Learning Demand Functions for Bike-Sharing Using Spatio-Temporal Graph Neural Networks

Alexander Julius Busch
University of Hamburg
Hamburg, Germany

Kaifeng Lu
University of Toronto
Toronto, Canada

***Abstract*—Bike-sharing usage prediction has been implemented and analyzed using a variety of models, ranging from linear and logistic regression models with extensive spatial features [[1]], decision features, ARIMA [[2]] to deep learning models with convolutional [[2]] and graph features [[3], [4]]. However, modeling and prediction of the underlying demand function that drives the usage has not been rigorously attempted to the best of our knowledge, possibly due to the ill-posed nature of the problem. Our goal is to learn a demand function from data, that is suitable for short term demand prediction, such as adaptive pricing applications. We propose defining properties of a demand function and extend the biologically inspired Spatio-Temporal Graph Neural Network (STGAT) architecture from traffic prediction to jointly predict both the actual usage rate and extrapolate a suitable demand function. We analyze predictive performance on a subset of historic ridership data in Toronto. We analyze measures of demand prediction comparing the adapted base STGAT architecture, an upscaled variant and a variation with a transformer backend. We find that the learned demand function successfully encodes the defined axiomatic aspects of the demand. Also, on the investigated data, all three models show very similar performance, significantly outperforming a linear baseline.**

***Index Terms*—Bike-Sharing, GNN, STGAT, Demand Prediction**

I. Introduction

Dock-based bike-sharing, i.e. a system of stations where bicycles are docked and taken from are a common way of implementing a bicycle sharing system. As bicycle sharing becomes one of the most popular ways to commute, it plays a significant role in the public transportation system. Considered for the city of Toronto, according to the Bike Share Toronto 2023 Business Review, the total number of rides in 2023 is estimated to be about 5.5 million, projected to 2025 to become more than 6.2 million [5]. The total number of stations deployed is planned to be more than $1,000$ with more than $10,000$ bikes available. However, the demand varies greatly between location, weekday, season and other factors, which leads to imbalances and congestions in the system. This results in customer dissatisfaction and unreliability of the system, jeopardizing the central role of bicycle in reaching emission-neutral transportation and providing convenience.

To properly implement dynamic solutions, such as adaptive dynamic pricing and terminal extensions, the demand needs to be reliably predicted. Since bike-sharing usage fluctuates nontrivially based on a multitude of different influences, such as weekday, daytime, events, weather and more features, machine learning and especially neural network approaches have been successfully employed to predict general usage [1], [2], [4]. However, only predicting the number of bikes taken in or out is not sufficient for implementing a dynamic pricing system: For example, when a small station is predicted to be completely full at a time, how much should we incentivize taking out bikes? The demand in this case could range from zero (e.g. in the night) to almost arbitrarily high (e.g. during rush-hours in the day), but this quantity can not be directly computed from a purely predictive model.

In this article, our goal is to learn a demand function from data that quantitatively encodes user demand. For this, we extend the STGAT [6] prediction model to simultaneously predict usage and an infered demand function and define an adapted loss function to capture the notion of the demand. Notably, it is not our goal to outperform other purely predictive approaches, but improve the learned demand function via simultaneous learning of the pure prediction task and verify the model capability.

The structure of this article is as follows:

- We give an overview on other approaches on bike-sharing usage prediction and the STGAT architecture.
- We then mathematize the notion of a demand function and formalize the bike-sharing joint prediction task.
- In the following section, we translate the notion of a demand function into a proper regularizer and describe our variation of the STGAT architecture and the data processing.
- We then compare the predictive performance of our model to a linear baseline. We investigate its quality over the prediction horizon and show prediction results. We qualitatively compare different modeling choices in the resulting demand model and evaluate quantitative aspects of the demand model.
- We conclude by discussing model advantages and disadvantages and give directions for future research.

A. Motivation

For an illustration of the difference between demand and usage, consider a bike-sharing system with docks. Bikes can be taken out at a dock and have to be docked in back at any dock, with the price of the ride being proportional to the time between the dockings. Consider a user that wants to go from A to B at some day, possibly to reach their workplace or possibly to spontaneously take a ride to a park as leisure activity. In the case that there are bikes available, the user takes one and the demand is equal to the number of bikes taken out at that station. However, if the station is empty, the user might either walk to a nearby station, take another transportation medium or cancel the planned ride all together. In this case, the demand can not be inferred from the rate, but probably exhibits other regularities: The station might not be full at similar dates and the local demand is greatly defined by people commuting regularly to work, or other complex regularities. However, manually extracting the components of the demand is infeasible: From the author's own experience with the Toronto bike-sharing system, what people do when a station is full depends on weather, close stations, personal and time considerations, daytime and other factors. Thus we propose to use a deep learning approach to learn a suitable demand function from usage data using proper regularization.

For those familiar with the notion of counterfactual reasoning, the demand can be seen as counterfactual quantity: It is the rate of bikes taken in or out if the station would not have been empty at that point.

II. Literature Review

A. Scope and Overview

Due to the vastly growing number of bike sharing systems, bike sharing usage prediction has been investigated by a number of authors in the last 15 years. While classical machine learning approaches such as regression and boosting have been used in the early developments, in recent years, deep learning approaches have been found to give significant advantages in the forecasting domain [3]. Based on the literature, we compare the following machine learning approaches to bike-share prediction. Notably our approach differs from all of them in two aspects:

- We predict rates obtained by a gaussian filter, providing a localized smooth average for a good demand prediction.
- As defined in Section III, we use a novel loss function that allows extrapolating demand to time horizons where the demand cannot be infered due to full or empty stations.

B. Classification Tree

We organize the approaches in a tree in Fig. 1 for a schematic overview.

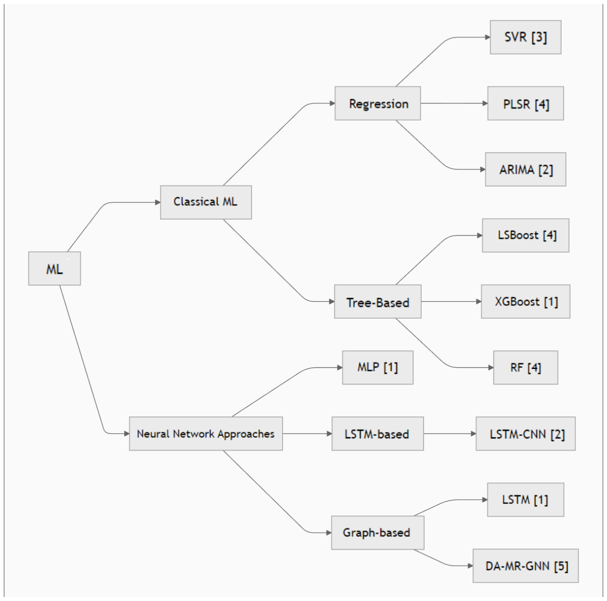

Fig. 1: A classification tree of the employed ML approaches

C. Articles

a) *Bike-Sharing Demand Prediction at Community Level under COVID-19 Using Deep Learning [2]:*

This article investigates short-term bike-sharing demand forecasting in Montreal. The authors compare deep learning techniques for predicting bike pickups 15 minutes ahead in six communities identified within the city's bike-sharing network. The study compares the performance of LSTMs (Long Short Term Memory, a recurrent network type), CNN (Convolutional Neural Network)-LSTM hybrids, and ARIMA (Auto-Regressive Integrated Moving Average). The authors trained the models based on two main feature attributes: historical demand data in 15-minute intervals and weather conditions. They found the CNN-LSTM hybrid model outperforms the other models significantly due to leveraging spatial information, and that especially ARIMA suffers from overfitting. An advantage of the ARIMA model is found to be its interpretability.

Similar to their approach, our base STGAT model incorporates the LSTM architecture in order to integrate historic data, however, the CNN architecture needs to operate on larger scale communities and cannot directly leverage the graph structure natural to bike-sharing stations and necessary for precise demand prediction. Notably our approach does not encode weather features.

b) *Using graph structural information about flows to enhance short-term demand prediction in bike-sharing systems [3]:*

In this paper, the authors investigate the influence of graph features on improving short-term bike-sharing demand prediction. It focuses on incorporating graph-based features derived from flow interactions such as Out-strength, In-strength, Out-degree, In-degree, and PageRank to improve prediction accuracy. It compares three machine learning models: XGBoost, Multi-Layer Perceptron (MLP), and LSTM. This study found that including graph-based features significantly improves model performance compared to using traditional features like meteorological data alone. The authors compare XGBoost, MLP and LSTM models and found the LSTM model to be the most effective for incorporating complex graph based features, while the overall performance of XGBoost is nearly comparable.

- The XGBoost architecture was found to provide good performance with little parameter tuning, while being less effective at handling complex time dependencies.
- The LSTM architecture has significantly higher training complexity compared to XGBoost.

Most notably, the graph-based features where found to improve the performance significantly. While our approach does not directly encode more complex graph features, the graph attention layer operating on the spatial closeness graph allows the STGATmodel to learn complex graph-based features, as shown in [7].

c) *Modeling bike-sharing demand using a regression model with spatially varying coefficients [8]:*

This article focuses on investigating how various factors such as land use, socio-demographic attributes, and transportation infrastructure influence bike-sharing usage at different stations. Notably, they define a graph model, where the data for each bike-sharing cluster is accumulated according to its catchment region. The catchment region is calculated using Thiessen polygons, and ensured to be non-overlapping. The authors propose a spatially varying coefficients (SVC) regression model that accounts for local spatial effects, unlike previous regression models that assume the factors are spatially homogeneous.

Most notably, their graph model is based on connectivity between station clusters. This sophisticated modeling approach allows to generalize to areas instead of precise station locations, which allows generalizing to catchment areas and thus new station locations, allowing planning stations in areas where the predicted usage is high.

While our approach does not allow generalization to new locations, this extension could be a valuable extension in order to leverage the model for planning problems. However, their approach does not regress on previous historic data, which limits its applicability to real time, adaptive prediction, but makes it rather suitable for general analysis purposes.

d) *Modeling Bike Availability in a Bike-Sharing System Using Machine Learning [1]:*

This paper explores predicting bike availability at San Francisco Bay Area Bike Share stations using machine learning algorithms. The authors apply three methods: Random Forest (RF) and Least Squares Boosting (LSBoost) for univariate regression and Partial Least-Squares Regression (PLSR) for multivariate regression. They found that factors like station neighbors, time of prediction, and weather conditions are significant in predicting bike availability. They found the random forest to significantly outperform the other model in MAE. Advantages of the all of the approaches, especially the RF model are its robustness to overfitting and simplicity in tuning, compared to deep learning approaches. However, the RF model requires independent observations, with performance decreasing significantly as the prediction horizon increases.

e) *Cross-Mode Knowledge Adaptation for Bike Sharing Demand Prediction Using Domain Adversarial Graph Neural Networks [4]:*

This recent paper focuses on improving the state of the art in bike sharing demand prediction by integrating additional features from other transport modes, here subway and ride-hailing traffic data, all from New York City. In order to integrate this heterogeneous graph structured data, recurrent CNNs are combined with graph convolutions to yield transport embeddings. The paper also introduces an adversarial training principle to learn these embeddings, such that the embeddings learned are optimized to be indistinguishable. These embeddings are then fed into multiple GNNs, combined into a single representation, and passed to a final prediction layer.

Notably the architecture allows extension to integrate very heterogeneous other modes of transportation significant to bike-sharing that would otherwise be very difficult to integrate. However, it is highly complex and depends on other transportation data, which makes it unsuitable for our problem. Similarly to our STGAT models, it uses GNN layers to process graph based data, which allows to learn complex graph-based features. Adapting the STGAT architecture to use their deeper GNN architecture might improve the prediction accuracy significantly.

III. Problem Formulation and Modeling

A. Axioms of Demand

From the described characteristics of the demand in Section I.A, we postulate the following properties of an ideal demand function, which we translate into a loss minimization problem for machine learning.

In the following, $\mathrm{IR}(s,t), \mathrm{OR}(s,t)$ denote the rates of bikes docked in or taken out at station $s \in \mathbb{N}$ and time $t \in \mathbb{R}$ and ID, OD the respective demand functions. The predicates $\mathrm{atmax}(s,t), \mathrm{atmin}(s,t)$ are defined to be 1 if the station is at its maximum capacity or empty respectively, and 0 otherwise.

- If the station is empty at a time, the out-demand should be greater or equal than the bikes taken out: $\mathrm{OD}(s,t) \geq \mathrm{OR}(s,t) \;\text{ if }\; \mathrm{atmin}(s,t)$
- If the station is completely full, the in-demand should be lower than the rate of bikes docked. $\mathrm{OD}(s,t) \geq \mathrm{OR}(s,t) \;\text{ if }\; \mathrm{atmax}(s,t)$
- Otherwise, the demand should be equal to the respective in or out rates:
  - $\mathrm{ID}(s,t) = \mathrm{IR}(s,t) \;\text{ if }\; \text{not } \mathrm{atmax}(s,t)$
  - $\mathrm{OD}(s,t) = \mathrm{OR}(s,t) \;\text{ if }\; \text{not } \mathrm{atmin}(s,t)$
- Both in-demand and out-demand functions should be smooth, i.e. its intuitively it should not change abruptly. This can be modeled in several different ways. We propose to demand that the third derivative is small, i.e. $\frac{d^3}{\mathrm{d}t^3}(\mathrm{ID}(s,t) + \mathrm{OD}(s,t)) \ll 1$.

  The motivation for this is that the third derivative is the smallest derivative providing a natural shape for demand. Notably, the functions with minimal third derivatives are quadratic functions, while functions minimizing the first or second derivative are piecewise constant or affine linear functions respectively, which would be an unnatural way to extend a demand function. The minimization of the third derivative can be seen as analogous to motion planning in robotics, where one minimizes jerk (the third derivative of the position) in order to obtain a smooth trajectory with small changes in acceleration.

B. *Relaxation to Loss Function*

From the ideal properties, we construct an analogous relaxed loss function, which allows to model the problem as minimization problem. We define the loss in terms of three components:

- the rate error $\left(\widehat{\mathrm{IR}} - \mathrm{IR}\right)^2 + \left(\widehat{\mathrm{OR}} - \mathrm{OR}\right)^2$
- the demand constraint violation

$$\begin{gathered}\left(\mathrm{IR} - \widehat{\mathrm{ID}}\right)^2 \cdot (\mathrm{atmax} \cdot [\mathrm{IR} \geq \mathrm{ID}] + 1 - \mathrm{atmax}) \\ + \left(\mathrm{OR} - \widehat{\mathrm{OD}}\right)^2 \cdot (\mathrm{atmin} \cdot [\mathrm{OR} \geq \mathrm{OD}] + 1 - \mathrm{atmin})\end{gathered} \quad (1)$$

- the smoothness violation $S_3\left(\widehat{\mathrm{ID}}\right)^2 + S_3\left(\widehat{\mathrm{OD}}\right)^2$, where S_3 is a numerical finite difference approximation of the third derivative.

The final loss is now formed as mean over stations and times of the previous terms:

$$\frac{1}{N_s \cdot N_t} \sum_{s,t} \mathrm{RateError} + \mathrm{DemViol} + \alpha \cdot \mathrm{SmoothViol}, (2)$$

where $\alpha \in \mathbb{R}$ is a regularization constant and we used abbreviations to denote the error components.

IV. Proposed Solution

In order to suitably solve the minimization problem, we adapt the spatio-temporal graph attention neural network architecture (STGAT) from [7]. We base our adaptation on the open implementation from [9]. Generally, a graph neural network (GNN) is a biologically inspired approach to machine learning, which operates on graph-structured data. The fundamental layer employed in a graph neural network is a graph convolutional layer (GCN), which computes output node features by computing and then aggregating features from each incoming node. The graph attention layer (GAT) employed in the STGAT is an extension of this layer, which weights the computed features by a learned attention score in order to compute a more refined representation. Notably, the fundamental principle of an attention mechanism, itself biologically inspired, has proven to be successful in many machine learning domains.

The STGAT architecture in [7] is constructed to predict car traffic velocities at measurement points, given historical velocities over all measurement points as graph. We adapt the final linear layer to give bike in-, out-rate and in-, out-demand predictions. We choose to investigate the STGAT architecture because of its performance in predicting roughly the next 45 minutes, given the last hour of information, which is a time horizon that would be useful for demand prediction for a dynamic pricing problem, and the similarity of the problems.

Additionally, because bike-sharing depends significantly on time feature information [[10]],

The base network consists of the following layers:

- A graph attention layer with 8 heads, followed by dropout.
- An LSTM layer with hidden size 32, which takes the reshaped historic data over all nodes. As modification of the original architecture, we concatenate additional daytime and day of week features to this layer's input. We encode both with sinusoidal encodings due to their periodic nature.
- A second LSTM layer with hidden size 128.
- A final linear layer operating on the last LSTM prediction, outputting information for all 9 timepoints.

As a variation of the LSTM-based standard STGAT architecture, we additionally investigate a version, where we replace the LSTMs with a transformer, consisting of four blocks. The transformer architecture can be seen as a successor of RNNs, fundamentally building on multihead attention and improving parallelizability of training.

A. *Modeling Details*

We use the historic bike-sharing data provided from the city of Toronto's open data portal. For all computations, we use the data for the full month May 2024. For the station geographical locations, we use the current live information provided. As of November 2024, there are 861 stations in the data. Notably, because historic station capacities are not given and station capacities frequently change due to extension and relocation, we have to estimate when a station is full or empty from the bikes taken in and out. For an exploratory spatial data analysis of Toronto's bike-sharing system, where we also cover other details of the data processing, see our Medium article.

a) *Station Occupancy Estimation:* The behavior at most stations follows a clear daily pattern, where bikes show wave-like patterns, however, the cumulative number of bikes either increases or declines over the month. This is probably due to bikes taken out for repairing or relocation, which are not logged in the ridership data. In order to estimate when a station is empty or full, we thus take daily minima and maxima and consider the station full if the cumulative number of bikes at that timepoint is within 2 bikes close to the maximum or minimum, respectively, as marked in Fig. 2. Notably, we aim to err on the side of overestimation, because if a station is almost full at a time, usually users remember this behavior and refrain from using this station during that time, although the demand exists.

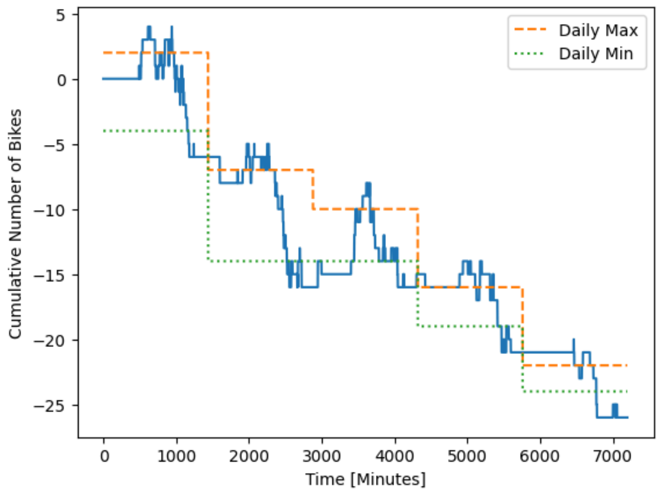

Fig. 2: Cumulative number of bikes in a station over several days with estimated capacity bounds marked.

b) *Temporal Horizon and Rate Calculation:* In modeling bike sharing demand, the prediction time horizon and the averaging horizon fundamental parameters. In [2], the number of pickups in 15 minutes was found to be reliably predictable, which is equivalent to predicting a uniform average over 15 minutes. In our approach, we follow [7] and predict 9 timepoints in 5 minute intervals over the next 45 minutes, given 12 timepoints in the past, for all stations at the same time. Due to our averaging approach, which averages over partial timepoints in the future, we choose the standard evaluation horizon to be 20 minutes, which has in [1] also been found to be the most effective horizon for prediction using standard ML models. Because we are interested in finding a suitable demand function over time, which is related to rate data, a way of averaging the bike pickups has to be chosen. Notably, if one chooses to predict the exact number of bikes taken out or in each minute, one finds almost random behavior, because whether one arrives a minute later or not depends on many other factors, which are insignificant to the demand. Thus, we choose to average with a gaussian filter with $\sigma = 10$ min to calculate rate information. For this standard deviation of 10 minutes, thus roughly 63% of the information accumulated lies in the interval ± 10 min around each datapoint and $\approx 95\%$ in the ± 20 min interval. Empirically, this interval yields nontrivial prediction results. (Notably, smoothing with $\sigma = 60$ min renders the prediction task trivial, yielding similar accuracies for both linear and more complex models.)

c) *Graph Featurization:* For the input graph, we choose to apply an analogous featurization as in the main architecture [7]. We choose to connect two stations, if their distance is lower than a threshold d_{min}, here, we chose a walking distance of $500m$. Additionally, because several stations are farther outside, we choose to connect each station to at least the other N_{min} closest stations. Empirically, we found $N_{\text{min}} = 10$ to improve the prediction performance slightly.

V. Performance Evaluation

For the evaluation of our demand prediction task, we choose to compare three reference models:

- The base STGAT model, with number of nodes and final output size adapted to the problem, without dropout.
- An upscaled and regularized variant, with LSTM sizes $(128, 256)$, dropout of 0.95 and weight decay 0.4.
- A variation, where we replace the LSTMs by a decoder-only transformer with 4 layers, 8 attention heads, and an embedding size of 32.

We split the whole month data into separate days and use approximately 70% of these for training, 15% for validation for hyperparameter optimization and the rest for testing. Notably, the predictive performance of the models depends strongly on the exact split chosen, which is likely due to the limited data investigated, as the testing days might fall on special holidays or other unusual days, where the behavior is significantly different than in the training set. However, this split ensures the model is tested on fully unseen days, as opposed to only unseen segments of these. Also, as in [7], we normalize the data by calculating the Z-score for model input and inverting the transformation for output, i.e. the loss calculated is dimensionless.

In the following, if not noted otherwise, the root mean squared error (RMSE) and mean absolute error (MAE) will always be in $\left[\frac{\text{Bikes}}{\text{Hour}}\right]$, while the mean squared error (MSE) is in $\left[\frac{\text{Bikes}}{\text{Hour}}\right]^2$ and the loss is dimensionless.

A. Predictive Comparison

In order to compare the predictive quality, we compare the RMSE, MSE and MAE on the test set in Table I, as in [7].

TABLE I: Quantitative comparison of the predictive metrics of models evaluated on the test set, along with model size.

Model	Model size	RMSE	MSE	MAE	Loss
Linear	2449.11 MiB	2.28	6.28	1.64	2.60
STGAT	16.45 MiB	1.55	3.07	0.857	0.787
STGAT-upscaled	35.54 MiB	1.55	3.09	0.868	0.812
STGAT-transformer	8.90 MiB	1.57	3.11	0.868	0.8

Notably, the predictive performance of all STGAT models is very similar, with the transformer giving a slightly higher RMSE of 1.57. All models outperform the linear baseline model in all metrics significantly. Also, the specific performance seems to not benefit from upscaling the model, although a larger model would probably be beneficial for a larger data set of multiple years.

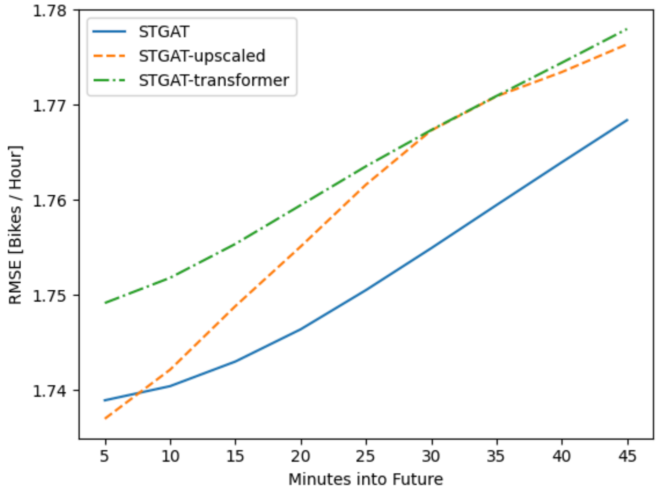

Fig. 3: RMSE for different prediction horizons

In order to see, how the prediction horizon influences our specific models, we compare the RMSEs for each horizon separately, shown in Fig. 3. As before, the absolute error is similar, both over models as well as over the horizon. Both smaller models, the base STGAT and transformer variant share the same nearly linear rise over the prediction horizon. Notably, the upscaled variant seems to learn small horizon prediction better than long-prediction horizon. We hypothesize that this is due to the model overfitting partially to the specific shape of the curve.

B. Demand Evaluation

For the qualitative illustration of the demand extrapolation, we show true rates, and predicted rates and demands for a section of the train data for one station in Fig. 4 for the STGAT base model.

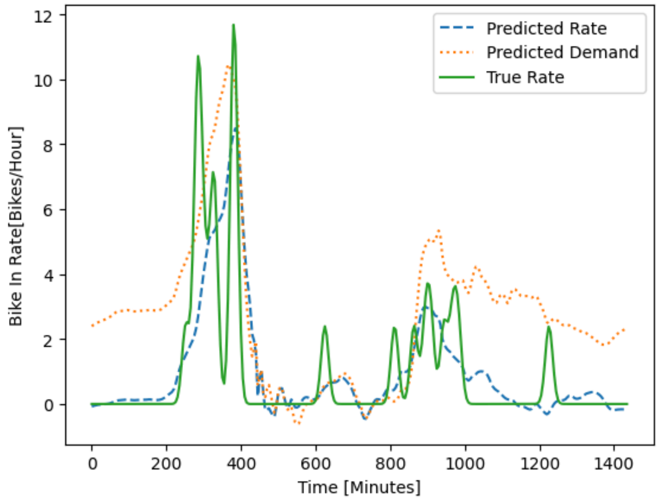

Fig. 4: Example comparison of demand and rate predictions along with ground truth for a horizon prediction of 20min.

We can see that the model's demand predictions generally follow similar shapes as its prediction, but are higher on various points. Qualitatively, we can see several aspects of the model results:

- The model assigns an in-demand of ≈ 2 bikes per hour until ca. 5 in the morning, although the actual rate and prediction is at 0, which is the behavior, we want. Likely, there is demand at night, but the station is full, so it is not matched by actual rate, but can be extrapolated from other nights.
- The demand is generally higher than both the true and predicted rates, which is desired and a fundamental property of the demand.
- The predictions, despite on the training data, do not match the high spikes fully and miss later spikes in the evening. We hypothesize that these are difficult to predict, because they depend on bikes taken out spontaneously, as it seems that the station is full in the evening.

In order to qualitatively compare the characteristics of the demand predictions, we compare a sample over a day of test data in Fig. 5.

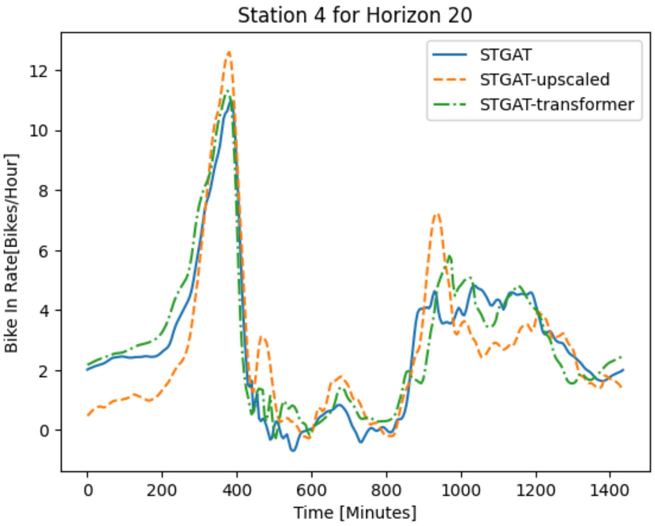

Fig. 5: Comparison of the demand predictions of different models.

We see that, the smaller models, the STGAT and its transformer variation, are close to each other. We hypothesize that the smaller models stick closer to their rate prediction, because the limited model parameters encourage higher sharing of the parameters contributing to demand and rate prediction. Notably the upscaled version seems to give more extreme - both higher and lower - predictions. The upscaled version also seems to produce visually smoother curves, although its numerical smooth violation is slightly higher, which is likely due to its generally higher range estimations.

In order to estimate quantitative measures of the demand prediction, we calculate the individual components of the demand loss function from Section III, shown in Table II. In order to have interpretable metrics, we take the roots of the squared loss components, which are indicated by the 'R' prefix, i.e. the root demand mean square violation is in units $\left[\frac{\text{Bikes}}{\text{Hour}}\right]$. Also we calculate the mean absolute difference of the demand and the rate prediction, to see how far the model extrapolations lie.

TABLE II: Comparison of demand metrics on the test set.

Model	RDemMSViol	RSmoothViol	MADemRateDiff
Linear	2.83	0.0603	4.65
STGAT	1.27	0.00529	1.08
STGAT-upscaled	1.29	0.00706	1.23
STGAT-transformer	1.29	0.00543	1.08

We see analogous results to the previous findings: The STGAT slightly improves the demand rate violation by 0.02 bikes per hour versus upscaled and transformer variants. The linear model shows a significantly higher demand violation, about 10 × higher smoothness violation and aggressively extrapolates demand. Notably, the upscaled version also shows about 10% higher demand extrapolation, which is consistent with Fig. 4.

We have additionally tested several variations of the STGAT, including increasing the number of model heads and number of GAT layers. Usually, the graph featurization is one of the most important design choices in a graph neural network, however, the precise density seems to have only a small impact in this case. We suspect that this is due to the main part of the parameters being in the later layers, which ignore the graph structure.

C. *Discussion of the Models*

We conclude and list the main advantages and disadvantages of the investigated models.

- The linear model has a simple structure, but its model size is about 100 × higher than the base STGAT model. It shows poor model capability and bad performance in all metrics on the test set, rendering it impractical for the precise prediction task at hand. However, with more elaborate feature engineering, as in [8], simple regression models can generally perform well and are more easily interpretable.
- The base STGAT adaptation shows the yields performance in all quantitative metrics, being outperformed only slightly in 5min horizon prediction by the upscaled variant. With a model size of $\approx$ 16MiB, it appears to be robust to overfitting for this particular dataset. However, for larger and more diverse datasets over multiple months, a larger model is likely needed.
- The upscaled STGAT variant shows slightly worse, but comparable performance to the base variant. Notably, its size makes it prone to overfitting and it needs tuned dropout and weight decay regularization. However, we found that it has the capacity to fully fit the training data and thus probably is a good choice for a larger dataset. Notably, we found upscaling the LSTM sizes, rather than the multihead graph attention layers, to be necessary to achieve higher model capability. Also, this model shows visually smoother and potentially less noisy demand curves, which however requires further investigation.
- The transformer variant is slightly outperformed by the STGAT, but has a model size of only half of the parameters. Notably, we found that we could achieve a lower train loss with this model, which suggests that the small deep LSTMs in the base STGAT forget some of the previous information. However, it requires about 20% more iterations to converge.

VI. Conclusions and Recommendations

In our article, we have defined the bike-sharing demand prediction problem and proposed axioms for a natural demand function.We have translated the demand axioms into a continuous regularized loss function for training neural network models. We then adapted the STGAT architecture from [7], to jointly predict rates and demand and evaluated qualitative aspects of the resulting demand prediction and the quantitive quality of the predictions. We have seen:

- Several variants of the STGAT model yield sufficiently accurate results for rate prediction.
- The demand functions learned using our regularized loss are smooth and extrapolate the data in a natural way.

For future investigation, we find that several aspects are worth further work:

- Developing a short term adaptive pricing model, that incentivizes and shows routes to users from nearby stations with high in demand to stations with high out demand, given their target and start point seems to be the most natural application of our model.
- Training the model on multiple months of data and increasing the prediction quality seems to be a central problem, as the accuracy of the demand naturally correlates with prediction capability of the model.
- Integrating other features such as weather, wind, or other relevant features from [10] into the demand prediction model is also likely to be a promising improvement.
- Investigating successful graph-based models from other domains in their role for demand prediction is also likely to improve accuracy and the generalized demand metric.
- Finally, exploring spatial differences of the demand prediction to the true rates is promising for investigating the most problematic times and stations in the city in order improve the relocation of bikes and stations.

VII. Code

We provide the full implementation of our solution and evaluation in our github repository:

- Github permanent link to the main repository's readme: Link
- A Google colab notebook that allows running and evaluating the main STGAT model: Link

References

[1] H. I. Ashqar, M. Elhenawy, M. H. Almannaa, A. Ghanem, H. A. Rakha, and L. House, “Modeling bike availability in a bike-sharing system using machine learning,” in *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 2017, pp. 374–378. doi: 10.1109/MTITS.2017.8005700.

[2] A. Mehdizadeh Dastjerdi and C. Morency, “Bike-Sharing Demand Prediction at Community Level under COVID-19 Using Deep Learning,” *Sensors*, vol. 22, no. 3, 2022, doi: 10.3390/s22031060.

[3] Y. Yang, A. Heppenstall, A. Turner, and A. Comber, “Using graph structural information about flows to enhance short-term demand prediction in bike-sharing systems,” *Computers, Environment and Urban Systems*, vol. 83, p. 101521, 2020, doi: https://doi.org/10.1016/j.compenvurbsys.2020.101521.

[4] Y. Liang, G. Huang, and Z. Zhao, “Cross-Mode Knowledge Adaptation for Bike Sharing Demand Prediction Using Domain-Adversarial Graph Neural Networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 5, pp. 3642–3653, 2024, doi: 10.1109/TITS.2023.3322717.

[5] J. Hanna, “Bike share toronto 2023 Business Review.” [Online]. Available: https://www.toronto.ca/legdocs/mmis/2023/pa/bgrd/backgroundfile-240804.pdf

[6] S. Zhang, Y. Guo, P. Zhao, C. Zheng, and X. Chen, “A Graph-Based Temporal Attention Framework for Multi-Sensor Traffic Flow Forecasting,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 7743–7758, 2022, [Online]. Available: https://api.semanticscholar.org/CorpusID:234906657

[7] X. Kong, W. Xing, X. Wei, P. Bao, J. Zhang, and W. Lu, “STGAT: Spatial-Temporal Graph Attention Networks for Traffic Flow Forecasting,” *IEEE Access*, vol. 8, no. , pp. 134363–134372, 2020, doi: 10.1109/ACCESS.2020.3011186.

[8] X. Wang, Z. Cheng, M. Trépanier, and L. Sun, “Modeling bike-sharing demand using a regression model with spatially varying coefficients,” *Journal of Transport Geography*, vol. 93, p. 103059, 2021, doi: https://doi.org/10.1016/j.jtrangeo.2021.103059.

[9] T. C. Julie Wang Amelia Woodward, “Predicting Los Angeles Traffic with Graph Neural Networks.” [Online]. Available: https://medium.com/stanford-cs224w/predicting-los-angeles-traffic-with-graph-neural-networks-52652bc643b1

[10] E. Eren and V. E. Uz, “A review on bike-sharing: The factors affecting bike-sharing demand,” *Sustainable Cities and Society*, vol. 54, p. 101882, 2020, doi: https://doi.org/10.1016/j.scs.2019.101882.