FORECASTING SHORT-TERM BIKE SHARING DEMAND AT STATION-LEVEL USING DEEP NEURAL NETWORKS

ROB VAN DER WIELEN

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2084445

COMMITTEE

dr. Giacomo Spigler
dr. Nevena Ranković

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

December 4th, 2023

WORD COUNT

8261

ACKNOWLEDGMENTS

First, I would like to thank my supervisor, Dr. Giacomo Spigler, for his consistent support and expert advice throughout the process of writing this thesis. I'm also grateful to my friends and family for their unconditional support and encouragement. Your confidence has been a constant source of strength for me, and I cherish the motivation that helped me finalize this last phase of my study.

FORECASTING SHORT-TERM BIKE SHARING DEMAND AT STATION-LEVEL USING DEEP NEURAL NETWORKS

ROB VAN DER WIELEN

Abstract

As urban transportation evolves, bike sharing emerges as a sustainable and convenient option, integral to modern city landscapes. Accurately predicting bike sharing demand is critical for operational efficiency and user satisfaction, particularly at the station level where demand fluctuates dynamically. This thesis addresses the challenge of employing LSTM and CNN-LSTM neural networks to predict station-level bike-sharing demand in New York City, a crucial yet underexplored area in urban transportation planning. The study analyzes data from 263 bike stations, enriched with key spatial, temporal, and weather features, including examples like bike lane lengths, holidays, precipitation, and temperature. It demonstrates that LSTM and CNN-LSTM models significantly outperform the baseline ARIMA model, with the CNN-LSTM model achieving an RMSE of 0.8000 and an MAE of 0.5290. Embeddings for station IDs are used to effectively map these IDs into a meaningful representation, crucial for accurate demand modeling at each station. Importantly, including the station capacity in the models markedly improves predictive performance. Furthermore, the study introduces a novel approach by employing an ensemble CNN-LSTM model, consisting of a generic sub-model trained on common features across all stations and a specific sub-model focused on station-specific features and trained on the residual of the first sub-model. While this ensemble model performs well, it shows equivalent predictive performance to the singular CNN-LSTM model that integrates all variables. Future research should integrate advanced techniques for capturing hidden station correlations and develop strategies to buffer against the variances of high and low demand periods.

1 DATA SOURCE, ETHICS, CODE, AND TECHNOLOGY STATEMENT

The research utilized data from various sources: the primary dataset from the Citi Bike website ("Citi Bike System Data", n.d.), supplemented by New York City's infrastructural data from NYC OpenData ("NYC OpenData", 2023), weather data from OpenMeteo ("Open-Meteo", 2023), and geographical mapping from OpenStreetMap ("OpenStreetMap", 2023). Work on this thesis did not involve collecting data from human participants or animals. The original owner of the data in this thesis retains ownership of the data during and after the completion of this thesis. All the figures presented in this work belong to the author. To improve the author's original content, a generative language model, ChatGPT by OpenAI ("ChatGTP", 2023), was employed. No additional tools or services were utilized. A comprehensive list of all software deployed in this study is presented in Table 8 of Appendix A. The codes implemented throughout the thesis are accessible via this GitHub repository.

2 INTRODUCTION

For long, bikes have played an important part in city transportation (Pan et al., 2019). In the last decade, there has been growing attention on Bike Sharing Systems (BSS) (Faghih-Imani et al., 2014). To decrease emissions, reduce energy consumption, and improve physical activity, BSS are widely adopted in many major cities all around the world (Lin et al., 2018; Shaheen et al., 2010; Wang & Kim, 2018). However, effectively meeting user demand in these systems presents significant operational challenges. For instance, during morning rush hours, there can be a notable imbalance, such as a shortage of available bikes in residential areas, due to the increased volume of commuting trips (Yang et al., 2020). These challenges highlight the importance of accurate demand prediction in BSS.

In the realm of BSS demand prediction, researchers have categorized their models into three primary levels: city-level, cluster-level, and station-level (Lin et al., 2018). Station-level demand (i.e., predicting the demand for each station), characterized by its high variability, presents formidable challenges for precise predictions. This complexity might explain the prevalent trend in the academic literature towards broader, macroscopic predictions, focusing primarily on city and cluster levels (Wang & Kim, 2018). Cluster-level demand predictions involve partitioning cities into grids, and these predictions have demonstrated a high degree of accuracy (Mehdizadeh Dastjerdi & Morency, 2022; Wang & Kim, 2018). However, this kind of demand prediction fails to address the differences between individual stations (Yu et al., 2023). Given these challenges, there's a

pressing need to delve deeper into station-level forecasting (Lin et al., 2018; Ren et al., 2019).

Academic literature has found that deep learning models consistently outperform basic models like ARIMA (Mehdizadeh Dastjerdi & Morency, 2022), Random Trees (Wang & Kim, 2018), and Support Vector Machines (SVM) (Ai et al., 2019) in predicting the BSS demand. This is because deep learning models are better at capturing the spatial-temporal relationship that is inherent to this kind of data (Liang et al., 2022; Lin et al., 2018; Mehdizadeh Dastjerdi & Morency, 2022). Long Short-Term Memory (LSTM) networks, a type of neural network, generally excel in recognizing time series data (Sherstinsky, 2020; Zhao et al., 2017). However, they often fall short in capturing spatiotemporal characteristics, which are important in BSS predictions (Ai et al., 2019). To address this limitation Ke et al. (2017), to forecast short-term passenger demand, proposed an end-to-end structure that combines Convolutional Neural Networks (CNN) and LSTM networks to concurrently capture spatial and temporal dependencies.

2.1 *Project definition*

This study leverages Convolutional Long Short Term Memory (CNN-LSTM) networks to analyze temporal and spatial demand patterns, focusing on short-term, station-level bike demand.

In the realm of BSS demand prediction, various machine learning algorithms and neural networks, including LSTMs, RNNs, and CNNs, have been utilized (Liang et al., 2022; Wang & Kim, 2018). Notably, both Ai et al. (2019) and Mehdizadeh Dastjerdi and Morency (2022) have implemented CNN-LSTMs for cluster-level demand forecasting. These studies primarily explored supply and demand dynamics, overlooking the impact of station capacity, a factor found to influence forecasting performance by El-Assi et al. (2017). Distinguishing this research, we focus on station-level demand predictions and integrate station capacity into our CNN-LSTM model, a novel approach to this kind of demand prediction. Furthermore, this study will also include a model that operates in two phases: initially, it predicts demand using variables common to all stations, such as weather and time. Subsequently, a second model, focusing on station-specific variables like coordinates and station IDs, and trained only on the residual from the first sub-model, refines this prediction. This bifurcated approach effectively merges universal and localized factors, with the first model capturing overarching trends and the second adjusting for specific station conditions. This methodology echoes the approach in Gao et al. (2021), where the model was segmented into three sub-models for energy consumption prediction, namely time, spatial, and historical models.

2.2 *Social and Scientific Relevance*

Societal relevance: Accurate station-level predictions of bike demands throughout the day are pivotal for optimal bike scheme management and fleet re-balancing among these stations (Yang et al., 2020). It can alleviate urban traffic congestion, diminish greenhouse gas emissions, reduce transportation expenses, and bolster multi-modal transport connections (Sohrabi et al., 2020). Additionally, it also allows the scheduling of the Vehicle-Grid-Integration (VGI) service (i.e., the policy of charging electric bikes in a manner that benefits the grid while meeting driver's needs) (Pratt, n.d.; Ren et al., 2019).
Scientific relevance: While only a few studies have used CNN-LSTMs in the BSS cluster level-demand predictions, no study has yet explored the potential of CNN-LSTM models for station-level BSS demand predictions, particularly with a focus on the total bike capacity at stations. Highlighting a gap in the literature, Mehdizadeh Dastjerdi and Morency (2022) underscores the limited application of hybrid CNN-LSTM architectures in analyzing station-level BSS demand. Thus, devising a model that zeroes in on station-level BSS demand while also considering station capacity promises to enrich existing scholarly work.

2.3 *Research Questions*

To address the primary research objective of this study, the following main question is posed:

To what extent can Neural Networks predict the bike sharing demand at station level?

To answer this broad main research question three sub-questions were formulated:

- **RQ1:** To what extent can Long Short-Term Memory (LSTM) and Convolutional Long Short-Term Memory (CNN-LSTM) networks predict the station-level BSS demand, when compared to the RMSE of the baseline ARIMA model?
- **RQ2:** To what extent is the RMSE of the CNN-LSTM influenced by the inclusion of the station's capacity?
- **RQ3:** How does the combined predictive capability of two separate models – one using generic features for all stations and another incorporating unique station-specific features – compare to a singular comprehensive model in predicting station-level BSS demand?

3 RELATED WORK

3.1 *Predicting bike demand*

Cities typically offer two BSS service models: docked and dockless BSS. The proliferation of dockless BSS has introduced several challenges, including obstructing pedestrian pathways and the issue of bike abandonment (Wang & Kim, 2018). Consequently, the docked BSS program (i.e., a bike-sharing program that enables users to pick up and return bicycles to different stations) has gained prominence in recent years as a more favored alternative (Shaheen et al., 2010). As mentioned before, a considerable amount of attention has already been devoted to the bike-sharing demand prediction problem, because these systems allow for real-time data collection using sensors, which generate large quantities of, often, public data (Albuquerque et al., 2021). However, most studies have focused on city and cluster-level demand predictions. The station-level demand predictions have gained less attention from academics. These types of bike demand predictions are challenging due to dynamic patterns across stations. This complexity, called data noise, can make station-level demand predictions less accurate and makes cluster analysis a preferred approach over individual station predictions (Mehdizadeh Dastjerdi & Morency, 2022). While cluster-level analysis can capture broader trends, station-level predictions are important for understanding specific station demands. Setting a grid size too large in cluster analysis can overlook individual station nuances, and if it's too small, it can lead to computational challenges (Lin et al., 2018). Furthermore, Yu et al. (2023) highlighted that even though clustering possibly captures local demand better than city-level predictions, it fails to address the differences between individual bike-sharing stations. Hence, station-level demand predictions and modeling are a better solution for bike fleet management because they meet the user's demand at each station (Mehdizadeh Dastjerdi & Morency, 2022).

3.2 *Models for BBS predictions*

Empirical studies have explored various predictive models for BSS demand predictions. Caulfield et al. (2017) used a logistic regression model to study the dynamics of BSS schemes in smaller cities. Ashqar et al. (2017) Employed uni-variate models of Random Forest, LSBoost, and multivariate model of Partial Least-Squares Regression to predict bike demand at BSS stations. While uni-variate models generally outperformed the multivariate approach, the latter was beneficial for networks with many correlated stations. Sathishkumar et al. (2020) combined various Machine

Learning models to forecast hourly bike rentals, with the Gradient Boosting Machine proving the most effective. While traditional models have shown varied efficacy in BSS demand predictions, Deep Learning models are adept at multivariate time series forecasting, capturing both long and short-term patterns (Lai et al., 2018). Moreover, these models offer an end-to-end integrated learning architecture that seamlessly captures spatial and temporal dependencies, making them especially effective for tasks like forecasting short-term BSS demand (Ke et al., 2017). Pan et al. (2019) utilized a basic LSTM network, noting its superior prediction performance for different timestamps compared to several machine learning models. Wang and Kim (2018) applied RNNs for short-term bike demand predictions across different time intervals. Notably, studies by Ai et al. (2019) and Mehdizadeh Dastjerdi and Morency (2022) used the Conv-LSTM (or hybrid CNN-LSTM). This model, especially when enriched with additional input features, demonstrated superior capability in capturing spatiotemporal correlations and provided deeper insights into bike demand patterns. Among the evaluated models, the Conv-LSTM stood out as the most effective for BSS demand predictions.

3.3 *Input data*

A wide variety of explanatory variables were considered in past research. First, it is evident that the inclusion of weather variables consistently enhances the forecasting performance (Campbell et al., 2016; Caulfield et al., 2017; Chen et al., 2016; Kim, 2018). Sathishkumar et al. (2020) in their study ranked selected features on their importance (i.e., in terms of their reduction of the RMSE). The weather features humidity, temperature, sun hours, precipitation and visibility were all in the top ten (Sathishkumar et al., 2020). Temporal factors, such as time of the day, day of the week, and holidays are consistently utilized in BSS demand predictions, with the hour of the day identified as the most crucial feature (Kim, 2018; Mehdizadeh Dastjerdi & Morency, 2022; Pan et al., 2019). Additionally, spatial variables like proximity to parks, bike lanes, railway stations, recreational facilities, and businesses have been linked to bike usage (Etienne & Latifa, 2014; Faghih-Imani et al., 2014; Liang et al., 2022). Moreover, it is important to acknowledge that station capacity varies, and this can influence the model's forecasting accuracy. The level of station usage may be influenced by the number of available docking stations, and this capacity can change over time, potentially affecting the model's forecasting accuracy. While comparable studies of Mehdizadeh Dastjerdi and Morency (2022) and Ai et al. (2019) did not account for station capacity in their cluster-level

demand predictions, El-Assi et al. (2017) found that it is a crucial factor to incorporate in these types of prediction models.

3.4 *Prediction window*

Then, besides the selection of relevant variables, the choice of prediction window in time series forecasting is also a critical decision that can significantly impact the accuracy of predictions. The literature suggests that the size of the prediction window plays a pivotal role in determining the precision of real-time forecasts. For instance, Mehdizadeh Dastjerdi and Morency (2022) emphasized the importance of using the last 24 hours (i.e., 96-time steps) of historical data as input for forecasting bike pickups for the subsequent 15 minutes, which was also found by Ashqar et al. (2017) to be the optimal widow. Similarly, C. Xu et al. (2018) experimented with windows of 10, 15, 20, and 30 minutes, concluding that longer intervals resulted in more accurate results, likely due to reduced data noise in extended intervals.

3.5 *Residual learning*

The division of models, where the second model exclusively learns the residual, exemplifies residual learning in neural networks. This technique markedly enhances model training and addresses the vanishing and exploding gradient issues (Szegedy et al., 2017; Zhang et al., 2019). In the realm of traffic predictions, Gao et al. (2021) effectively employed a ResNet structure to boost efficiency in traffic load forecasting. Their method, which used spatial, historical, and temporal data, highlights the versatility and efficacy of residual. Hence, this could be beneficial in the case of bike demand predictions.

This study demonstrates the relevance of setting up a robust deep-learning model to capture the demand patterns for different stations. Several models will be evaluated and compared to a baseline model.

4 METHOD

The methodology comprises several stages that are defined on a high level in Figure 1. Initially, the study area and accompanying dataset are described. This is followed by an exploration of the data and subsequent data-prepossessing steps. Then, the various algorithms employed in this study and the experiments conducted are detailed. Finally, the process of hyperparameter tuning is outlined. Due to time constraints in this study, it was not feasible to conduct hyperparameter tuning on the entire dataset. Consequently, the dataset was divided into two subsets: one for hyperparameter tuning and another for training and evaluating the model. This approach is illustrated in Figure 1.

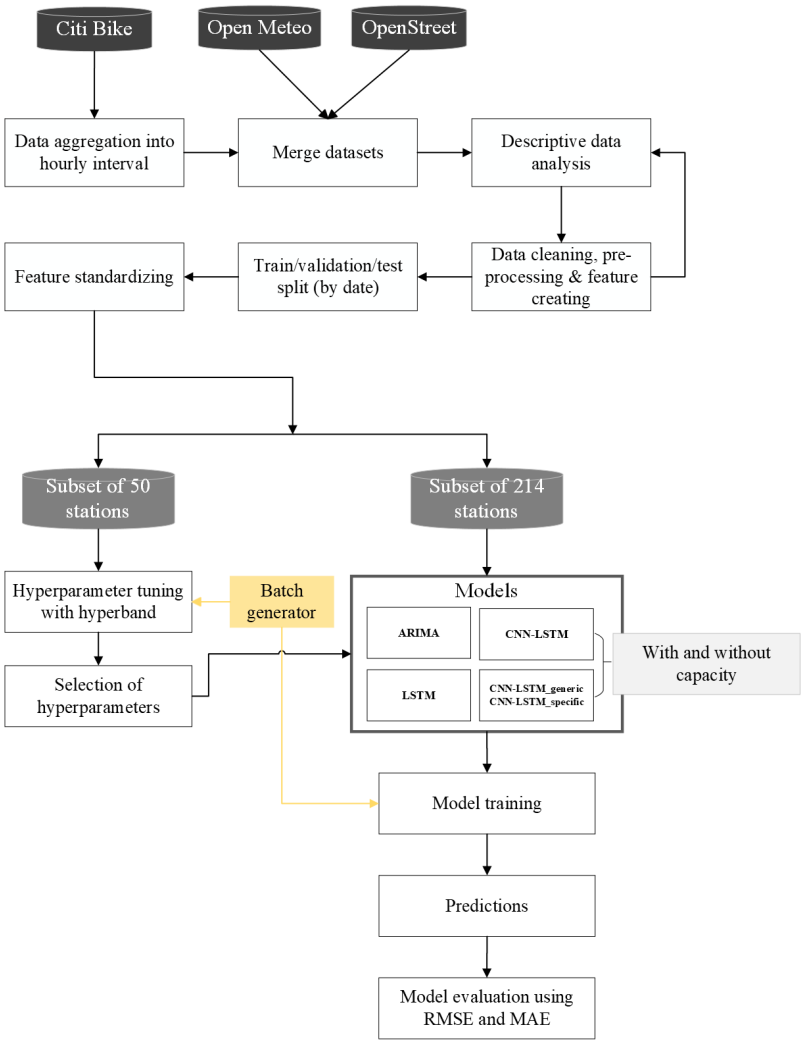


Figure 1: High-level methodology framework

4.1 *Study area and dataset description*

This research focuses on New York City, predominantly within the Manhattan neighborhood. It is notable to note that the methodology and scope of this study, particularly the approach applied to BSS demand predictions, are applicable in other urban contexts with similar data availability. Three primary data sources were utilized in his study.

First, the New York's Citi Bike program's dataset serves as the foundation for analyzing bike-sharing demand, with the records for this study spanning from January 2019 to October 2023 ("Citi Bike System Data", n.d.). Each transaction includes trip duration, bike check-out and check-in times, station names, geographical coordinates, and more. The original dataset did not include information on station capacity; this detail was accessible solely through the Citi Bike API, which provides real-time data. Consequently, we must assume that station capacities remained constant over the study period. The data was formatted by counting the number of pick-ups for each station for every hour, yielding a raw dataset of approximately 28 million rows. Figure 2 shows a map with the stations that are included in this study. Further data preprocessing is explained in later sections. Second, weather variables were sourced from the Open-Meteo API ("Open-Meteo", 2023), offering historical data since 1940, which is particularly valuable because it is available on an hourly basis. The dataset includes temperature, humidity, precipitation, snowfall, weather conditions, and cloud cover. Lastly, OpenStreetMap ("OpenStreetMap", 2023) provided spatial features essential for the study, such as locations of transportation hubs, commercial buildings, retail outlets, restaurants, and educational institutions, each with associated geographical coordinates. The name of the facility accompanied by the longitude and latitude were retrieved in a CSV file.

Figure 2: Selected bike stations in New York

4.2 *Descriptive data analysis*

The visual representation of Citi Bike demand in Figure 3 indicates a significant variation in usage patterns over time. The red line marks the onset of the COVID-19 pandemic in March 2020, which led to a drastic drop in demand due to lockdowns and social distancing measures. This pandemic-induced impact is a pivotal consideration in this analysis. To mitigate the anomaly presented by the pandemic, this study proceeds with data from 2021 onward, thus focusing on the new normal in urban mobility.

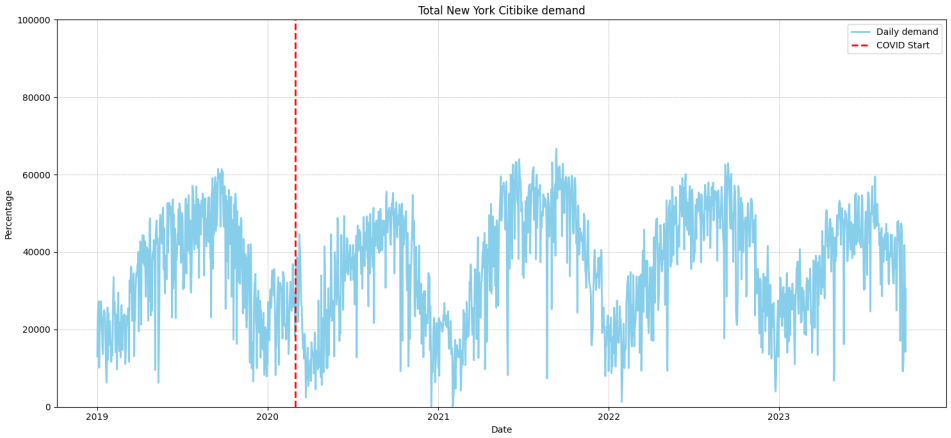


Figure 3: The demand aggregated by day, including the start of the Corona pandemic (March 1st, 2020)

Analysis of a bar chart that includes the demand by day of the week shows a distinct pattern: weekdays, especially from Tuesday through Friday, see the highest bike usage, suggesting a strong link to work-related commuting. Weekends, in contrast, exhibit a demand reduction, hinting at different usage motives (Figure 15, Appendix B). Furthermore, yearly trends broken down by hour and month demonstrate peak demand during morning and evening rush hours, with minimal activity at midnight. Seasonality affects usage, as warmer months, particularly August and September, outpace the colder months of January and February in terms of rides (Figure 14 & Figure 16, Appendix B). These temporal features show that daily and hourly factors, alongside seasonal trends, are significant predictors of demand in bike-sharing systems.

Then, to underscore the significance of station-level BSS demand predictions versus cluster-level, data were segmented into 10 clusters via K-means clustering. This popular method for grouping stations is well-established in cluster-level BSS demand prediction literature (Caggiani et al., 2018; H.

Xu et al., 2018; Yu et al., 2023). A visualization of the cluster distribution is provided in Figure 17 (Appendix B). Figure 4 focuses on four randomly selected stations within a single cluster, plotting hourly demand against days of the week. The plots reveal both commonalities and stark differences in usage patterns. For instance, Station Central Park North & Adam Clayton Powell (a) shows peak demand over weekends, whereas S Ave & E 103 St. (a) experiences higher weekday demand. These observations align with what was given by Yu et al. (2023), confirming that cluster-level predictions do not fully capture the nuanced variability among individual stations and show the need for station-level demand predictions.

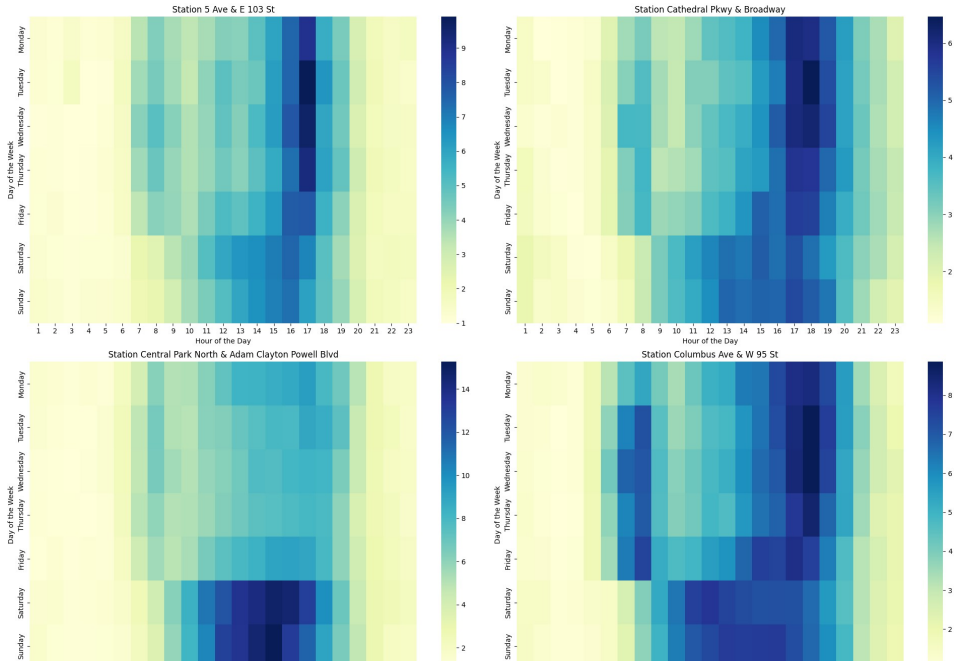


Figure 4: Heat map showing the demand by hour of the day and day of the week for 4 randomly selected stations in the same cluster using K-means clustering

4.3 *Data pre-processing*

4.3.1 *Data cleaning*

Subsequent data preprocessing involved several steps. Initially, only stations operational throughout the entire five-year period were retained. Next, some stations in the dataset were rarely used over the 5-year span. Hence, stations with an average demand below one bike per hour (totaling fewer than 41,637 bikes) were excluded. Additionally, any stations located

outside of New York City's boundaries were removed. Also, a few stations did not provide the total capacity. These stations were also omitted from the dataset, resulting in 263 stations under consideration for this study, as depicted in Figure 2. Lastly, in alignment with the data description section, records before January 2021 were omitted to mitigate the distortive effects of the pandemic, resulting in a refined dataset of 4,682,474 entries, each denoting the number of hourly pickups (i.e., demand) at a specific station.

4.3.2 *Feature engineering*

Temporal features such as year, month, day, and hour were derived from the data's timestamps. Furthermore, from the analysis in Figures 14 & 15 (Appendix B), peak demands were observed at different times of the day and week. To account for this, a weekend variable was included. Additionally, distinct time slots are represented for the morning, morning rush hour, lunchtime, afternoon, afternoon rush hour, and evening, following the approach of Mehdizadeh Dastjerdi and Morency (2022). The variables 'Month', 'Day of the week', 'Year', and 'Time slot' are inherently categorical and lack any ordinal relationship that might influence the demand prediction. Consequently, these variables were subjected to one-hot encoding. This encoding process transforms each category into a new binary feature, enabling the model to treat each category as a separate entity without assuming any order. This approach prevents the model from inferring a false ordinal relationship among the variables, which could potentially skew the demand predictions.

Weather conditions on an hourly basis and a public holiday variable, known from other studies to significantly impact BSS demand, were integrated. The weather code, encompassing a hundred different levels, was later excluded from the dataset. Its one-hot encoding, typically applied to numerical features, would have led to a substantial increase in dimensionality.

A wide variety of spatial features are utilized in the academic literature for predicting BSS demand (Eren & Uz, 2020; Kabak et al., 2018). This study employed the most influential spatial features, namely land-use and infrastructure features. A 200-meter buffer zone around each station was utilized to process these spatial features, this is consistent with prior research (Du et al., 2019; El-Assi et al., 2017). The calculation for the land-use features involved determining the distance between bike stations and the various amenities, filtering pairs within the 200-meter threshold, and summing these for each station. Bike lane data was obtained from a GeoJSON file containing all New York City bike lanes ("NYC OpenData", 2023). For this feature, stations were converted into a GeoDataFrame, with a 200-meter buffer applied to each, allowing for the calculation of the total bike lane

length within each station's radius. An overview of all the aggregated variables is given in Table 1.

Table 1: Aggregated Variables for BSS Demand Prediction

Variable Category	**Description**
Weather Features	Relative humidity, Total precipitation (mm), Cloud cover, Wind speed, Sunshine duration
Temporal Features	Hour of the day, Day of the week, Month of the year, Weekend, Time slot, Year, Holiday presence
Business Area	Number of businesses in a 200m radius
Alternative Transportation	Number of bus, train, and rail stations in a 200m radius
Universities	Number of universities in a 200m radius
Recreation and Parks	Number of recreational and park areas in a 200m radius
Bike Lanes	Total length of bike lanes in km in a 200m radius

4.3.3 *Embeddings*

In this thesis, the focus is on the station-level BSS demand predictions. Therefore, the inclusion of station ID is crucial for accurately modeling demand patterns at each station. However, the station IDs are categorical variables and cannot be used directly in neural network models in their raw form. A common practice for categorical variables would be to one-hot-encode the variable. However, with 263 stations in the dataset, this would lead to a very big increase in dimensionality. This is often called the 'curse of dimensionality', which can lead to data sparsity, multicollinearity, and over-fitting (Daum & Huang, 2003). This is where embeddings become vital. Embedding effectively maps categorical variables, like station IDs, into Euclidean spaces, learned by a neural network during training (Guo & Berkhahn, 2016). Embeddings provide a more compact and meaningful representation by mapping similar stations close to each other in the embedding space. Once these embeddings were trained they were concatenated with other model inputs, to form a comprehensive input array for subsequent layers of the neural network.

4.3.4 *Splitting of the data*

The data was partitioned into a training, validation set, and test set. The training set comprised the first 70% of the data and the validation 10%

of the data. The remaining 20% of the data was reserved as a test set to evaluate the final model's performance on unseen data, similar to (Ai et al., 2019; Mehdizadeh Dastjerdi & Morency, 2022). Table 2 includes an overview of the data split, with the belonging intervals.

An alternative way would be to use k-fold cross-validation. However, in time series data, traditional k-fold cross-validation is not typically used because it involves random splitting, which can disrupt the temporal order of the data. A variant known as time series cross-validation can be applied. In this approach, the data is split into k folds with a rolling or expanding window. In this thesis, the Keras Tuner package was used. Implementing time series cross-validation is more challenging because Keras Tuner doesn't support generators with cross-validation. The tuner then needs to be separately run for each fold and the results must then be aggregated manually.

Table 2: Data Partitioning Strategy

Data Usage	**%**	**Time Interval**
Training	70%	Jan 2021 - Nov 2022
Validation	10%	Dec 2022 - Feb 2023
Testing	20%	Mar 2023 - Oct 2023

4.3.5 *Standardization*

The inclusion of total capacity in the model addresses the variability in station sizes and their demand patterns. Standardizing the numerical variables, including total capacity, normalizes their scales, enhancing model training since deep learning algorithms are sensitive to input magnitudes. Furthermore, it can help with mitigating outlier influences, ensuring that no single feature disproportionately affects the learning process. It is important that the standardizing is applied after the data is split. This prevents data leakage, which occurs when information from the test set is used for standardizing the parameters. The standardizing should be calculated only on the training set. Standardizing was also applied to the target variable, the station demand.

4.4 *Models*

Baseline model - Arima
The baseline for this study is an ARIMA model, a common choice for baseline models in BSS prediction due to its familiarity and ability to handle non-stationary data by detrending, thus stabilizing the dataset over time (Chen et al., 2016; Siami-Namini et al., 2018). While we won't delve into the

specifics of ARIMA's parameters, it's generally observed that for extensive time-series data with spatial-temporal elements, such as in bike-sharing systems, ARIMA tends to be surpassed by more advanced models like LSTM and CNN-LSTM models (Mehdizadeh Dastjerdi & Morency, 2022).

Convolutional Neural Network (CNN)
The main focus is on the implementation of a CNN-LSTM learning framework for the BSS demand predictions. Convolutional Neural Networks (CNNs) are specialized deep neural networks for processing data with a grid-like structure, such as images. A CNN is composed of convolutional, pooling, and fully-connected layers. The convolutional layers apply filters to the input to detect features and are defined by following the equation of (Mehdizadeh Dastjerdi & Morency, 2022):

$$l_t = f\left(\sum(x_t \odot k_t) + b_t\right) \tag{1}$$

Where l_t is the output feature map, f is the activation function, x_t is the input, k_t is the convolutional kernel, $\odot$ denotes the element-wise multiplication, and b_t is the bias. The fully connected layers then interpret these feature maps to perform classification. This architecture allows CNNs to efficiently handle the complexity and variability of the data.

Long-Short-Term-Memory network (LSTM)
Long Short-Term Memory Networks (LSTMs) are a special kind of Recurrent Neural Network (RNN) capable of learning long-term dependencies. They were introduced to overcome RNN's weakness on the long-term memory (Ke et al., 2017). LSTMs are capable of remembering and forgetting redundant information and transferring it forward into the network.

The LSTM architecture comprises several gates that control the flow of information, following the equations of (Ai et al., 2019; Mehdizadeh Dastjerdi & Morency, 2022):

- **Forget Gate**: Determines what information is discarded from the cell state.
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$
- **Input Gate**: Controls the update of the cell state with new information.
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$
- **Update Cell**: Updates the old cell state, C_{t-1}, into the new cell state C_t.
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

- **Output Gate**: Determines the next hidden state.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

- **Next Hidden State**: The hidden state for the current timestep.

$$h_t = o_t * \tanh(C_t) \tag{6}$$

Here, σ denotes the sigmoid function, W and b represent the weight matrices and bias vectors for each gate, h_t is the hidden state at time t, x_t is the input at time t, and C_t is the cell state at time t.

Convolutional Long-Short-Term-Memory Neural Network (CNN-LSTM)
While Long Short-Term Memory (LSTM) networks excel at capturing temporal relationships, they are not ideal for forecasting passenger demand due to their inability to account for the spatial dependencies inherent in such data. To address this, a CNN-LSTM network is proposed, combining Convolutional Neural Network (CNN) and LSTM into a single end-to-end deep learning architecture (Ai et al., 2019). This model employs CNN layers for feature extraction from input data, while the LSTM component supports sequence prediction. This synergy effectively captures both temporal and spatial dependencies (Mehdizadeh Dastjerdi & Morency, 2022), a technique that has been applied in various fields, including traffic speed (Cao et al., 2020) and flow predictions (Narmadha & Vijayakumar, 2021), as well as on-demand ride services (Ay et al., 2022). The core concept of the CNN-LSTM algorithm is to transform all inputs, memory cell values, hidden states, and various gate functions, as delineated in Equations 1-5, into 3D tensors. The conv-LSTM operation is illustrated in Figure 5.

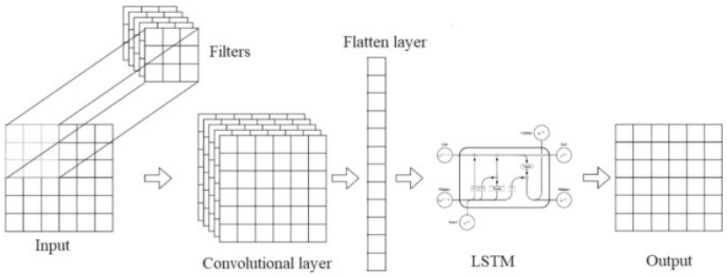


Figure 5: This image is adapted from Ay et al., 2022. It shows the working of the CNN-LSTM algorithm

4.5 *Experimental setup*

4.5.1 *Model configurations*

This thesis evaluated the predictive performance of various model configurations, categorizing the candidate models into three distinct groups:

Group 1: Basic LSTM Model: The first group, or model, is a basic LSTM model with all the temporal, spatial, and weather features included. The number of LSTM layers is given in the hyperparameter tuning. This model also includes a dropout layer, which is important as it helps to prevent overfitting.

Group 2: CNN-LSTM Models without station capacity: The second group consists of two CNN-LSTM models, both without taking into account the station capacity. The first model in this group uses all temporal, weather, and spatial features to predict station-level demand for BSS. The second model divides the feature set: one part uses features that are generic to all stations (i.e., temporal and weather features) to predict the station demand, while the other uses unique station features (i.e., spatial features) to predict the station demand. The first model also does not include any embeddings for the stations' IDs. The latter model was trained on the residual demand not captured by the generic feature model to only learn the station-specific variation, by subtracting the predicted demand of the first model from the target demand during training. The final prediction for station-level demand of the second model in this group is the summation of these two models' test outputs.

Group 3: CNN-LSTM Models with station capacity: The third group also features two CNN-LSTM models. However, these models incorporate station capacity. This integration aims to evaluate the impact of station capacity on the overall performance of the models. The first model of this group again employs a comprehensive set of features, while the second model splits the feature set as is also done in group 2. Each group's configuration is carefully crafted to dissect the influence of different feature sets and structural components on the models' forecasting accuracy.

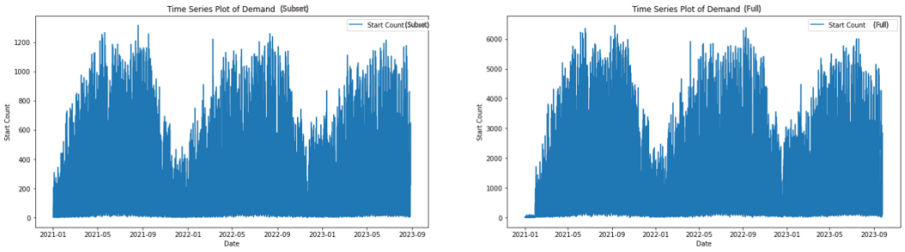
4.5.2 *Prediction window*

Ashqar et al. (2017) state that the usage of a 15-minute prediction window was found as optimal for forecasting subsequent bike pickups. This prediction window was also employed in the study of (Mehdizadeh Dastjerdi & Morency, 2022). However, the data collection in this study was conducted on an hourly basis, as aggregating at a 15-minute interval would quadruple the dataset size, rendering it impractical within the given time constraints. Therefore, the hourly aggregation was the timestep employed in this study.

Additionally, (Mehdizadeh Dastjerdi & Morency, 2022) highlights the superiority of utilizing the last 24 hours of historical data, equivalent to 24 timesteps, over other durations such as 6, 12, 48 hours, or one week as the input for forecasting the bike demand the next hour. Besides this, LSTMs are not able to handle such long sequences due to the exploding and vanishing gradient problem. For these reasons, one day (24 timestamps) was chosen as the input for the next hour demand predictions, denoted as $t + 1$ (with each timestep representing 1 hour). The dimension of the input is then (256, 24, 42). The first value is the batch size, the second value is the time step and the third is the number of features.

4.6 *Hyperparametertuning*

Due to time constraints in this thesis, comprehensive hyperparameter tuning across the entire dataset was not feasible. For example, tuning hyperparameters only for the LSTM model using the full dataset already exceeds 12 hours. To address this, a subset of approximately 50 stations, representing about 1.3 million rows, was selected randomly for hyperparameter tuning. Although this is a smaller portion of the data, it is still substantial enough for effective training and still demands considerable computational time. To ensure this subset is representative, the station demand is plotted against dates for both the subset and the full dataset in Figure 6. This plot shows almost an identical distribution of the demand between the subset demand (a) and the full demand (b), hence this indicates that the subset probably is representative of the full dataset.



(a) (b)

Figure 6: Demand distribution subset model (a) and full model (b)

After optimizing the hyperparameters on this subset, the model was trained on the remaining data, excluding the 50 stations used in tuning. This ensured that the model was evaluated on unseen data. The full dataset, minus the subset, includes over 200 stations and about 3.7 million rows, providing a solid basis for training and evaluation. This approach balances

the practical limitations of hyperparameter tuning with the need for a thorough model evaluation.

4.6.1 *Hyperparametertuning method*

Because of the challenge of computational resource limitations with a large amount of data in this study, particularly concerning RAM, a batch generator was employed for loading the data into the models. This method allowed for the streaming of data in manageable batches (in this study 256), ensuring that memory usage was kept low while still effectively feeding the model during the training and validation phases. This generator was employed in both the hyperparameter tuning and in training of the model.

For hyperparameter tuning, the Hyperband method was selected for its computational efficiency. It smartly allocates resources by training multiple models for a limited number of epochs, focusing on the most promising ones.

ARIMA

The decision on the parameters for our ARIMA model leveraged the Auto Arima function from the PMDarima package over manual tuning. The ARIMA model requires that the data is stationary. An autocorrelation plot showed a gradual decline in autocorrelation, which is a common characteristic of non-stationary (Figure 18, Appendix B). Furthermore, in the conducted exploratory research it was found that the data contains daily, monthly, and seasonal trends, which also is an indication of non-stationarity. However, the Augmented Dickey-Fuller test contradicted this by indicating stationarity with a p-value of 0.0. To reduce the chance of mistakes, the auto ARIMA approach was adopted over manual tuning.

The range of values was deliberately limited to manage computational resources effectively. Specifically, the parameters p, d, q were explored within the bounds of 1 to 3. The ARIMA model's performance was evaluated based on the Akaike Information Criterion (AIC), with lower values indicating a better model fit. The outcomes of the auto ARIMA function showed varying AIC values for different combinations of p, d, q. The best model, as per the lowest AIC value, turned out to be ARIMA(3,1,3).

LSTM

In the provided table (Table 3) on LSTM model hyperparameters, key options include the number of units, dropout rate, optimizer type, number of LSTM layers, and the learning rate. The dropout rate, varying from 0.2 to 0.6, is particularly important as it helps prevent overfitting by randomly deactivating a portion of neurons during training, thus enhancing the model's generalization capability. Additionally, the model employs ReLU

activation in the dense layer(s) when included. Table 3 gives an overview of the hyperparameters that were tested for the LSTM model.

Table 3: Hyperparameters Tuning Options for LSTM Model

Hyper-parameters	**Values**
Number of Units	32, 64, 96, 128
Dropout Rate	0.2, 0.4, 0.6
Number of LSTM Layers	1, 2
Learning Rate	0.1, 0.01, 0.001

CNN-LSTM

Following the insights from prior studies (Lin et al., 2018; Mehdizadeh Dastjerdi & Morency, 2022), the CNN-LSTM models were constructed with two convolutional layers, C^1 and C^2, followed by a single LSTM layer. The number of hidden layers was determined during hyperparameter tuning; if C^2 was found to be 0, then this means that the model only would use one hidden layer. These convolutional layers were stacked without a pooling layer, and the output of the second CNN layer was passed through a ReLU activation function. ReLu is a widely used activation function in the hidden layers of Neural Networks. In the CNN-LSTM models, 'same' padding was used in Conv1D layers to maintain the input sequence length through the convolutional process, ensuring consistent temporal dimensionality. Table 4 gives an overview of the hyperparameters that were tested for the CNN-LSTM models. The selected hyperparameters of both the LSTM and the CNN-LSTM models are given in Table 5. The models from groups 2 and 3 have the same hyperparameters.

Table 4: Hyper-parameter selection in CNN-LSTM models

Hyper-parameters	**Valuesl**
Filters 1st Conv1D Layer	{32, 64, 96, 128}
Kernel Size	{3, 5}
Filters 2nd Conv1D Layer	{0, 32, 64, 96, 128}
LSTM Units	{30, 50, 70, 90}
Dropout Rate	{0.2, 0.4, 0.6}
Learning Rate	{0.1, 0.01, 0.001}

Both of the models include an embedding layer. This embedding layer learned the embeddings for all the station IDs in the training set. All the station IDs that were present in the data were also in the training set.

Keras offers an Embedding layer that can be used to learn the embedding during training of the model. This layer requires that the input data be integer encoded so that each station ID is represented by a unique integer (Brownlee, 2021). The embedding layer had an input dimension of 263 (i.e., the number of stations). The rule of thumb for determining the embedding size is the number of unique categories, divided by 2, but no bigger than 50 (Muriuki, 2018). So, the size of the embedding was set to 50. The station ID was included as one of the inputs, these were passed through the embedding layer. The station ID was also loaded in the model in batches with the shape of (256,1), meaning that each batch contained 256 station ID's.

Table 5: Model structure with selected hyperparameters LSTM and CNN-LSTM models

CNN-LSTM		LSTM	
Layer (type)	Units/values	Layer (type)	Units/values
Conv1D	64	LSTM	96
Dropout	0.2	Dropout	0.2
LSTM	70	LSTM	96
Embedding	50	Embedding	50
Dropout	0.2	Dropout	0.2
Flatten		Flatten	
Repeat Vector	50	Repeat Vector	50
Concatenate	114	Concatenate	114
TimeDistributedDense	64	TimeDistributedDense	64
TimeDistributedDense	1	TimeDistributedDense	1

4.7 *Optimization*

In the process of hyperparameter tuning, various optimization algorithms were tested, including 'adam', 'sgd', and 'rmsprop'. Among these, 'rmsprop' performed the best for both the LSTM and CNN-LSTM model, yielding the most effective results on the validation data. Additionally, the learning rates were evaluated. For both models, a learning rate of 0.01 proved to be the most optimal. Lastly, the number of epochs was set by experiment for all of the models.

In the latter stages of hyperparameter tuning, it was noted that while training loss consistently decreased, validation loss fluctuated significantly, yet remained substantially lower than the training loss. This pattern, coupled with the implementation of early stopping, led to the models

halting at relatively low epochs. The disproportionately high training loss in comparison to validation loss indicated potential underfitting. To address this, the learning rate was reduced across all models to achieve a more gradual descent towards the global minimum. Consequently, this adjustment to a learning rate of 0.0001 resulted in an increased number of epochs and a steadier reduction in validation loss, enhancing the overall model performance.

4.8 *Model performance metrics*

To evaluate the algorithms' effectiveness, two standard metrics for regression problems were used: Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). RMSE is valuable because it gives more weight to larger errors, offering a clearer view of overall prediction accuracy. MAE, on the other hand, simply averages the absolute errors, providing an easy-to-understand measure of the average prediction error. While MAE helps in getting a straightforward understanding of the model's performance, RMSE will be our primary focus as it more accurately reflects the model's effectiveness in handling varied error magnitudes.

4.9 *Software*

All the models that were run, were run in Python version 3.10.12 using Google Colab with a GPU backend. In Appendix A, Table 8 gives all the programs and Python packages that were used in this thesis.

5 RESULTS

To address the primary question of the study regarding the capability of Neural Networks in forecasting bike-sharing demand at the station level, four distinct model groups were established. Table 6 in the study compares the predictive performance of these models.

Sequential interpretation of models' outcomes will commence with the baseline ARIMA, followed by the LSTM, the CNN-LSTM incorporating all variables, the ensemble CNN-LSTM model, and concluding with an exploration of the effects of including the station's capacity.

The station-level demand predictions require a tailored approach to data visualization. Since representing the demand for all stations would not be useful, two stations were selected at random to demonstrate the selected models' predictive performance. For these stations, the model's hourly predictions were plotted against the actual demand values for

August 22, 2023, covering the entire range from midnight to 11:00 PM. Additionally, the test data was also aggregated on a daily level to assess the models' performance. Furthermore, it is worth mentioning that since all values were standardized, the performance metrics, such as MEA and RMSE, are also presented in a standardized format. This means that the MEA can not be interpreted as the absolute difference in actual bikes.

Table 6: Performance comparison of selected models

Model	Test MAE	Test RMSE
Group (1): LSTM	0.5464	0.8654
Group (2): Without Capacity		
CNN-LSTM	0.5377	0.8244
$\text{CNN-LSTM}_{\text{generic}}$ + $\text{CNN-LSTM}_{\text{specific}}$	0.5509	0.8478
Group (3): With Capacity		
CNN-LSTM	0.5290	0.8000
$\text{CNN-LSTM}_{\text{generic}}$ + $\text{CNN-LSTM}_{\text{specific}}$	0.5483	0.8010
Group (4): ARIMA (Baseline model)	0.7140	1.1410

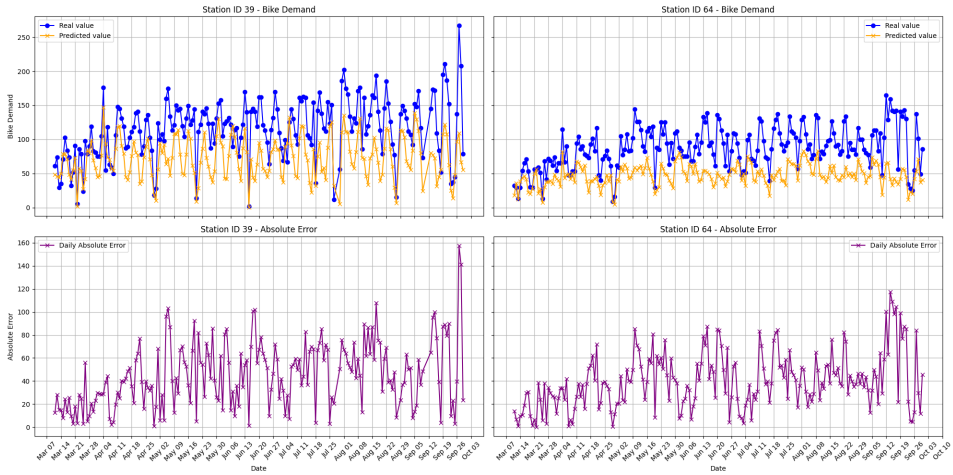
5.1 *ARIMA*

In the evaluation of the models, the ARIMA model is utilized as a baseline due to its historical prevalence in time series forecasting. In Table 6, it can be observed that the ARIMA model yields an MAE of 0.714 and an RMSE of 1.141, which does not match the superior performance of LSTM and CNN-LSTM models. This outcome is consistent with existing literature, which posits that ARIMA models, though effective for simpler or shorter time series, may not capture complex patterns in data spanning multiple years as efficiently as LSTM-based approaches. These findings underscore the utility of LSTM and CNN-LSTM models in handling the intricacies of long-term demand forecasting, benefiting from their ability to learn from large datasets and capture non-linear dependencies.

5.2 *LSTM*

Table 6 reveals that the LSTM model achieved an MAE of 0.5464 and an RMSE of 0.8654. This performance surpasses that of the baseline ARIMA model, yet does not reach the performance of the more complex CNN-LSTM models in terms of RMSE.

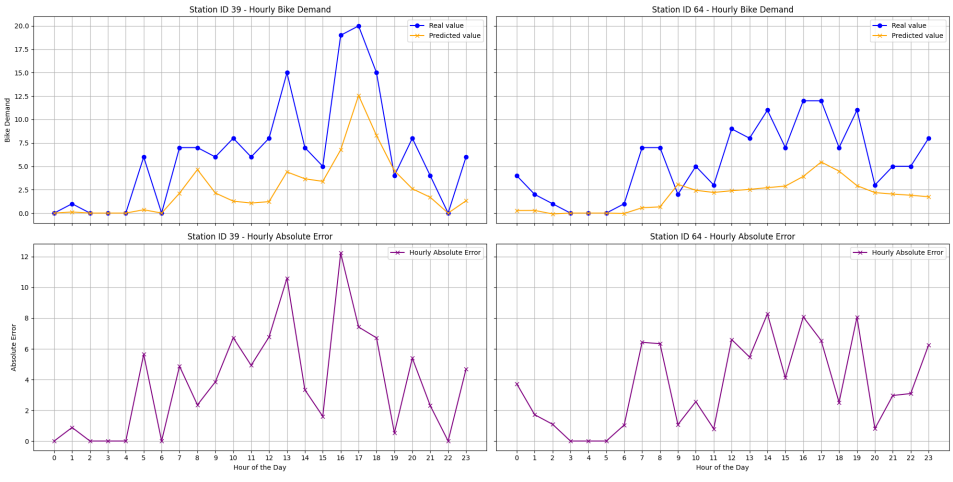
Figures 7 and 8, illustrating the actual versus predicted values for the selected stations, give a closer examination of the LSTMs predictive behavior. Figure 7 shows the model's ability to track the general demand trend, as evidenced by the overall pattern of predictions following the actual values. However, the consistent underestimation of demand, where predictions fall below actual values, suggests that the model's estimates may lack the complexity necessary to capture the variability of the data or between stations, which biases predictions toward the mean.



(a) (b)

Figure 7: Actual and predicted hourly data aggregated by day as well as the corresponding Absolute Error for selected stations on the test data by the LSTM model, with (a) station 39 and (b) station 64

In Figures 8, the model's ability to track temporal trends is apparent, as it successfully follows the demand patterns. Despite these broader movements, there is a clear bias towards the mean, with predictions notably lower than actual values. This indicates that while the model is adept at grasping general temporal patterns, it struggles to capture station-specific fluctuations, which is essential for precise station demand forecasting. The mean demand for the test set is about 7.9 bikes per hour.



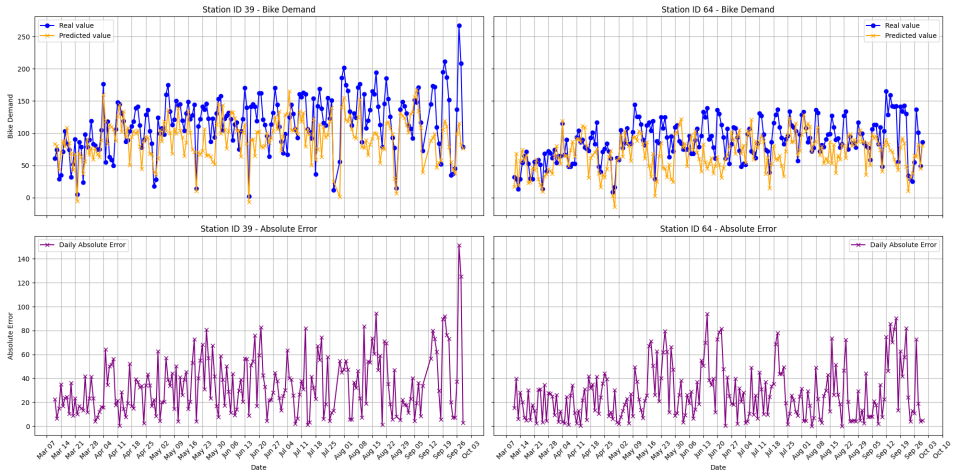
(a) (b)

Figure 8: Actual and predicted hourly data as well as the corresponding Absolute Error for selected stations on the test data by the LSTM model, with (a) station 39 and (b) station 64

A heatmap, presented in Figure 19 (Appendix C), which details the mean absolute error across different hours and days, further clarifies the model's performance fluctuations. Particularly during the afternoon peak commuting times on weekdays, the model struggles with the increased demand variability. In contrast, the more accurate weekend predictions suggest that the model performs better during periods of consistent demand, likely due to reduced variability. However, it's noteworthy that the MEA remains also relatively high on weekends.

5.3 *CNN-LSTM*

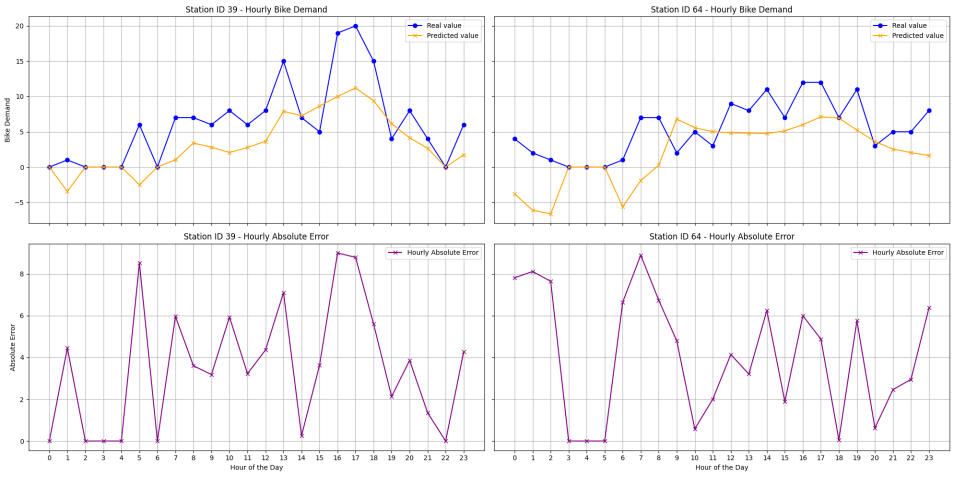
The results displayed in Table 6 indicate that the CNN-LSTM model, when factoring in station capacity, outperforms both the baseline ARIMA and LSTM models with an RMSE of 0.8000 and an MAE of 0.5290. This enhancement is apparent in the daily aggregated prediction plot in Figure 9, where predicted values align more closely with actual values, mostly keeping the absolute error under 60. This suggests that the model's hourly predictions aggregated by day are quite accurate, with the total daily prediction error remaining below 60 bikes. But still, also this model has a slight bias towards the mean, with most predicted values below the actual values.



(a) (b)

Figure 9: Actual and predicted hourly data as well as the corresponding Absolute Error for selected stations on the test data by the CNN-LSTM model, with (a) station 39 and (b) station 64

Despite this marked improvement, the model still encounters difficulties in capturing the usage patterns shown in Figure 10, particularly during rush hours when demand variability peaks. While it more accurately reflects the variances between different stations than the LSTM model, the model tends to default towards mean values during these times of high fluctuation. It does not as effectively model the sharp increases in usagen during these hours. This is further illustrated in the heatmap in Figure 20 (Appendix C), where the mean absolute error, though lower than that of the LSTM models for most hours, is still relatively high during these critical periods of high demand.



(a) (b)

Figure 10: Actual and predicted hourly data as well as the corresponding Absolute Error for selected stations on the test data by the CNN-LSTM model, with (a) station 39 and (b) station 64

In Figure 10, the CNN-LSTM model displays instances of negative demand, which is not possible in real life since the model does not account for bike returns. This anomaly is likely due to the standardization of the target variable, 'demand,' which can produce negative values if the actual demand is below the average used for scaling. The model's tendency to predict negative numbers during low usage times suggests it might be misinterpreting these quiet periods. While such negative values are not feasible in practice, they have been included in the output to provide a clear picture of the model's performance limitations and to highlight specific areas needing improvement in demand forecasting.

5.4 *CNN-LSTM with split*

The ensemble model, which combined predictions from two sub-models, yielded an MAE of 0.5483 and an RMSE of 0.8010, as indicated in Table 6. This model outperformed both the baseline ARIMA and the LSTM model in terms of RMSE. When compared to the CNN-LSTM model that processed all variables simultaneously, the ensemble model's MAE is marginally higher by 0.02, while the RMSE is about the same. In this thesis, the method of evaluation is the RMSE, hence we can say that the ensemble CNN-LSTM model performed equally to the normal CNN-LSTM model.

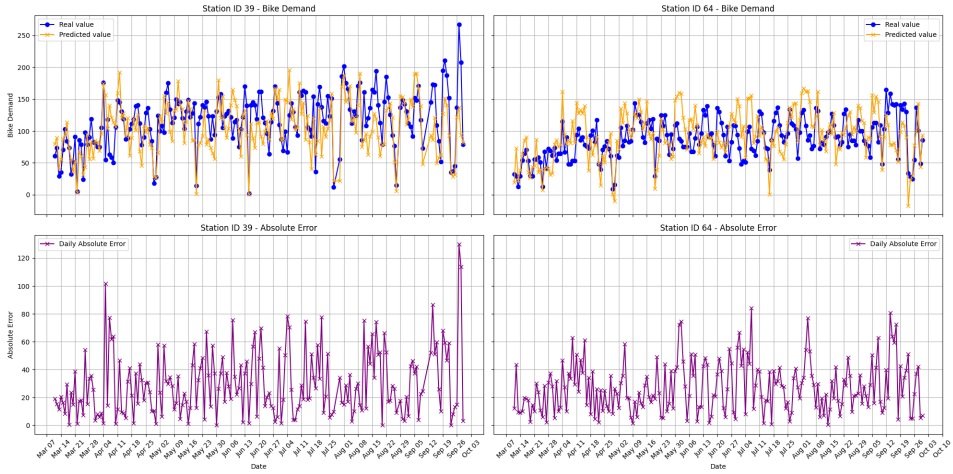
Table 7 shows the performance of the CNN-LSTM ensemble model, distinguishing between the generic and specific configurations. The CNN-$\text{LSTM}_{\text{generic}}$ model, which was trained on variables that are the same for

all stations, demonstrates commendable predictive accuracy, as indicated by the MAE and RMSE values of 0.5466 and 0.8642, respectively. This performance notably exceeds that of the ARIMA and the LSTM models. Additionally, the CNN-LSTM$_{\text{specific}}$ model achieved a further reduction in RMSE of approximately 0.06. It is worth mentioning that the CNN-LSTM$_{\text{specific}}$ model was trained on the residual demand of the CNN-LSTM$_{\text{generic}}$ model, therefore this model only predicts residual values. As a result, the performance metrics are higher.

Table 7: Performance comparison of the ensemble model

Model	Test MAE	Test RMSE
CNN-LSTM ensemble model		
CNN-LSTM$_{\text{generic}}$	0.5466	0.8642
CNN-LSTM$_{\text{specific}}$	1.1120	1.5308

A closer examination of the ensemble CNN-LSTM model's performance, as depicted in Figure 11, reveals that its daily aggregated predictions do slightly better than the normal CNN-LSTM model. It seems to more precisely track the actual demand, with fewer values below the actual predictions.

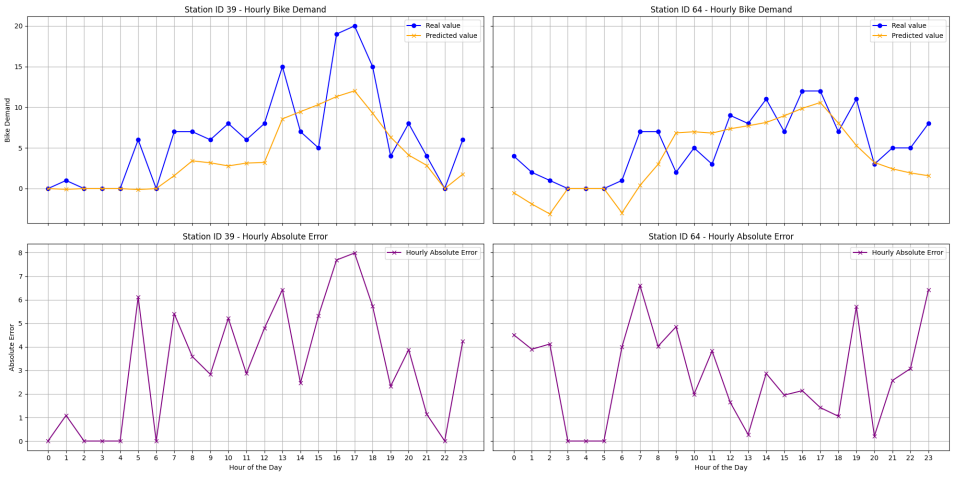


(a) (b)

Figure 11: Actual and predicted hourly data aggregated by day as well as the corresponding Absolute Error for selected stations on the test data by the ensemble CNN-LSTM model, with (a) station 39 and (b) station 64

In Figure 12, the ensemble CNN-LSTM model's hourly predictions look to more closely match actual demand, with all Absolute Error values for the chosen stations staying under eight. This is a slight improvement over

the standalone CNN-LSTM model, which sometimes had errors above eight. Despite this, the ensemble model, similar to the LSTM and CNN-LSTM models, struggles with accurately predicting the sudden spikes in demand during peak hours, often biasing to the mean. Lastly, Figure 12 demonstrates an improvement in predicting lower demand levels than the CNN-LSTM, with fewer values below zero.



(a) (b)

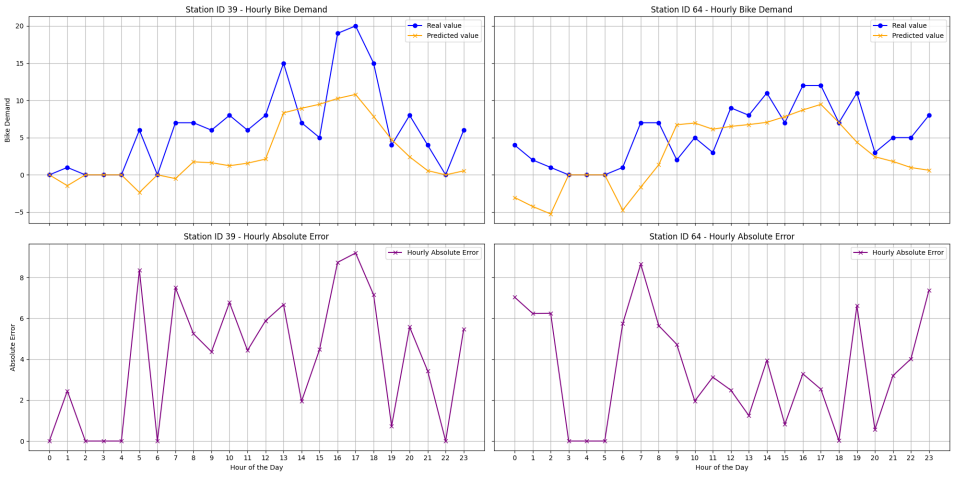
Figure 12: Actual and predicted hourly data aggregated by day as well as the corresponding Absolute Error for selected stations on the test data by the ensemble CNN-LSTM model, with (a) station 39 and (b) station 64

Figure 21 in Appendix C reveals that the error distribution of the ensemble CNN-LSTM model closely resembles that of the original CNN-LSTM model. The ensemble model's errors are a little bit more evenly spread throughout the day, in contrast to the normal CNN-LSTM model, which primarily errors during peak hours and weekends. However, these variances are very small and do not allow for definitive conclusions about the models' comparative performance.

5.5 *Station capacity*

The inclusion of station capacity significantly affects the models' performance, as evidenced by the performance metrics of the CNN-LSTM models. The CNN-LSTM model without the station capacity yielded an RMSE of 0.8244 and an MAE of 0.5377. Thus, the RMSE is 0.0244 higher than the model that included the station's capacity. This is strengthened by the ensemble CNN-LSTM model, which also omitted station capacity and registered a slight deterioration with an MEA of 0.5509 and an RMSE of 0.8478, as detailed in Table 6. This shows the importance of including

station capacity in the model to enhance prediction performance. Figure 13 presents the ensemble CNN-LSTM model's performance without considering capacity, selected due to its higher RMSE disparity. The plot reveals that most of the predicted values of these models tend to be lower than that of the model that included the capacity. Besides this, the plot shows a similar distribution.



(a) (b)

Figure 13: Actual and predicted hourly data aggregated by day as well as the corresponding Absolute Error for selected stations on test data, by the ensemble CNN-LSTM model without capacity, with (a) station 39 and (b) station 64

6 DISCUSSION

6.1 *Discussion of results*

This thesis aimed to find the extent to which Neural Networks can predict bike-sharing demand at the station level. While prior studies have predominantly focused on demand predictions at the cluster and city levels, station-level prediction remains less explored (Mehdizadeh Dastjerdi & Morency, 2022; Wang & Kim, 2018). To address this gap, this research was structured around a central main question: To what extent can Neural Networks predict the bike-sharing demand at the station level? This broad main question is supported by specific sub-questions. Following the presentation of results, a detailed discussion of these sub-questions is provided below.

Research Question 1: *To what extent can LSTM and CNN-LSTM networks predict the station-level BSS demand when compared to the baseline model?*

The findings in this study indicate that both the LSTM and CNN-LSTM models significantly outperform the ARIMA model. Notably, the CNN-LSTM model demonstrates superior predictive capabilities over the LSTM model. While the LSTM model effectively captures general temporal patterns, it looks like it struggled with station-specific dependencies. This aligns with existing literature, suggesting the LSTM network's proficiency in temporal data but limited capacity in spatial aspects pertinent to bike stations (Sherstinsky, 2020; Zhao et al., 2017). In contrast, from the generated plots for the two randomly selected stations it looked like the CNN-LSTM model was quite good in identifying both general trends and station-specific nuances, aligning with previous studies that highlight its superior performance (Ai et al., 2019; Ke et al., 2017; Mehdizadeh Dastjerdi & Morency, 2022). These studies described that when this model was enriched with additional input features it is superior in capturing the spatial-temporal correlations and is the most effective for station-level demand predictions. In this study, the CNN-LSTM model with the station capacity included was the best-performing model. However, it's important to note that during peak demand periods with high variability, while it did follow the general trend for specific stations, the CNN-LSTM model did not fully capture the sharp increases in usage. The complexity of station-level demand variability is a primary reason for the limited focus on this level of demand predictions in the literature (Kim, 2018). This issue is not confined to times of high demand; the model also appears to struggle with periods of low usage, often underestimating to the point of predicting negative demand.

Research Question 2: *To what extent is the RMSE of the model's influenced by the inclusion of the station's capacity?*

This aspect has been overlooked in comparable studies of Ai et al. (2019) and Mehdizadeh Dastjerdi and Morency (2022) that also used CNN-LSTM networks, but for cluster demand predictions. As Mehdizadeh Dastjerdi and Morency (2022) suggests, station usage might be influenced by available docking stations, potentially impacting model performance. This study reveals that incorporating station capacity significantly enhances the predictive performance of both standard and ensembled CNN-LSTM models, as evidenced by a marked decrease in RMSE. This finding supports the observations in El-Assi et al. (2017), which underscore the importance of station capacity in bicycle trip generation models.

Research Question 3: *What is the comparative predictive efficacy of a dual-model approach—consisting of generic and specific models against a singular CNN-LSTM model?*

The ensemble did not surpass the singular CNN-LSTM model, which integrated all variables, in terms of performance. It performed equally well in terms of RMSE as the CNN-LSTM model. This is in line with the findings of Gao et al. (2021), where the ensemble model also outperformed the ARIMA model and, in our case, the LSTM model. However, that study did not use a CNN-LSTM network to compare the results. The study by Gao et al. (2021) suggests the time sub-model has the biggest contribution to the ensemble's performance. Similarly, in this study, the generic model, which incorporates temporal and weather features predominantly appears to achieve the best performance. Visual inspection of the hourly trends for two randomly selected stations indicated that the ensemble model might be slightly more adept at capturing specific station trends. However, this observation does not translate into a statistically significant difference in performance metrics and, therefore, does not warrant definitive conclusions.

6.2 *Limitations*

As with every study, this thesis also has some limitations that should be mentioned. The first limitation arises from the inclusion of station capacity data, which was not available in the original dataset and was subsequently sourced from the "NYC OpenData" (2023) API. This API provided only the current station capacities, necessitating the assumption that these capacities remained constant over time. This assumption overlooks the potential for capacity changes over time, which could influence the predictive performance of the model. Additionally, due to time constraints, the hyperparameter tuning for the models was conducted on a subset of 50 stations, chosen at random. Although this subset appeared to be representative of the entire dataset upon visual inspection, there remains the possibility of disparities between the characteristics of the full dataset and the selected sample. Lastly, even though other studies say it's best to predict bike usage utilizing 15-minute intervals (Ashqar et al., 2017; Mehdizadeh Dastjerdi & Morency, 2022), this thesis used one-hour intervals. Using a 15-minute interval would make the dataset four times bigger, which is not feasible due to time constraints.

6.3 *Future work*

The importance of uncovering hidden correlations in bike-sharing demand, as noted by Lin et al. (2018), cannot be overlooked. This study employed embeddings to bring similar stations into closer proximity within the model's feature space. However, more sophisticated techniques, such as

graph convolutional networks that can learn hidden inter-station correlations, could potentially yield improvements. These methods were not explored here due to their complexity, but they present a valuable avenue for future research. This study reveals that the models struggle with capturing station-specific trends during high and low-demand hours. Future research should explore strategies to mitigate this issue, in addition to this study's approach of incorporating time-of-day dummies, such as 'afternoon rush' and 'morning rush', to enhance prediction accuracy. Additionally, there is room to investigate whether incorporating other features could enhance model performance at the station level. While this study focused on a set of core features, other research at the cluster level considered variables such as the number of bikes in an area (Ai et al., 2019), proximity to bike facilities (Faghih-Imani et al., 2014; Liang et al., 2022), and population density (Faghih-Imani et al., 2014). Exploring the impact of these additional factors could lead to even more accurate demand prediction models.

6.4 *Contribution*

This research addresses the gap in station-level demand predictions by applying LSTM and CNN-LSTM algorithms. The findings demonstrate that CNN-LSTM models can surpass traditional models such as ARIMA and standard LSTM models in terms of performance for station-level demand predictions. A significant discovery is that including the station capacity markedly enhances the model's ability to predict demand. These insights can aid decision-makers in improving bike redistribution efforts by providing a more nuanced understanding of station-specific demand patterns. However, caution is advised during peak hours, as the model exhibits some difficulty in accurately predicting these high-demand periods. Academics may also find these results valuable for developing more sophisticated station-level demand prediction models by using CNN-LSTM networks and including the station's capacity in the models.

7 CONCLUSION

This study investigated the capability of neural networks to forecast bike-sharing demand at individual stations. Specifically, it compared the performance of LSTM and CNN-LSTM models to a baseline ARIMA model across approximately 260 stations in New York City. The results indicate that both the LSTM and CNN-LSTM models outperform the ARIMA model significantly. With the latter achieving the best result. Moreover, incorporating data on the total capacity of the station further improved the

predictive performance of these neural network models. Additionally, the study introduced an ensemble CNN-LSTM model, which employed a two-tiered approach: a sub-model that learned from general station features such as weather and time, and a secondary sub-model that focused on the residuals of the first, incorporating station-specific features like location, nearby restaurants, and businesses. While this composite model performed well, it did not outperform the CNN-LSTM model that simultaneously considered all variables. Future research may build on these findings to refine and enhance the predictive models discussed in this thesis.

REFERENCES

Ai, Y., Li, Z., Gan, M., Zhang, Y., Yu, D., Chen, W., & Ju, Y. (2019). A deep learning approach on short-term spatiotemporal distribution forecasting of dockless bike-sharing system. *Neural Computing and Applications*, *31*, 1665–1677.

Albuquerque, V., Sales Dias, M., & Bacao, F. (2021). Machine learning approaches to bike-sharing systems: A systematic literature review. *ISPRS International Journal of Geo-Information*, *10*(2), 62.

Ashqar, H. I., Elhenawy, M., Almannaa, M. H., Ghanem, A., Rakha, H. A., & House, L. (2017). Modeling bike availability in a bike-sharing system using machine learning. *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 374–378.

Ay, M., Kulluk, S., Özbakır, L., Gülmez, B., Öztürk, G., & Özer, S. (2022). Cnn-lstm and clustering-based spatial–temporal demand forecasting for on-demand ride services. *Neural Computing and Applications*, *34*(24), 22071–22086.

Brownlee, J. (2021). How to use word embedding layers for deep learning with keras [Accessed: 2023-11-14].

Caggiani, L., Camporeale, R., Ottomanelli, M., & Szeto, W. Y. (2018). A modeling framework for the dynamic management of free-floating bike-sharing systems. *Transportation Research Part C: Emerging Technologies*, *87*, 159–182.

Campbell, A. A., Cherry, C. R., Ryerson, M. S., & Yang, X. (2016). Factors influencing the choice of shared bicycles and shared electric bikes in beijing. *Transportation research part C: emerging technologies*, *67*, 399–414.

Cao, M., Li, V. O., & Chan, V. W. (2020). A cnn-lstm model for traffic speed prediction. *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 1–5.

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

Chatgtp. (2023).

Chen, L., Zhang, D., Wang, L., Yang, D., Ma, X., Li, S., Wu, Z., Pan, G., Nguyen, T.-M.-T., & Jakubowicz, J. (2016). Dynamic cluster-based over-demand prediction in bike sharing systems. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 841–852.

Citi bike system data [Accessed on: April 20, 2023]. (n.d.).

Daum, F., & Huang, J. (2003). Curse of dimensionality and particle filters. *2003 IEEE aerospace conference proceedings (Cat. No. 03TH8652)*, *4*, 4_1979–4_1993.

Du, Y., Deng, F., & Liao, F. (2019). A model framework for discovering the spatio-temporal usage patterns of public free-floating bike-sharing system. *Transportation Research Part C: Emerging Technologies*, *103*, 39–55.

El-Assi, W., Salah Mahmoud, M., & Nurul Habib, K. (2017). Effects of built environment and weather on bike sharing demand: A station level analysis of commercial bike sharing in toronto. *Transportation*, *44*, 589–613.

Eren, E., & Uz, V. E. (2020). A review on bike-sharing: The factors affecting bike-sharing demand. *Sustainable cities and society*, *54*, 101882.

Etienne, C., & Latifa, O. (2014). Model-based count series clustering for bike sharing system usage mining: A case study with the vélib'system of paris. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *5*(3), 1–21.

Faghih-Imani, A., Eluru, N., El-Geneidy, A. M., Rabbat, M., & Haq, U. (2014). How land-use and urban form impact bicycle flows: Evidence from the bicycle-sharing system (bixi) in montreal. *Journal of transport geography*, *41*, 306–314.

Gao, Y., Zhang, M., Chen, J., Han, J., Li, D., & Qiu, R. (2021). Accurate load prediction algorithms assisted with machine learning for network traffic. *2021 International Wireless Communications and Mobile Computing (IWCMC)*, 1683–1688.

Guo, C., & Berkhahn, F. (2016). Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*.

Kabak, M., Erbaş, M., Cetinkaya, C., & Özceylan, E. (2018). A gis-based mcdm approach for the evaluation of bike-share stations. *Journal of cleaner production*, *201*, 49–60.

Ke, J., Zheng, H., Yang, H., & Chen, X. M. (2017). Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation research part C: Emerging technologies*, *85*, 591–608.

Kim, K. (2018). Investigation on the effects of weather and calendar events on bike-sharing according to the trip patterns of bike rentals of stations. *Journal of transport geography*, *66*, 309–320.

Lai, G., Chang, W.-C., Yang, Y., & Liu, H. (2018). Modeling long-and short-term temporal patterns with deep neural networks. *The 41st international ACM SIGIR conference on research & development in information retrieval*, 95–104.

Liang, Y., Huang, G., & Zhao, Z. (2022). Bike sharing demand prediction based on knowledge sharing across modes: A graph-based deep learning approach. *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 857–862.

Lin, L., He, Z., & Peeta, S. (2018). Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. *Transportation Research Part C: Emerging Technologies*, *97*, 258–276.

Mehdizadeh Dastjerdi, A., & Morency, C. (2022). Bike-sharing demand prediction at community level under covid-19 using deep learning. *Sensors*, *22*(3), 1060.

Muriuki, G. (2018). Deep learning; personal notes part 1 lesson 4: Structured learning, natural language processing, collaborative filtering. dropout, embeddings, back prop through time. [Accessed: 2023-11-14].

Narmadha, S., & Vijayakumar, V. (2021). Spatio-temporal vehicle traffic flow prediction using multivariate cnn and lstm model. *Materials today: proceedings*.

Nyc opendata [Accessed: 30-10-2023]. (2023).

Open-meteo [Accessed: 30-10-2023]. (2023).

Openstreetmap [Accessed: 30-10-2023]. (2023).

Pan, Y., Zheng, R. C., Zhang, J., & Yao, X. (2019). Predicting bike sharing demand using recurrent neural networks. *Procedia computer science*, *147*, 562–566.

Pratt, K. (n.d.). *Vehicle-grid integration program*. https://www.energy.ca.gov/programs-and-topics/programs/vehicle-grid-integration-program#:~:text=Vehicle%2Dgrid%20integration%20(VGI),still%20meeting%20drivers'%20mobility%20needs. (accessed: 21.09.2023).

Ren, S., Luo, F., Lin, L., Hsu, S.-C., & Li, X. I. (2019). A novel dynamic pricing scheme for a large-scale electric vehicle sharing network considering vehicle relocation and vehicle-grid-integration. *International Journal of Production Economics*, *218*, 339–351.

Sathishkumar, V., Park, J., & Cho, Y. (2020). Using data mining techniques for bike sharing demand prediction in metropolitan city. *Computer Communications*, *153*, 353–366.

Shaheen, S. A., Guzman, S., & Zhang, H. (2010). Bikesharing in europe, the americas, and asia: Past, present, and future. *Transportation research record*, *2143*(1), 159–167.

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, *404*, 132306.

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018). A comparison of arima and lstm in forecasting time series. *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, 1394–1401.

Sohrabi, S., Paleti, R., Balan, L., & Cetin, M. (2020). Real-time prediction of public bike sharing system demand using generalized extreme value count model. *Transportation Research Part A: Policy and Practice*, *133*, 325–336.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. *Proceedings of the AAAI conference on artificial intelligence*, *31*(1).

Wang, B., & Kim, I. (2018). Short-term prediction for bike-sharing service using machine learning. *Transportation research procedia*, *34*, 171–178.

Xu, C., Ji, J., & Liu, P. (2018). The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets. *Transportation research part C: emerging technologies*, *95*, 47–60.

Xu, H., Pu, P., & Duan, F. (2018). Dynamic vehicle routing problems with enhanced ant colony optimization. *Discrete Dynamics in Nature and Society*, *2018*, 1–13.

Yang, Y., Heppenstall, A., Turner, A., & Comber, A. (2020). Using graph structural information about flows to enhance short-term demand prediction in bike-sharing systems. *Computers, Environment and Urban Systems*, *83*, 101521.

Yu, L., Feng, T., Li, T., & Cheng, L. (2023). Demand prediction and optimal allocation of shared bikes around urban rail transit stations. *Urban Rail Transit*, *9*(1), 57–71.

Zhang, H., Dauphin, Y. N., & Ma, T. (2019). Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321*.

Zhao, Z., Chen, W., Wu, X., Chen, P. C., & Liu, J. (2017). Lstm network: A deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, *11*(2), 68–75.

APPENDIX A

Table 8: Summary of tools and packages used

Tool/Package	**Version**
Jupyter	6.4.8
Google Colab	-
Overleaf	-
ChatGPT	4
Python (3.10.12)	
Keras	2.10.0
Matplotlib	3.6.3
Numpy	1.23.5
Pandas	1.5.3
Scikit-learn	1.2.2
Tensorflow	2.10.1
Seaborn	0.12.2
Geopandas	0.13.2
Shapely	2.0.2
Statsmodels	0.14.0
Pmdarima	2.0.4

APPENDIX B

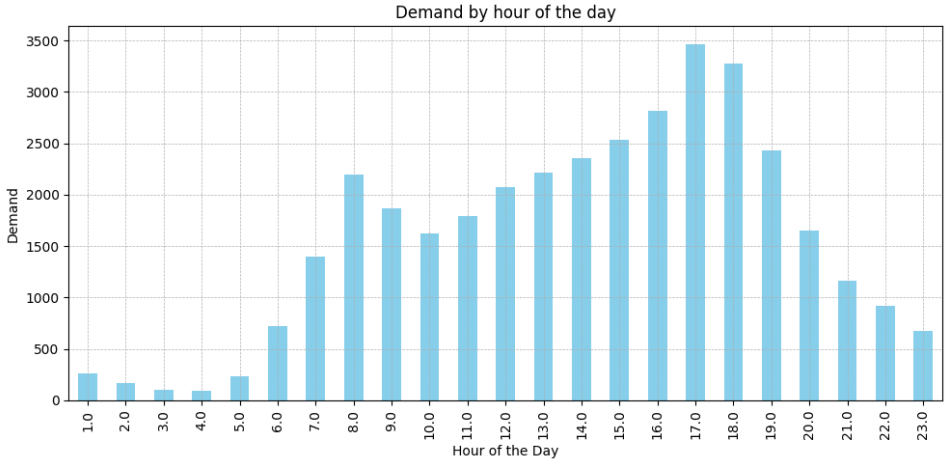


Figure 14: Total average demand by hour of the day

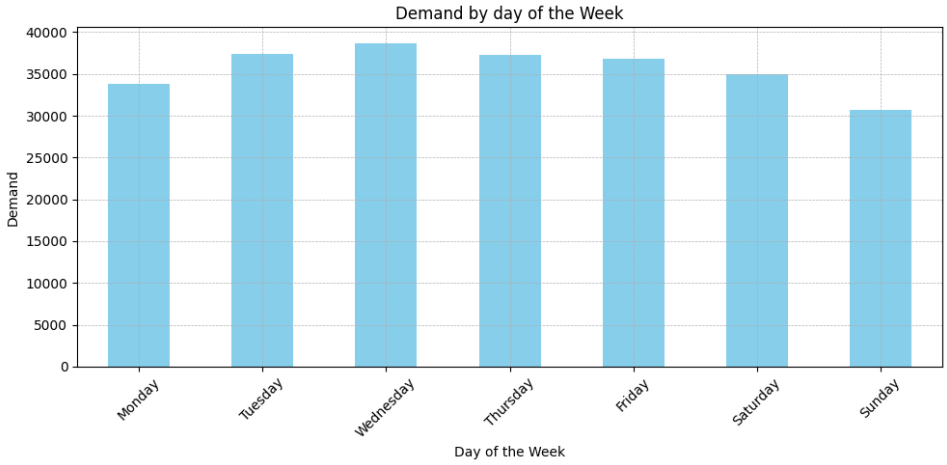


Figure 15: Total average demand by day of the week

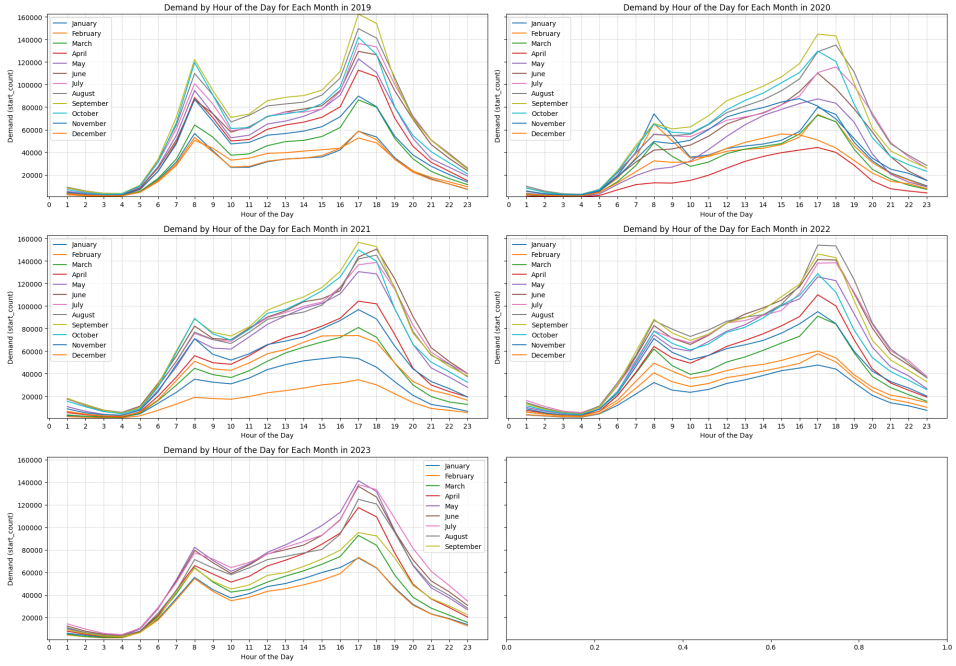


Figure 16: Total demand by hour of the day and grouped by day of the month

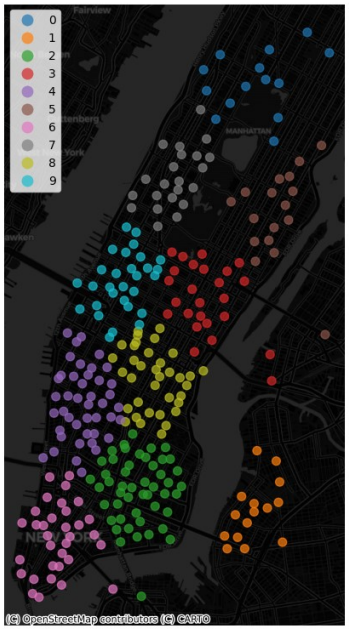


Figure 17: Bike stations clusters with K means clustering

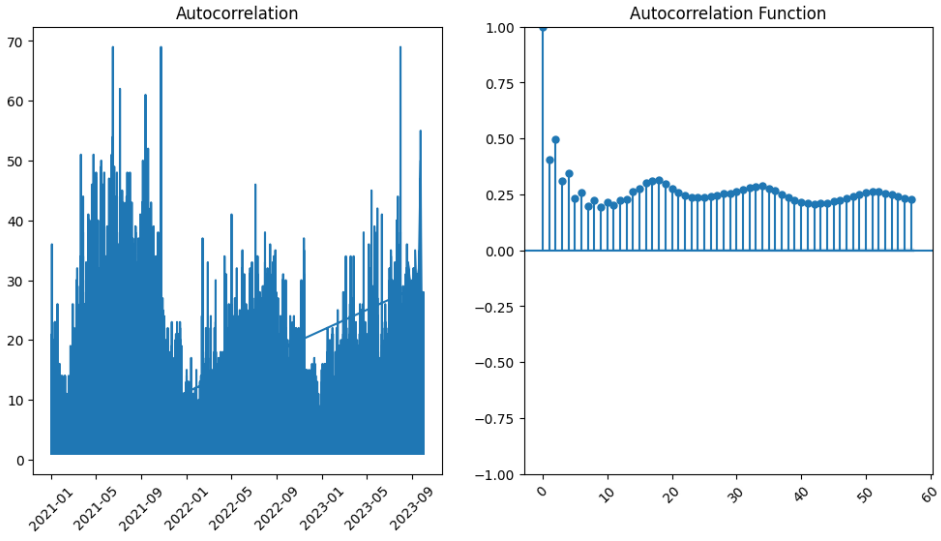


Figure 18: ACF plot showing the autocorrelation

APPENDIX C

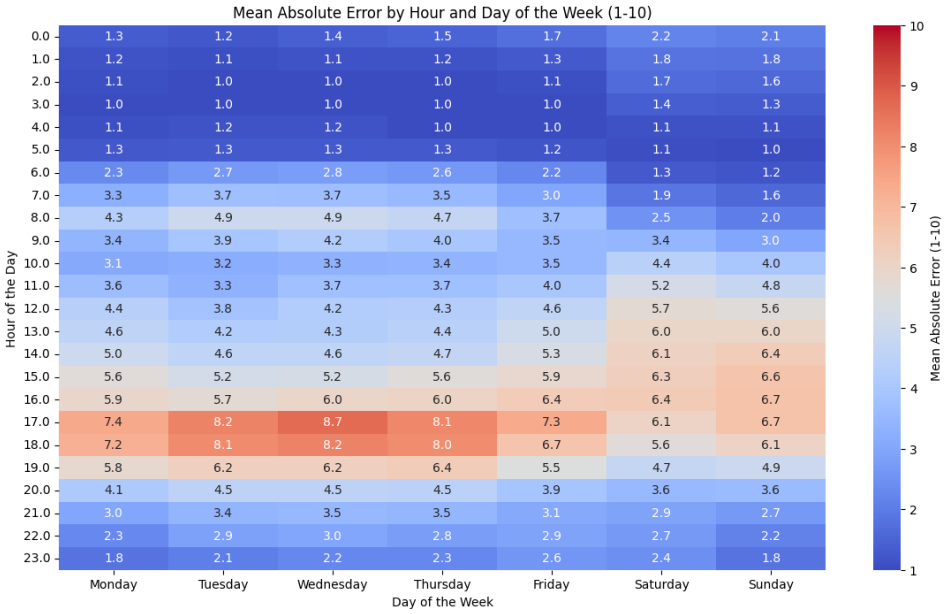


Figure 19: Heatmap showing the Mean Absolute error for the LSTM prediction model grouped by time and hour of the day

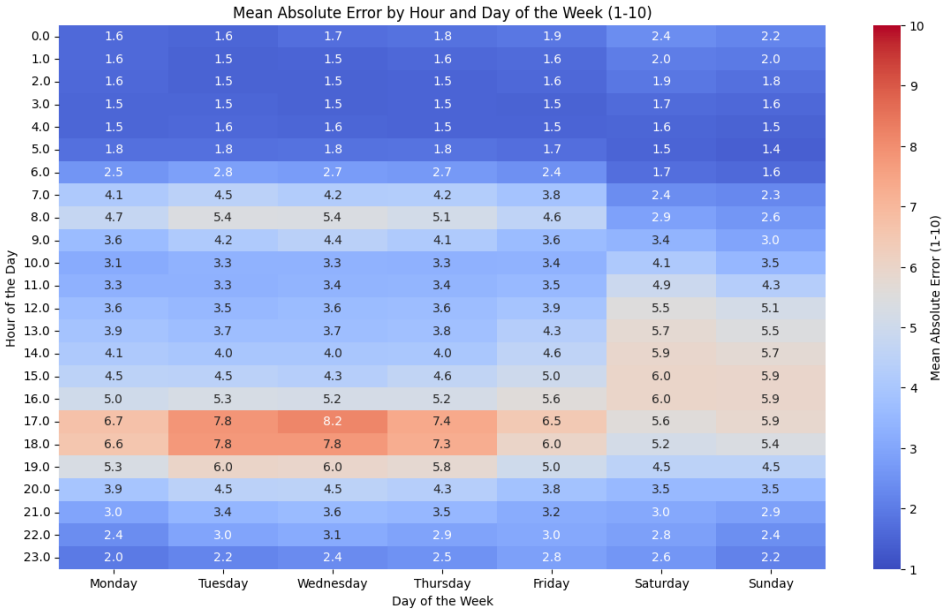


Figure 20: Heatmap showing the Mean Absolute error for the CNN-LSTM prediction model grouped by time and hour of the day

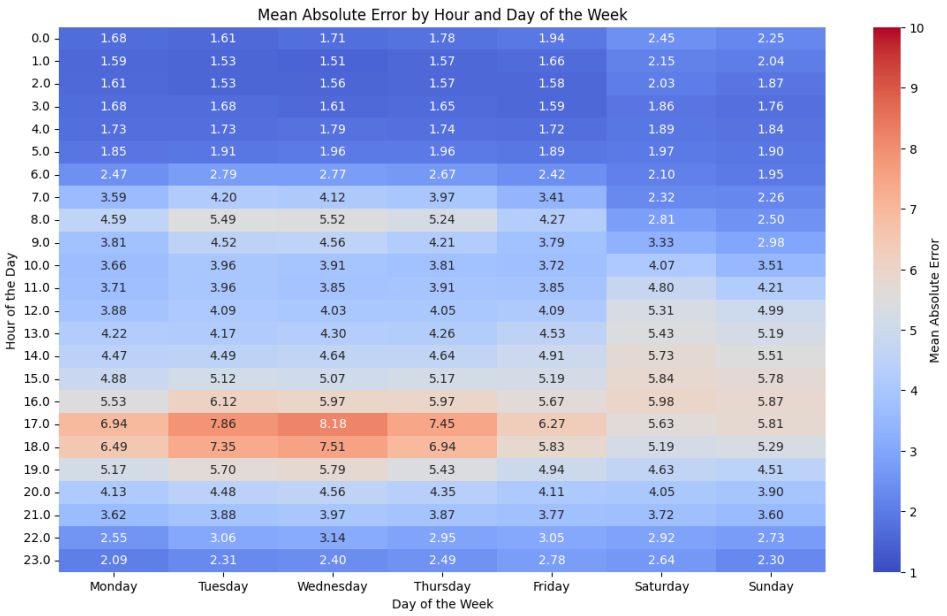


Figure 21: Heatmap showing the Mean Absolute error for the ensemble CNN-LSTM prediction model grouped by time and hour of the day