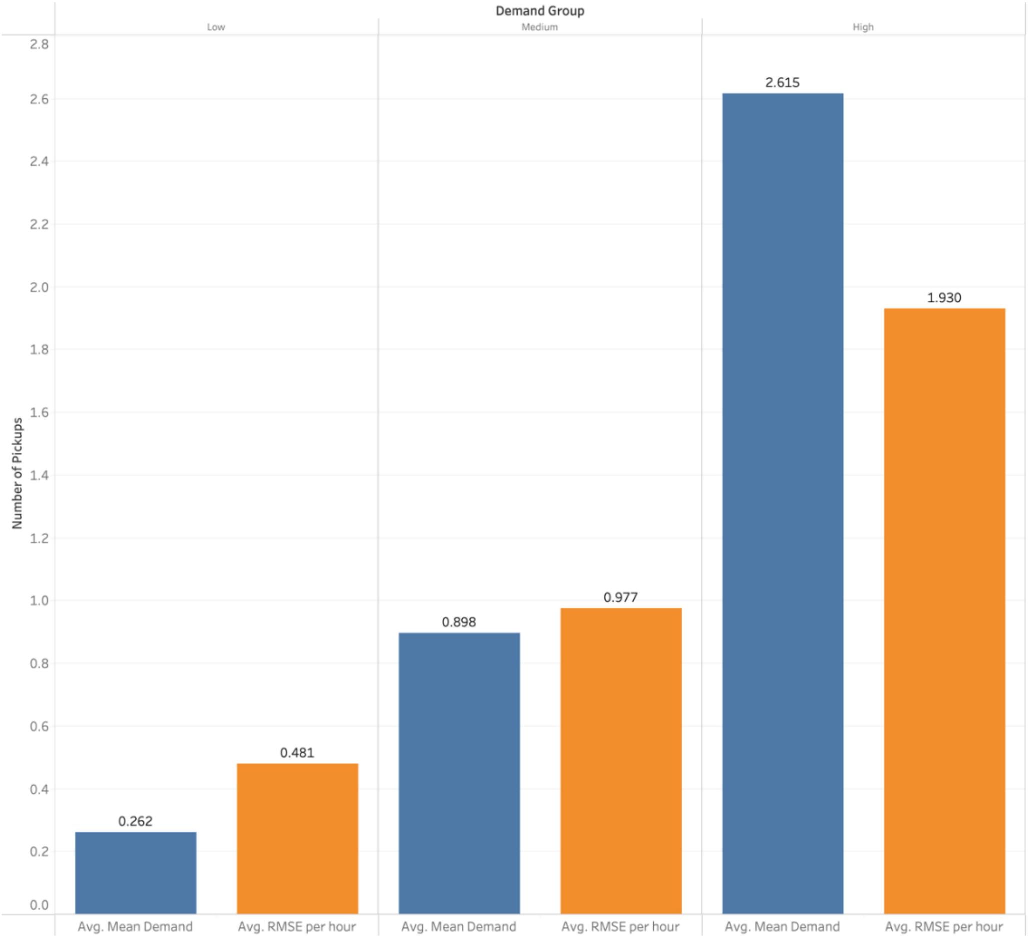


Figure 13. Comparison between Average of mean demand with RMSE per hour of CNN-LSTM 336 by demand groups

For deeper investigation, we run the logarithmic trendline model between r-squared and mean demand of stations. Figure 14 represents the correlation of the primary metric for model accuracy and the mean demand of each station. Appendix 6 reports the numerical result for the logarithmic equation for tested models. All the models get a significant logarithmic correlation with p-values less than 0.0001. The results of the experiment found clearly support for the previous discussion that the model predicts more accurately for high-density stations. It leads to good results in meeting customer expectations in crowded areas. This prediction ability is crucial in increasing satisfaction and maintaining the sustainability of bike-sharing system's growth.

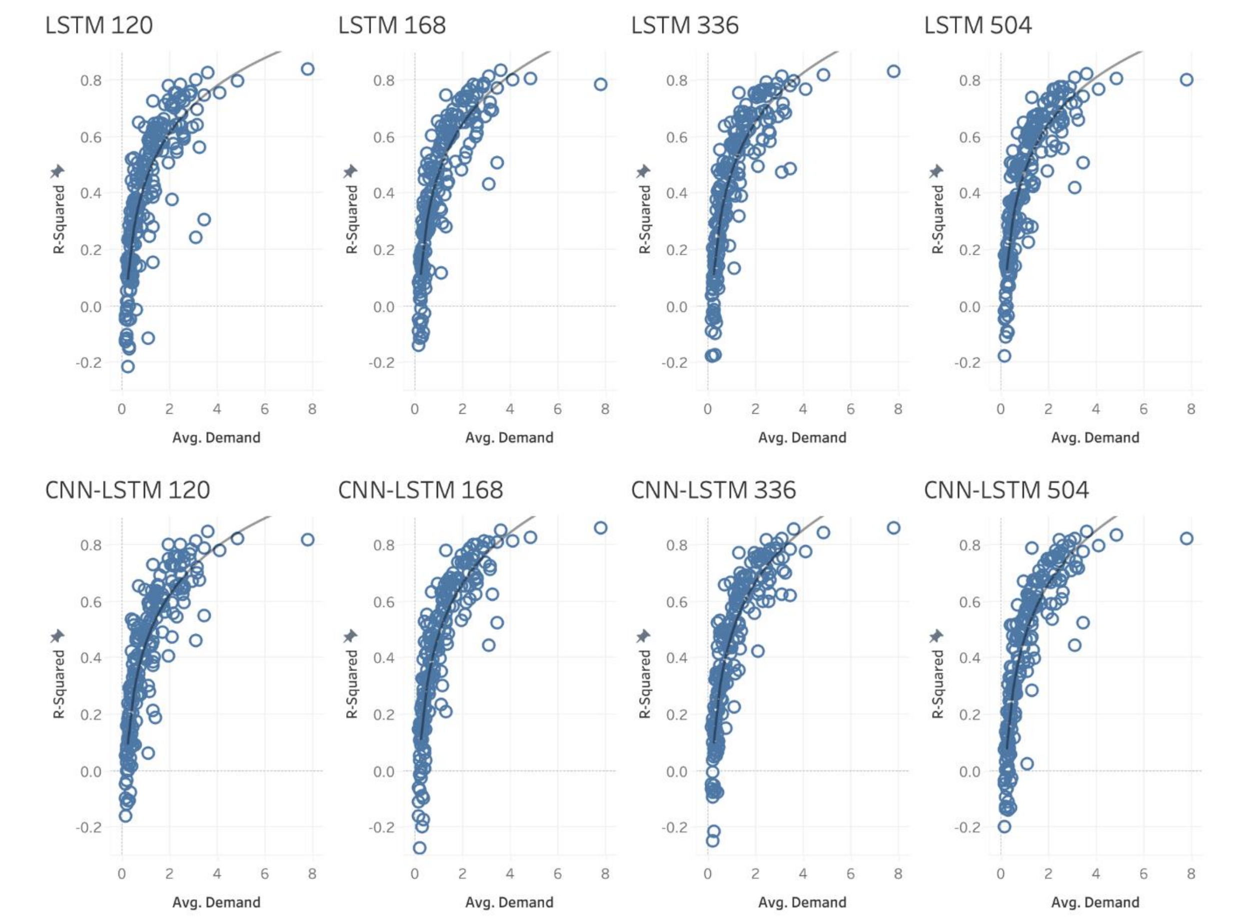


Figure 14. Logarithmic relation of R-squared and Mean demand at each station

5.5. Weather Forecasting in Prediction Accuracy

In the previous section, we concluded that the period interval considering for pickup demand is important, and 336 hours is the optimal lags for prediction models. In that experimental design, we used 1-hour historical weather data, the current weather and forecasting weather in the next 3 hours. This setup is because we believe customers need time to prepare to go out by bike, and the delay is one hour. Besides, users also check the upcoming short-term weather forecast to decide if it is suitable for cycling. However, another question arises about how many hours the user will look at the weather forecast to make a decision. To answer this question, we redesigned our experiment with historical demand in 336 hours and changed the amount of weather information in the coming hours, starting at 2 hours, 3 hours, 4 hours, and 5 hours respectively. Table 2 shows the results of our new experiments.

As initially predicted, the increase or decrease in the amount of weather information affects the accuracy of the model's prediction results. We can observe the pattern more accurately with increasing average r-squared, and MAE, RMSE decreasing as the amount of weather information is increased from 2 hours to 4 hours. Then the accuracy decreases as we increase to 5 hours for both LSTM and CNN-LSTM models. This finding can be explained as users only look at weather forecasts for their trips for the next 4 hours. It also corresponds to the analysis of the user's trip duration.

	Metric		
Model	R-Squared	MAE	RMSE
CNN-LSTM 336_2	0.417	1.334	2.205
CNN-LSTM 336_3	0.429	1.322	2.182
CNN-LSTM 336_4	0.435	1.320	2.182
CNN-LSTM 336_5	0.432	1.321	2.187
LSTM 336_2	0.434	1.347	2.218
LSTM 336_3	0.425	1.348	2.230
LSTM 336_4	0.428	1.350	2.229
LSTM 336_5	0.420	1.355	2.239

Table 2. Prediction model testing results with different weather forecasting interval

5.6. *Discussion for Future Research*

Despite the limitations of our computing resources, we optimized our code and model to pre-process the huge dataset (12 million rows) and train the prediction models. Our research experiment results provide solid evidence to select the proper timeframe for the deep learning model. We approach the topic using traditional statistical analysis techniques to analyze Spatiotemporal data properties to show the traditional method's limitations. Furthermore, we implement modern deep learning models to evaluate relationships of natural factors affecting the model's accuracy. Our model is able to reach 90% accuracy for some stations and achieves an average error for the whole bike-sharing system of 2 pickups for 2 hours at each station or one pickup per hour. This result is promising. Our research has shown that the CNN-LSTM model with a lag interval of 336 hours and 4 hours of weather forecast conditions gives the highest accuracy at RMSE of 2.182 for 2 hours and average R-Squared for 200 stations at 0.435. This provides a good starting point for discussion to develop a good prediction model.

From the results of this study, future research should consider the return bikes at each station to predict the net demand at each station. In addition, the model can be developed into professional analytical applications to support existing prediction systems.

References

Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M. and Farhan, L., 2021. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, *8*(1), pp.1-74.

Atluri, G., Karpatne, A. and Kumar, V., 2018. Spatio-temporal data mining: A survey of problems and methods. *ACM Computing Surveys (CSUR)*, *51*(4), pp.1-41.

Bahadori, M.T., Yu, Q.R. and Liu, Y., 2014. Fast multivariate spatio-temporal analysis via low rank tensor learning. *Advances in neural information processing systems*, *27*.

Caulfield, B., O'Mahony, M., Brazil, W. and Weldon, P., 2017. Examining usage patterns of a bike-sharing scheme in a medium sized city. *Transportation research part A: policy and practice*, *100*, pp.152-161.

Castro, P.S., Zhang, D., Chen, C., Li, S. and Pan, G., 2013. From taxi GPS traces to social and community dynamics: A survey. *ACM Computing Surveys (CSUR)*, *46*(2), pp.1-34.

Elsworth, S. and Güttel, S., 2020. Time series forecasting using LSTM networks: A symbolic approach. *arXiv preprint arXiv:2003.05672*.

Graves, A., 2012. Supervised sequence labelling. In *Supervised sequence labelling with recurrent neural networks* (pp. 5-13). springer, berlin, Heidelberg.

Graves, A. and Schmidhuber, J., 2008. Offline handwriting recognition with multidimensional recurrent neural networks. *Advances in neural information processing systems*, *21*.

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural computation*, *9*(8), pp.1735-1780.

Jiang, Z., Sainju, A.M., Li, Y., Shekhar, S. and Knight, J., 2019. Spatial ensemble learning for heterogeneous geographic data with class ambiguity. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *10*(4), pp.1-25.

Kong, X., Li, M., Ma, K., Tian, K., Wang, M., Ning, Z. and Xia, F., 2018. Big trajectory data: A survey of applications and services. *IEEE Access*, *6*, pp.58295-58306.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems, 25*.

LeCun, Y. and Bengio, Y., 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, *3361*(10), p.1995.

Luo, X., Li, D., Yang, Y. and Zhang, S., 2019. Spatiotemporal traffic flow prediction with KNN and LSTM. *Journal of Advanced Transportation*, *2019*.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J. and Khudanpur, S., 2010, September. Recurrent neural network based language model. In *Interspeech* (Vol. 2, No. 3, pp. 1045-1048).

Mozer, M.C., 1991. Induction of multiscale temporal structure. *Advances in neural information processing systems, 4*.

Ojo, S.O., Owolawi, P.A., Mphahlele, M. and Adisa, J.A., 2019, November. Stock market behaviour prediction using stacked LSTM networks. *In 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC) (pp. 1-5)*. IEEE.

Siami-Namini, S., Tavakoli, N. and Namin, A.S., 2018, December. A comparison of ARIMA and LSTM in forecasting time series. *In 2018 17th IEEE international conference on machine learning and applications (ICMLA) (pp. 1394-1401)*. IEEE.

Staudemeyer, R.C. and Morris, E.R., 2019. Understanding LSTM--a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.

Wang, S., Cao, J. and Yu, P., 2020. Deep learning for spatio-temporal data mining: A survey. *IEEE transactions on knowledge and data engineering*.

Zhou, H., Li, L. and Zhu, H., 2013. Tensor regression with applications in neuroimaging data analysis. Journal of the American Statistical Association, 108(502), pp.540-552.

Jessica Quach, R. M. (2008). Exploring the weather impact on bike sharing usage through a clustering analysis.

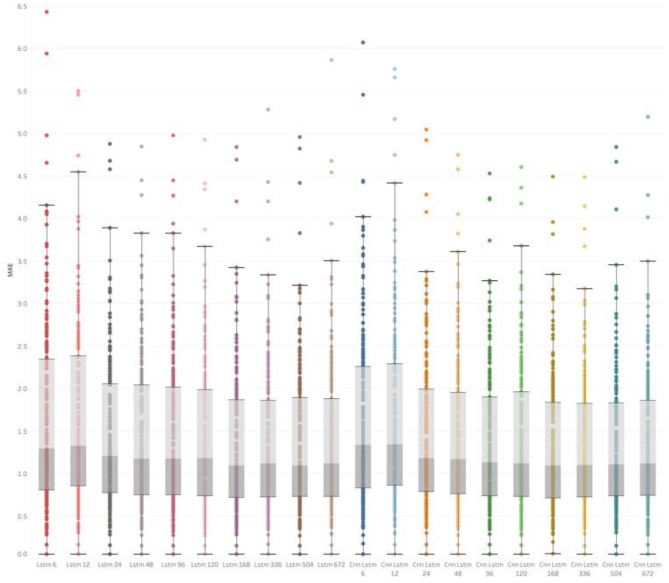
visual crossing weather. (2020, March 23).

Willberg, E. (2019, February). BIKE SHARING AS PART OF URBAN MOBILITY IN HELSINKI– A USER PERSPECTIVE.

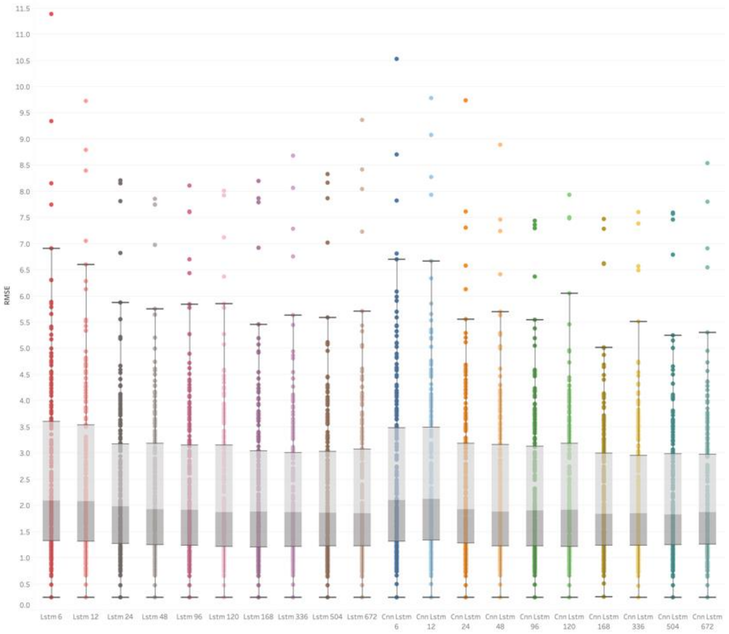
Xinwei Ma, Y. Y. (2022, January 23). Short-Term Prediction of Bike-Sharing Demand Using Multi-Source Data: A Spatial-Temporal Graph Attentional LSTM Approach.

Lei Ji, Y. Z. (2019). Carbon futures price forecasting based with ARIMA-CNNLSTM model. *7th International Conference on Information Technology and Quantitative Management*, 37.

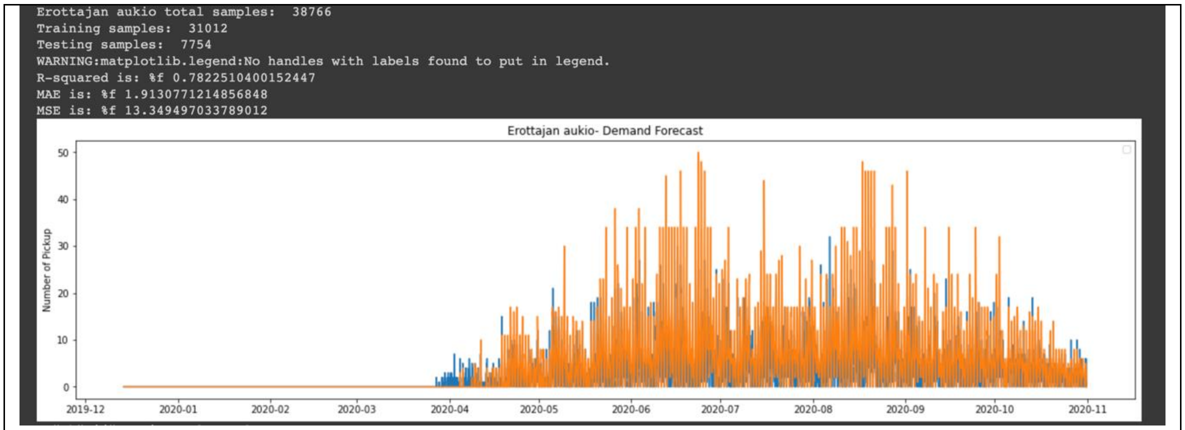
Appendix



Appendix 1. Boxplot distribution of MAE results for 200 stations

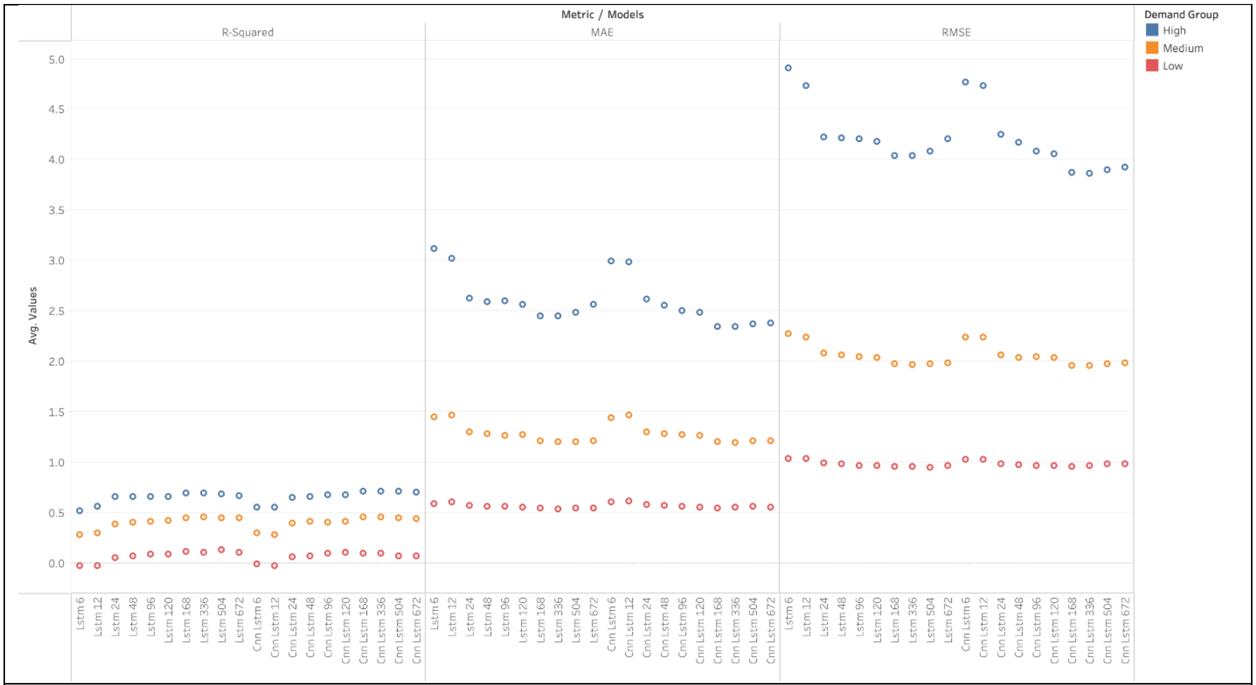


Appendix 2. Boxplot distribution of RMSE results for 200 stations



```
Erottajan aukio total samples:  38766
Training samples:  31012
Testing samples:  7754
WARNING:matplotlib.legend:No handles with labels found to put in legend.
R-squared is: %f 0.7822510400152447
MAE is: %f 1.9130771214856848
MSE is: %f 13.349497033789012
```

Appendix 3. Sample prediction result at Erottajan Aukio station



Appendix 4. Average R-Squared, MAE, RMSE of testing model for high, medium, and low demand stations

		Metric		
Models	Demand Group	R-Squared	MAE	RMSE
LSTM 6	High	0.5185	3.1192	4.9071
	Medium	0.2745	1.4488	2.2752
	Low	-0.0277	0.5902	1.0325
LSTM 12	High	0.5591	3.0234	4.7326

	Medium	0.2956	1.4626	2.2394
	Low	-0.0245	0.6041	1.0308
LSTM 24	High	0.6535	2.6251	4.2227
	Medium	0.3863	1.3000	2.0821
	Low	0.0484	0.5678	0.9899
LSTM 48	High	0.6545	2.5919	4.2127
	Medium	0.4025	1.2842	2.0585
	Low	0.0649	0.5564	0.9805
LSTM 96	High	0.6564	2.5973	4.2074
	Medium	0.4093	1.2660	2.0431
	Low	0.0891	0.5562	0.9659
LSTM 120	High	0.6581	2.5637	4.1787
	Medium	0.4170	1.2686	2.0337
	Low	0.0889	0.5479	0.9661
LSTM 168	High	0.6878	2.4488	4.0366
	Medium	0.4440	1.2090	1.9781
	Low	0.1091	0.5392	0.9547
LSTM 336	High	0.6879	2.4508	4.0386
	Medium	0.4537	1.2023	1.9613
	Low	0.1029	0.5377	0.9576
LSTM 504	High	0.6829	2.4875	4.0833
	Medium	0.4497	1.2038	1.9704
	Low	0.1267	0.5431	0.9463
LSTM 672	High	0.6659	2.5585	4.2021
	Medium	0.4480	1.2062	1.9800
	Low	0.1025	0.5464	0.9612
CNN-LSTM 6	High	0.5518	2.9952	4.7681
	Medium	0.2939	1.4365	2.2396
	Low	-0.0152	0.6002	1.0236
CNN-LSTM 12	High	0.5539	2.9883	4.7306
	Medium	0.2810	1.4649	2.2389
	Low	-0.0280	0.6108	1.0290
CNN-LSTM 24	High	0.6500	2.6123	4.2519
	Medium	0.3924	1.3017	2.0647
	Low	0.0547	0.5770	0.9848
CNN-LSTM 48	High	0.6593	2.5564	4.1738
	Medium	0.4139	1.2759	2.0372
	Low	0.0667	0.5673	0.9769
CNN-LSTM 96	High	0.6714	2.4991	4.0851
	Medium	0.4024	1.2671	2.0414
	Low	0.0900	0.5611	0.9647

CNN-LSTM 120	High	0.6777	2.4877	4.0540
	Medium	0.4128	1.2585	2.0398
	Low	0.1023	0.5536	0.9598
CNN-LSTM 168	High	0.7095	2.3462	3.8687
	Medium	0.4543	1.2012	1.9582
	Low	0.0983	0.5424	0.9592
CNN-LSTM 336	High	0.7119	2.3462	3.8600
	Medium	0.4556	1.1965	1.9532
	Low	0.0939	0.5487	0.9622
CNN-LSTM 504	High	0.7090	2.3716	3.8941
	Medium	0.4448	1.2054	1.9712
	Low	0.0670	0.5569	0.9804
CNN-LSTM 672	High	0.7046	2.3773	3.9285
	Medium	0.4364	1.2125	1.9851
	Low	0.0648	0.5553	0.9829

Appendix 5. Sample prediction result at Erottajan Aukio station

Model	Coefficients		R-squared	p-value
	Ln(avg_of_station_demand)	intercept		
LSTM 120	0.239921	0.447037	0.741643	<0.0001
LSTM 168	0.246484	0.47442	0.790976	< 0.0001
LSTM 336	0.249682	0.47843	0.812139	< 0.0001
LSTM 504	0.238293	0.478658	0.807997	< 0.0001
CNN-LSTM 120	0.24484	0.454246	0.766907	< 0.0001
CNN-LSTM 168	0.258422	0.484889	0.805415	< 0.0001
CNN-LSTM 336	0.264113	0.486253	0.825437	< 0.0001
CNN-LSTM 504	0.270616	0.474817	0.821038	< 0.0001

Appendix 6. Logarithmic equation of average of mean demand and R-Squared of prediction models