GRAPH NEURAL NETWORK-BASED DESIGN DECISION SUPPORT FOR SHARED MOBILITY SYSTEMS

Yinshuang Xiao
Walker Dept. of Mechanical Engineering
The University of Texas at Austin
Austin, Texas 78712-1591
Email: yinshuangxiao@utexas.edu

Faez Ahmed
Dept. of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
Email: faez@mit.edu

Zhenghui Sha*
Walker Dept. of Mechanical Engineering
The University of Texas at Austin
Austin, Texas 78712-1591
Email: zsha@austin.utexas.edu

Emerging shared mobility systems are gaining popularity due to their significant economic and environmental benefits. In this paper, we present a network-based approach to predicting travel demand between stations (e.g., whether two stations have sufficient trips to form a strong connection) in shared mobility systems to support system design decisions. In particular, we answer the research question of whether local network information (e.g., the network neighboring station's features of a station and its surrounding points of interest (POI), such as banks, schools, etc.) would influence the formation of a strong connection or not. If so, to what extent do such factors play a role? To answer this question, we propose using graph neural networks (GNNs), in which the concept of network embedding can capture and quantify the effect of local network structures. We compare the results with a regular artificial neural network (ANN) model that is agnostic to neighborhood information. This study is demonstrated using a real-world bike sharing system, the Divvy Bike in Chicago. We observe that the GNN prediction gains up to 8% higher performance than the ANN model. Our findings show that local network information is vital in the structure of a sharing mobility network, and the results generalize even when the network structure and density change significantly. With the GNN model, we show how it supports two crucial design decisions in bike sharing systems, i.e., where new stations should be added and how much capacity a station should have.

***Keywords**: Shared mobility systems; Socio-technical systems; Complex networks; Artificial neural network; Graph neural network.*

*Corresponding author.

1 Background and Introduction

The shared mobility system is a typical complex socio-technical system in that its functionality and complexity are closely related to human and social behaviors [1]. This emerging system has experienced rapid growth in the last decade due to its sustainable and environmentally friendly characteristics. Another major reason for the shared mobility system's popularity is its importance in last-mile transportation, making it appealing in congested urban areas. The growth of shared mobility systems opens up new opportunities for new modes of transportation but also poses challenges to the design and operation of such human-centered systems. For example, a common problem suffered by shared mobility systems is the rebalancing issue, i.e., bikes are delivered by users between stations, resulting in some stations being overcrowded and some being vacant. This issue is mainly due to imbalanced demands for points of interest at different locations [2] or suboptimal system design decisions, for example, imbalanced dock distribution (the distinct dock differences between stations within identified vital local service systems) in bike sharing systems (BSS) [3]. Effective design methodologies and solutions to these problems are essential for the success of system operation, a high customer retention rate, and long-term quality service.

To this end, in the existing literature, attempts have been made to develop vehicle-based and user-oriented rebalancing strategies [4–7]. Some other studies focus on system infrastructure design decisions, e.g., station location and capacity planning [3, 8, 9]. For example, in our previous study, a network-based design approach was proposed to balance the capacity difference between stations in the local service systems of a BSS network for enhanced robustness against

seasonal effects [3]. However, one challenge associated with these studies is that the design decisions found are difficult to validate. To address this challenge, it is necessary to have a dedicated and highly performed predictive model that can forecast travel demand in response to design decisions and strategies before they are actually implemented in practice, as shown in Figure 1. Additionally, a powerful predictive model will be beneficial to system design in terms of cost control, robust operation, and maintenance.

The rest of the paper is organized as follows. In Section 2, we introduce the reference framework for existing predictive models of shared mobility systems and especially the GNN models. Then, Section 3 illustrates the problem formulation, as well as a summary of our contributions. The proposed complex network-based approach and the associated methods for model analysis and evaluation are presented in Section 4. Section 5 takes the Divvy Bike in Chicago as a case study to demonstrate our approach. In Section 6, we discuss the limitations of the proposed model and clarify its utility constraints on the support of systems design. Finally, the paper is concluded in Section 7 with future work and closing thoughts.

2 Frame of Reference

2.1 Predictive models of shared mobility systems

The research on shared mobility prediction did not gain attention until the worldwide promotion of the third-generation shared mobility program in 2006 [10, 11]. In the early studies, researchers often employ simple statistic models to study the system dynamics and predict the available bikes at a station or the demand for usage [12–15]. For example, Froehlich et al. [13] proposed four predictive models, namely Last Value (LV), Historic Mean (HM), Historic Trend (HT), and Bayesian Network (BN), to forecast the availability of bikes at each station within a time frame ranging from 10 to 120 minutes. The parameters considered for all these models include the current time point, the current known number of bicycles, and the prediction window. In a separate study, Borgnat et al. [12] adopted a linear regression model to predict the number of bikes rented in an hour, incorporating weather, holiday effects, the number of subscribers, and the total number of bikes as input factors. These predictive models represent initial explorations of shared mobility prediction, but their scope is limited to capturing simple interactions, such as linear relationships between impact factors and user demand.

Classical machine learning models The recent advancement in machine learning (ML) literature has greatly promoted research on various predictive models of user travel patterns and helped to gain better insights into shared mobility. Compared to early studies, more factors influencing user travel behaviors are explored, including peak hours, surrounding Point of Interest (POI), spatial dependencies between serving stations, etc. [16–19]. The ML models can be put into two categories: classical ML models and deep neural network models [16]. Some popular classical ML models include different variants of the linear regression model [20–22], Bayesian models [23], and the Markov queuing model [24]. These models are advantageous over simple statistical models in that they improve interpretability. For example, in [22], the authors develop a log-linear mixed model to understand how factors, such as bicycle infrastructure attributes and land use characteristics, influence bicycle arrival and departure rates. The estimation results show that the numbers of arrivals and departures in a sub-city district positively correlate with the station density. However, the downside of classical models is that they are usually built with many assumptions, resulting in low model validity and predictive performance.

Deep neural network models Deep neural network (DNN) models typically outperform classical ML models in terms of predictive power but at the expense of losing interpretability. As summarized in a recent survey study [25], those advanced DNN models for shared mobility systems are variants of four classical DNN models, feedforward neural network (FNN)[1], recurrent neural network (RNN), convolutional neural network (CNN), and graph neural network (GNN). FNN is the simplest neural network where information flows in a forward direction, starting from the input nodes, passing through any hidden nodes, and ending at the output nodes, without any cycles or loops [26]. A representative study suggested a variation of FNN, known as the "pseudo-double hidden layer FNN," to forecast the demand for bike rentals [27]. RNN with nodes connected by feedback loops, enabling neural networks to exhibit temporal dynamic behavior, is frequently used to capture the long-term temporal dependency of shared mobility systems [28–30]. In the work of [30], a dual attention-based RNN is proposed to predict the demand for bike-sharing over the next 10 minutes by taking the usage data in the past 20, 30, 40, and 60 minutes as input, respectively. This model uses random walks to preserve relationships between bike stations in time-series data preprocessing, increasing adaptability to local changes. It also incorporates an attention mechanism to extract spatial and temporal features. The inherent ability of CNN to learn from multiple inputs and extract features provides an efficient way to handle the temporal dynamics and spatial dependencies of a shared mobility system and herein predicts demand patterns in a shared mobility system [31, 32]. For example, in [31], the authors adopt CNN to predict daily bicycle pickups at both city and station levels, and improved predictive performance is achieved at both levels compared to the simple ANN and autoregressive integrated moving average (ARIMA) time series models.

While FNN, RNN, and CNN have proven to be effective in solving certain problems in shared mobility systems, such as demand prediction [28, 30] and spatial interaction analysis between stations [17, 18], they have limitations. For example, FNN lacks the ability to capture complex interactions between stations; RNN is prone to gradient vanishing

[1] In this study we also refer to it as the simple/regular artificial neural network or ANN in short.

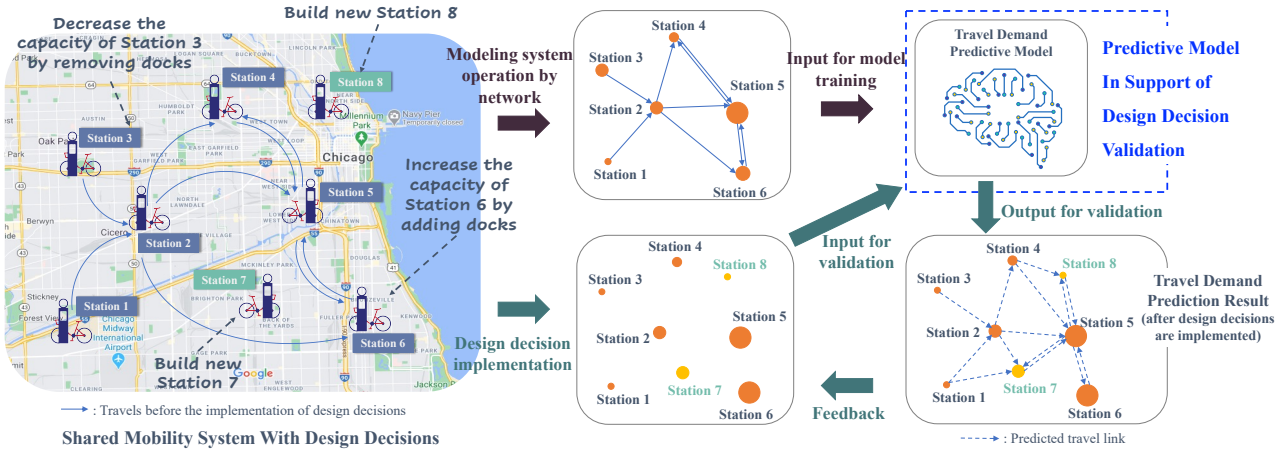


Fig. 1: Illustration of the network-based design decision support framework for the shared mobility system. In addition to supporting design decision validation, the highlighted predictive model in the blue dashed box can also: 1) be a surrogate model of a real-world experiment, 2) provide instant feedback for future design decision adjustment, and 3) assist in decision-making on system operation.

and gradient explosion problems; and CNN has limited ability to process graph-structured data [25]. To address these limitations, GNN, which is specifically designed to handle graph-structured data, was evident to be the solution.

2.2 Graph neural network (GNN) model

Graphs are an important representation for complex systems [33–35] in that they not only model the interconnection and interrelation between system elements but also the leverage of complex network theories [36]. Graphs are non-Euclidean data, as opposed to other regular Euclidean data, such as images (2D grids) and texts (1D sequences). Its high dimensionality hinders the direct usage of some advanced neural network models such as CNN. To fill this gap, a Graph Neural Network (GNN) [37] was proposed in 2008, and due to its outstanding performance, GNN has been widely used across domains since then [38, 39]. For example, Ahmed *et al.* [40] developed a GNN-based method to predict the competition relationships between different car models in a vehicle co-consideration network. The model provided great insight into the key engineering attributes that promote the formation of car competitions.

In transportation research, some representative studies include using GNN to forecast spatially heterogeneous traffic speed within the road network [41] and a novel Conv-GCN model combining a graph convolutional network (GCN) and a three-dimensional CNN for the prediction of short-term subway passenger flow [42]. Recently, a large number of GNN models have also been introduced for the study of shared mobility systems [17, 18, 43]. For example, in [17], the authors present a graph convolutional neural network (GCNN) based approach, which incorporates long short-term memory (LSTM) layers, to predict hourly demand in a large-scale bike sharing network. Meanwhile, they demonstrate the effectiveness of the proposed model considering both spatial and temporal dependencies between bike-sharing stations.

The fundamental idea of GNN is that each node within a network is defined by its features and network neighbors, so each node in a network can be represented by these two pieces of information. Such a representation is also referred to as node embedding. Following the acquisition of node representation, various downstream tasks, such as node/link/graph classification, node/link/graph regression, node clustering, link prediction, and graph match, can be accomplished [44]. Recently, many variants of GNN have been developed, each based on a different node embedding strategy [45–47]. For example, the well-known DeepWalk algorithm [46] generates node embedding in two steps, the first of which is to perform random walks on nodes in a graph to obtain node sequences. The skip-gram is then used in the second step to learn the node embeddings from the generated sequences [48].

GraphSAGE is another remarkable variant of GNN in that it is a general inductive framework. Unlike other frameworks that train individual embeddings for each node, GraphSAGE learns an embedding generating function by sampling and aggregating features from a node's neighborhood [45]. This inductive framework provides a solution for graphs with varying node counts. Even if an unseen node is introduced into the graph, its representation can still be properly generated by feeding its neighborhood feature into the trained embedding generating function. This is also the primary reason for us to choose GraphSAGE in this study to learn node embeddings of shared mobility networks. For example, in a BSS, the system expands or compresses its scale by introducing new stations or discarding old stations. A more detailed description of the algorithm can be found in [45].

After reviewing the current literature, we found that the existing GNN models only predict the rental and return demands at the station level, but do not inform where the return comes from and the rental goes [19, 27, 29]. Furthermore, most studies are keen on predicting short-term travel

demand (e.g., the number of bikes used in an hour) [17, 32], which are typically more advantageous for the analysis of dynamic system operation but do not predict the number of trips occurring from one station to another over a longer period, such as one month or one year. From the engineering system design point of view, the prediction of long-term relationships between stations is important because it provides a summative view of a system's connections, enabling the detection of potential design defects, without being affected by the fluctuations in rental and return demand. For instance, a station may experience fluctuations in its number of in- and out-connections in different short periods, resulting in potential rental issues (no bikes to rent) in the morning and return issues (no dock to return) in the afternoon. These short-term demand issues can be addressed through dynamic rebalancing strategies. However, if a station or the entire system has imbalanced in- and out-connections during a longer period of time, it is necessary to address this issue through optimal design decisions to change the current system architecture or structure, such as expanding existing stations by adding more docks or constructing new stations to ease the load at popular sites.

In our previous work [3, 49], we have demonstrated that a network is an effective representation and tool for studying shared mobility systems and have identified important local network structural patterns (called network motifs) that contribute to the formation of real-world BSS networks. The advantages of taking a network perspective in the research of shared mobility systems are twofold. First, as illustrated in Figure 1, it transforms the mobility system demand prediction to the network link prediction, thus providing a means to assess the relations between the origin and destination of each trip. Design decisions can also be easily incorporated into the network model, such as introducing the newly built stations as new nodes and enlarging the node size to reflect a station's expansion. Second, network models can provide means to investigate the interactions between local system structures (e.g., travel patterns among three stations in a local area) and global system performance (e.g., network robustness), which is essential to answer the following **research question**: whether and to what extent local network information (e.g., structure and node features) plays a role in the formation of shared mobility networks.

Therefore, in this study, our **research objective** is to develop a complex network-based predictive model considering neighborhood information to predict travel demand between stations in shared mobility systems in the long term to support systems design decisions. Travel demand refers to the existence of at least a specific number of trips that occur from one station to another over a period of time.

3 Problem Formulation

This study focuses only on docked shared mobility systems, i.e., the system includes fixed service stations with limited docking capacity. Our goal is to develop a predictive model and test its applicability to aid in design decision validation, such as whether expanding some popular stations can meet the rising travel demands. The essence of the predictive model is to predict the existence of a connection (with a determined number of trips to define the strength of the connection) between any two stations in the next year based on the previous year's trip data. Regarding the format of the trip data, to avoid seasonal effects, we split the yearly trip data into twelve months. Therefore, the problem is changed to predict the existence of the travel demand from one station to another in month i ($i = 1, ..., 12$) of next year with the trip data of month i in the previous year.

As shown in Figure 1, a network is used to model travel demand within a month in a BSS, where nodes represent the stations, and node attributes indicate station information, such as geographic coordinates, capacity, and the number of surrounding POIs. Since we are also interested in the influence of the link strength on the prediction accuracy, we define a series of link cutoffs where only if a link with the number of trips exceeds the defined cutoffs will be kept in the network. Therefore, the network under investigation is a directed unweighted network. We chose not to study a weighted network because a shared mobility system involves many uncertainties associated with social activities, resulting in the challenge for a predictive model to directly predict its link weight (i.e., how many trips would occur from one station to another) with meaningful accuracy. Thus, we arrived at forecasting the link weight in two stages, first predicting the link's existence and then forecasting the weight thereafter. Furthermore, a high-performance unweighted link prediction model is preferred to serve as a substitute model for crucial downstream analyses, such as the analysis of connections across the entire shared mobility system, compared to a low-performance weighted network model. With this network setup, the original prediction problem is transferred to a directed binary link prediction problem. This is a major difference between this study and existing work, where links are often undirected. Directed network data can cause more severe imbalanced data issues (e.g., a significantly large number of negative cases vs. a small number of positive cases), which poses significant challenges for machine learning algorithms to improve prediction accuracy.

In this paper, we adopt the graph neural network (GNN) model [40,44] based on GraphSAGE for its ability to capture and quantify the effect of local network structures through network embedding – node representation by sampling and aggregating features from its network neighborhood [45]. More details of GNN are given in Section 2.2. The contributions of this study can be summarized as follows:

1. We proposed a complex network-based approach based on GNN to predict travel demand between stations in shared mobility systems. By comparing to regular Artificial Neural Network (ANN) models, we showed that when two-hop neighbors' information of a station is included, the model's prediction performance is 8% higher than that without neighbors' information.
2. We tested the performance of the proposed predictive model when the link strength increases from weak to strong. The results show that the proposed predictive

model with network neighborhood information always outperforms the ANN model without neighborhood information, regardless of network density and typology. Moreover, we discover that as the network becomes sparser, the predictive performance (i.e., the F1-Score and the PR AUC) of both models decreases logarithmically.

3. The proposed approach creatively connects design decisions in shared mobility systems (e.g., where to add a station) with the link prediction problem in networks, thus providing a tool for system designers to test and experiment with their design strategies. This is particularly important for complex systems research because the verification and validation of the design of such systems have always been a significant challenge.

We published our preliminary results in the Proceedings of the 2022 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference [50]. In this paper, we made significant improvements on top of our prior study, and the main distinctions are: 1) this paper updates the model training process by employing Bayesian optimization for hyperparameter tuning and K-fold cross-validation, resulting in more reliable prediction results; 2) we deleted the approximation methods for GNN-based prediction. In the current work, we explore the model performance when link strengths are varied, which provides more insights into the generalizability of the proposed approach; 3) the content of Section 5.4 is newly added to show how the proposed predictive model can be utilized to support system design decisions.

4 Methodology

An overview of the complex network-based approach to predicting travel demand for shared mobility systems is shown in Figure 2. In this approach, we start by modeling a shared mobility system as a complex network using historical data, i.e., Period One data, including station attributes and trip data. After obtaining the network model, we utilize Period One data to train predictive models, including the ANN model in Section 4.2 and the GraphSAGE-based model in Section 4.3. The trained model is then employed to predict the network links in Period Two based on the updated nodal attributes. To evaluate the predictive performance of the models, the predicted links are compared with the actual ones, and the metrics quantifying the prediction accuracy are introduced in Section 4.4.

4.1 Node attributes

In this study, the node represents the bike sharing station and the node attributes indicate the station features. The node attributes considered here include the geographic coordinates of the station, the number of docks, and the POIs surrounding the stations. Geographic coordinates can be used to calculate the distance between two stations, and the number of docks at a station determines the maximum number of bikes that users can rent from or return to that station. Current research indicates that there are evident travel patterns between certain functional zones of the city due to user-specific travel purposes [18, 19]. In the study [18], for example, He and Shin divided POI into five major categories, including residential, cultural, recreational, commercial, and governmental. They found that travel behavior in BSS has a stronger correlation between stations in recreational and residential areas than between stations in recreational and commercial areas.

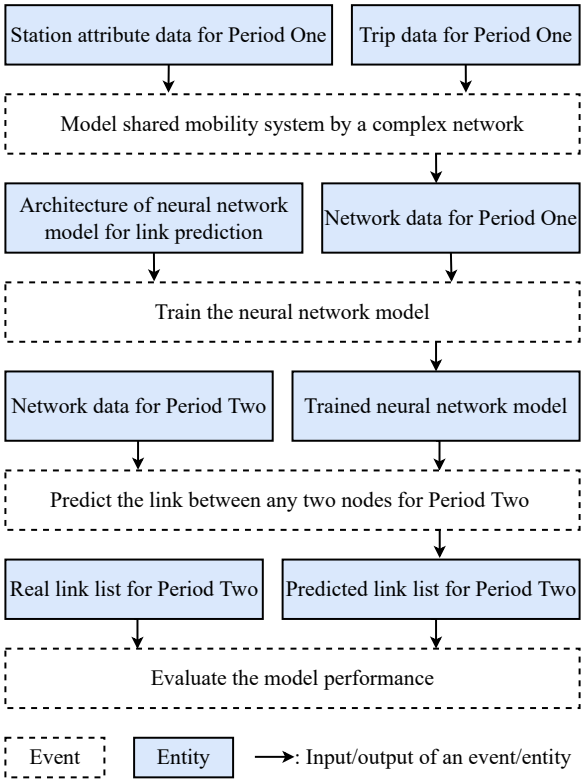


Fig. 2: Complex network-based prediction framework for shared mobility systems design support with Neural Network (**Period One**: Month i in year Y, **Period Two**: Month i in year $Y+1$, $i = 1, ...12$)

In this study, POI data are collected by Overpass turbo [51] which includes the name of each POI and its geographic coordinates. We first classify POIs into seven categories, including Financial, Education, Recreational&Tourism, Residential, Sustenance, Healthcare, and Transportation. The details of these categories are given in [50]. Then, we draw a circle of radius R with the target station in the center of the circle. Finally, we count the number of POIs in each category within the circle and treat the combination of seven counts as an attribute vector of the target station. Regarding the value of the radius, R, in reference [52], the authors calculated the cumulative percentage distribution of walking trips by distance based on data from the 2009 U.S. National Household Travel Survey. We learned from the distribution that 1.5 miles are the walking distance upper bound of 90% of walking trips. Therefore, POIs within 1.5 miles of a station provide the best representation of the station's surroundings. Taking Divvy Bike station 368 (Ashland Ave & Archer Ave) in Chicago as an example, its POI attribute vector in 2016 is $[2, 30, 0, 2, 4, 12, 12]$, indi-

cating that there are two banks, thirty education institutions, two healthcare institutions, four apartments, twelve restaurants, and twelve public transportation stops within a radius of 1.5 miles.

4.2 Baseline: ANN-based link prediction model

In this study, we take ANN as the baseline model. As shown in Figure 3, the architecture of a simple ANN model consists of an input layer, one hidden layer, and an output layer. Training a model starts by formulating link features. In a shared mobility network, the link features are determined by the two connecting nodes. Accordingly, we use the concatenation of the features of the start and end nodes with size N to represent the features of the directed link with size $2N$. To improve training stability, max-min normalization is adopted to transform different features into a similar scale. Then, the normalized features are connected to the input neurons in a one-to-one manner. The hidden layer embedded between the input and output layers is fully connected to these two layers and is the same size as the input layer. ReLU is used as the activation function for each neuron in the hidden layer. The activation function of the output neuron is a sigmoid function to determine whether there is a link from one node to another or not. This is a supervised learning model that learns how to map the input to the output, i.e., the link features to the link label. The stochastic gradient descent (SGD) algorithm is used throughout the training process to minimize binary cross-entropy loss. After obtaining the trained model, the updated link features for the following year are fed into the model to predict its network topology.

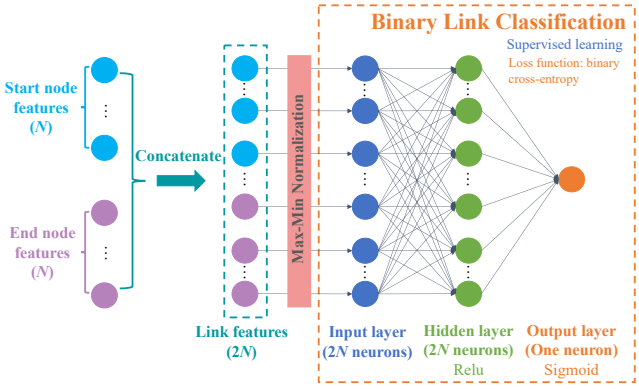


Fig. 3: Architecture of the ANN model for link prediction.

4.3 The GNN-based link prediction model

4.3.1 Model architecture

As illustrated in Figure 4, a GraphSAGE link prediction model comprises two major parts: node embedding and link prediction. The node embedding is to learn a representation for each node in a vector of size M. Given a central node and its two-hop neighborhood, we first randomly sample its direct in- and out-neighbors at the first hop. Then, the same procedure is repeated to the sampled hop-1 neighbors to get their hop-1 in- and out-neighbors, i.e., hop-2 neighbors of the central node. After that, the node features of hop-2 neighbors are normalized by max-min normalization and used as the representations of hop-1 neighbors. Lastly, the node embedding of the central node can be obtained by tracing from hop-2 neighbors and aggregating their embeddings to hop-1 nodes and then to the central nodes inversely. The aggregator used in this study is the mean aggregator, where the node embeddings are computed by averaging neighboring node features [45].

Similarly to the ANN model, the learned node embeddings of the start node and end node are concatenated to represent the link embedding with size $2M$. Because the data have already been normalized during embedding, the link embedding is connected directly to the input layer, which is followed by one hidden layer and one output layer. The size of the input and hidden layers is the same as that of the link embedding. Other settings are identical to the ANN model for a fair comparison. In contrast to the ANN model that uses node features as input, node embedding learns information about a node's neighbors in addition to its own features in the network.

4.3.2 Model training and evaluation

During the training process, two types of data are fed into the model. One is the network data including node features and network adjacency matrix. The other one is the labels of all candidate links in the network, where existing links are labeled as class 1 and non-existing ones as class 0. The network data for GraphSAGE is to learn node embeddings, while the label data is for the learning task in link classification. This entire procedure is an end-to-end training to minimize the binary cross-entropy loss function by SGD [40].

When using testing data from the next year to evaluate the trained model's predictive performance, our input consists of the network data, including the updated node list and node features as well as the approximate network adjacency matrix. This approximate adjacency matrix is critical to have a correct link prediction by better estimating the embedding of a node in the future year.

4.3.3 Methods for adjacency matrix approximation

GraphSAGE assumes that if an embedding generating function of one type of network is learned, it can be used to generate node embedding by the same type of network. The assumption is that the training and testing networks should be of the same domain and have similar characteristics. It should be noted that since the new nodes (without any neighborhood information) do not have neighbors, GraphSAGE cannot use the adjacency information to make predictions for these nodes. In this study, since the shared mobility networks for training and testing are of the same month but in different years, they share similar characteristics. However, the challenge is that the test network for a future year is unknown. To obtain the embedding of the testing nodes, an approximate adjacency matrix must be obtained to estimate their neighbors.

According to the study [53], there are several ap-

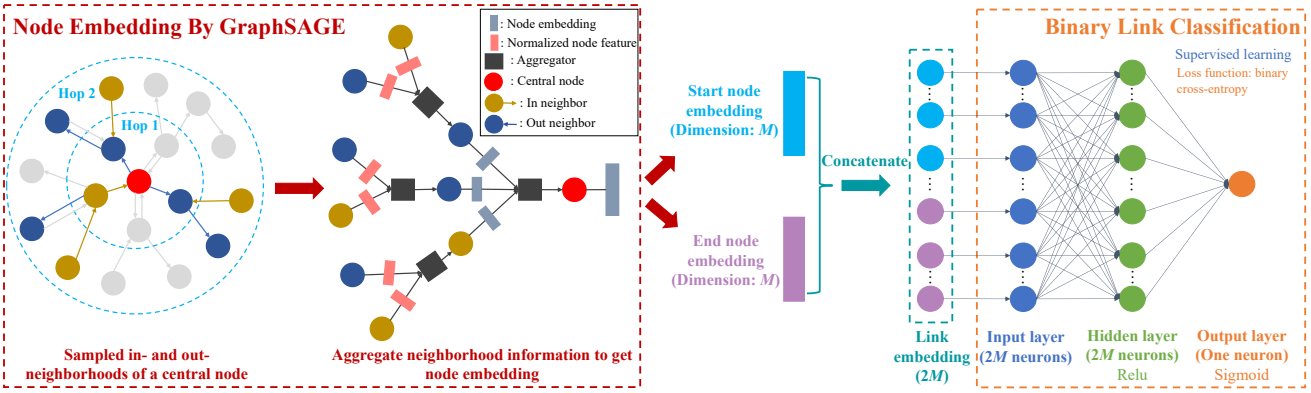


Fig. 4: Architecture of the GraphSAGE model for link prediction.

proaches to approximating the adjacency matrix, including directly using the training network or building a separate machine learning model for such an approximation. In our previous work [50], we tested three methods.

1) The first method uses the adjacency matrix of the Period Two network obtained from the ANN model as input for the node embedding generation.
2) The second method employs a modified Period One mobility network. In this method, for those stations retained from Period One work, their neighbors are copied directly into the Period Two adjacency matrix. For the stations removed from Period One network, thus do not present in Period Two network, they are ignored. For those stations newly introduced in the Period Two network, they are kept independent and no neighborhood information is included in the embedding.
3) Finally, we use the real Period Two network to learn the node embedding and take its prediction performance as the ground truth to compare with the other two approximation methods.

Based on the comparison results, we found that adopting the modified Period One mobility network to learn the node embedding of Period Two can generate the best prediction (the area under the precision-recall curve of the second method is much closer to the ground truth and exhibits 6-10% improvements compared to the first method). The adjacency matrix approximation by modifying the Period One network is thereby followed in this study.

4.4 Link Prediction Evaluation

Since link prediction in this study can be considered equivalent to binary classification, common metrics for binary classification can be adopted, including the confusion matrix, F1-Score, receiver operating characteristic (ROC) curve, precision-recall (PR) curve, and the area under the ROC and PR curves (a.k.a. AUC, area under curves). ROC AUC has a value between 0.5 (no skill) and 1.0 (perfect prediction); while PR AUC has a value between k (no skill) and 1.0 (perfect prediction), where k is the area under the no-skill PR curve, equal to the ratio of minority examples (class 1 links in our case) in the dataset. A higher AUC value indicates better predictive performance. For imbalanced classification problems where the majority of observations are negative cases and the minority of observations are positive cases, ROC analysis provides equal insights on the model's predictive performance in both cases. PR analysis focuses more on the model's ability to predict the minority case, i.e., the positive links in the networks under current investigation [54].

5 Case Study

In this section, we take Divvy Bike in Chicago as an example to demonstrate the utility of the proposed GNN-based models for shared mobility networks. In Section 5.2, the GraphSAGE link prediction model is compared with the ANN model to test whether local network information (i.e., node embedding features) impacts link prediction. In Section 5.3, we verify the generalizability of the proposed models by setting different link cutoffs. Finally, in Section 5.4, a system design case is formulated to illustrate how predictive models can support system design decisions. Here, the design decisions denote determining where to place new stations, how many docks to add or remove from existing stations, etc.

5.1 Data Source

The Divvy Bike data are available to the public [55], and the data for May 2016, referred to as Period One data, and May 2017, referred to as Period Two data, are used in this study. The data package contains both station and trip data. The station data includes the ID, name, geographic coordinates, number of docks, and online date for each station. The trip data recorded each trip's start and end station IDs, trip time and duration, and users' basic information (e.g., gender and birth year). We follow the approach described in our previous work [3] to process the data and build binary-directed trip networks by removing the links with less frequent trips (that is, those that occurred no more than four times in a month). Taking the Period One network as an example, a visualization of this binary-directed network is shown in Figure 5. The top hub stations' information for both two periods is listed in Table 1. The top five hub stations in Period One are observed to be the hubs in Period Two despite a slight change in ranking. In terms of POI information, a total of 2,269 POIs in Period One and 2,403 POIs in Period Two are

Table 1: Top five hub stations information in the trip networks of Period One (May 2016) and Period Two (May 2017)

Period One			Period Two		
Station ID	**Station Name**	**# of Connections**	**Station ID**	**Station Name**	**# of Connections**
287	Franklin St & Monroe St	320	77	Clinton St & Madison St	326
268	Lake Shore Dr & North Blvd	319	287	Franklin St & Monroe St	307
35	Streeter Dr & Grand Ave	317	35	Streeter Dr & Grand Ave	302
77	Clinton St & Madison St	316	91	Clinton St & Washington Blvd	295
91	Clinton St & Washington Blvd	303	268	Lake Shore Dr & North Blvd	295

collected based on the method presented in Section 4.1. According to Section 4.1, each station has geographic coordinates (two features), the number of docks (one feature), and POIs (seven features), for a total of ten features.

5.2 GraphSAGE-Based Link Prediction

5.2.1 Data preparation for ANN-based link prediction

In the ANN model, each candidate link within a trip network, represented by a pair of nodes, is a data sample. Consequently, there are 285,690 data samples in the Period One network, with 21,221 links classified as class 1 and 264,469 links as class 0. To reduce variability, the K-fold cross-validation approach [56] is applied, where K is set to 5. Therefore, we evenly split all class 1 links into five folds, i.e., 20% data for each fold. Meanwhile, to avoid imbalanced training, the same number of class 0 links is randomly drawn without repetition from the class 0 sample pool and added to each fold. With these treatments, there are around 4,244 class 1 samples and 4,244 class 0 samples in one fold. Then, the cross-validation process alternately retains the first to the fifth fold for validation and the remaining four folds for training. In terms of the testing dataset, given that Divvy Bike had 582 stations during Period Two (May 2017), the total number of potential links from each pair of nodes in the testing dataset is 338,142. The final result is reported by averaging the K prediction results.

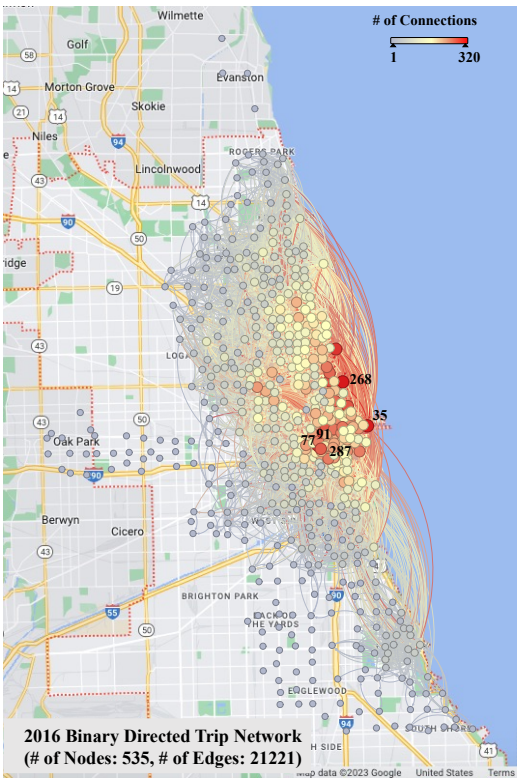


Fig. 5: A visualization of the Divvy Bike trip network in May 2016. The nodes represent docked bike stations, and the directed links are trips that occur from one station to another with a frequency of more than 3 times in a month.

5.2.2 Data preparation for GraphSAGE predictive model

In addition to the data discussed above, the GraphSAGE model also requires network data to learn the node embeddings. For the training model of node embeddings, we feed it with the entire Period One network. As stated in Section 4.3.3, due to the fact that the Period Two network is unknown from a prediction point of view, we take a modified Period One network as input for the node embedding of the Period Two prediction. For those stations that are no longer operated in Period Two, they are removed and 48 new stations are added as isolated nodes. To ensure a fair comparison, we stick to the same configuration in the second stage for training the link classification model as was adopted in the ANN model.

5.2.3 Experiment settings

We first employ Bayesian optimization [57] to perform hyperparameter tuning. The parameters that need to be optimized and their tuning ranges are listed in Table 2. In addition, we specify the objective of Bayesian optimization to minimize validation loss and set the training stopping criterion as 'no improvement in 10 epochs.' The number of tuning iterations is set to 15, the first five of which are random explorations. To reduce computational expenses, only the fold-one data is used to probe the best combination of hyperparameter values. The remaining four folds of the data are trained by following the same parameter settings.

Table 2: Hyperparameter tuning settings

Setting Items	Tuning Range
Minibatch size	[32, 240]
Learning rate	[1e-4, 1e-3]
Number of sampled in- and out-neighbors in two hops[1]	[5, 50]
Node embedding size	[10, 50]

[1]: Note that the numbers of sampled in-neighbors and out-neighbors can be different, we set them the same to simplify the model.

The results of hyperparameter tuning, as well as other hyperparameter values, are summarized in Table 3. Note that the epoch number of the optimized tuning results is 103. We extended it to 150 epochs for each training to further ensure the reliability of the training process. Finally, we ran experiments on single a machine with one NVIDIA P2200 GPU (5GB of RAM at 10Gbps speed), one 11th Gen Intel Core CPU (i9-11900 2.50GHz), and 32GB of RAM.

Table 3: Experiment parameter settings

Setting Items	Model Applied	Value
Neighborhood search depth	GraphSAGE	2
# of Sampled in- and out-neighbors in two hops		26
Node embedding size		26
Input and hidden layer size for GraphSAGE		52
Input and hidden layer size for ANN	ANN	20
Minibatch size	GraphSAGE and ANN	116
Epoch		150
Learning rate		3.49e-4
Dropout		0

5.2.4 Results for GraphSAGE-based link prediction

We first assess the performance of these two models using the confusion matrix and F1-Score, as shown in Table 4[2]. The left-hand side shows the confusion matrix and the F1-Score of the ANN model. The confusion matrix includes four different combinations of predicted and actual classes, where there are 271,108 $\pm$ 1,830 true negatives, 47,177 $\pm$ 1,830 false positives, 914 $\pm$ 89 false negatives, and 18,943 $\pm$ 89 true positives. The true negative rate (TNR) and the false positive rate (FPR) reveal that 85.18% $\pm$ 0.58% of links in class 0 are predicted correctly, while 14.82% $\pm$ 0.58% are not. Similarly, the true positive rate (TPR) and the false negative rate (FNR) indicate that 95.40% $\pm$ 0.45% of the links in class 1 are correctly predicted and 4.60% $\pm$ 0.45% are not.

Similar results of the GraphSAGE model are listed on the right-hand side. We observe from these two matrices that the ANN model shows a more accurate true positive prediction where the TPR is around 7% higher than that of the GraphSAGE model when taking 0.50 as the probability threshold. However, this outperformance is offset by the higher true negative prediction of GraphSAGE, which is approximately 6% higher than that of the ANN model. The same conclusion can be reached by comparing their F1-Scores, which show that the F1-Score of GraphSAGE is 0.1 higher than the ANN model.

We then compare these two models at the aggregated level by the ROC and PR AUCs. There are inconspicuous differences between the two ROC AUC values of the ANN and GraphSAGE models, both of which are equal to 0.96. Their high AUC value (greater than 0.95) indicates that these two models show identical and considerable performance when the predictions of the majority class and the minority class are treated equally important. However, the evident gap between the two PR curves shown in Figure 6 implies that the GraphSAGE model outperforms the ANN model when the minority class prediction is the focus, i.e., whether the class 1 (positive) links are correctly predicted or not. The PR AUC of the GraphSAGE model is about 8% higher than that of the ANN model. This implies that the local network information aggregated by GraphSAGE can enhance the model's performance in the prediction of positive links that are more important to design decisions.

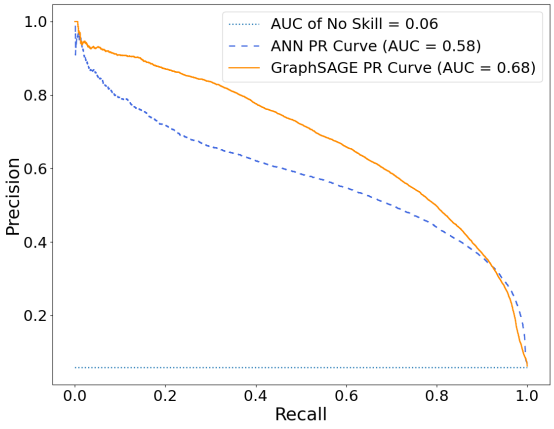


Fig. 6: PR curve example of Period Two link prediction using the ANN and GraphSAGE predictive models in fold four. The average PR AUCs are 0.59 $\pm$ 0.01 and 0.67 where the GraphSAGE model has a higher AUC than the ANN model in the PR curve.

[2]The training time for ANN and GraphSAGE are 7 minutes and 35 hours, respectively. The primary factor affecting the computational efficiency of GraphSAGE is the process of in- and out-neighborhood sampling [45]. However, since our proposed predictive model is not meant for real-time forecasting, we, therefore, prioritize improving model performance even if it means sacrificing computational efficiency.

Table 4: Confusion matrices of Period Two link prediction via the ANN and GraphSAGE models (probability threshold = 0.50)

		ANN Link Prediction		**GraphSAGE Link Prediction**	
		0	1	0	1
Actual	0	271108±1830 (TNR 85.18% ± 0.58%)	47177±1830 (FPR 14.82% ± 0.58%)	290619±1164 (TNR 91.31% ± 0.37%)	27666±1164 (FPR 8.69% ± 0.37%)
	1	914±89 (FNR 4.60% ± 0.45%)	18943±89 (TPR 95.40% ± 0.45%)	2259±122 (FNR 11.38% ± 0.61%)	17598±122 (TPR 88.62% ± 0.61%)
F1-Score		0.44 ± 0.01		0.54 ± 0.01	

5.3 GrapSAGE-Based Link Prediction for Networks With Different Link Strengths

In Section 5.2, we set the link cutoff at 3.03, which is the mean minimum link weight of monthly travel networks throughout the year from 2014 to 2017, following the approach described in our previous work [3]. To assess the generalizability of the proposed predictive model and demonstrate the importance of neighborhood information for different network sizes, we change the cutoff value from 0 to 16, corresponding to the ratio of positive links decreasing from 100% to 10%. Note that as the number of positive links declines, the data become even more imbalanced, with the majority of links being negative, thus making prediction even more challenging. Furthermore, to more easily trace the trend of prediction accuracy, we follow the experiment settings given in Table 3 and perform a five-fold cross-validation to train models with different link cut-off points.

5.3.1 Results for link prediction for networks with different link strengths

The evaluation metrics include F1-Score, ROC AUC, and PR AUC, all of which are averaged based on five-fold results. We first investigate the overall predictive performance of both models using F1-Score when the probability threshold is equal to 0.5, which is shown in Figure 7 (a). It is evident that the predictive powers of both GraphSAGE and ANN models decrease when the network becomes sparse, and the reason could be attributed to the aforementioned worse imbalanced issues of sparser networks. Furthermore, the consistently higher F1-Score of the GraphSAGE model suggests that neighborhood information indeed plays a role in improving prediction accuracy. The same conclusions can be drawn from the PR AUC plots in Figure 7 (b) when the emphasis is placed on the prediction of the minority class (positive links) across all probability thresholds.

In contrast, we find that regardless of network size, both ANN and GraphSAGE predictive models maintain the same high ROC AUC, which is around 0.96. This implies that these two models share a similar predictive power when putting the spotlight on both the majority and minority classes, and this power is robust enough to against the decline of the minority class.

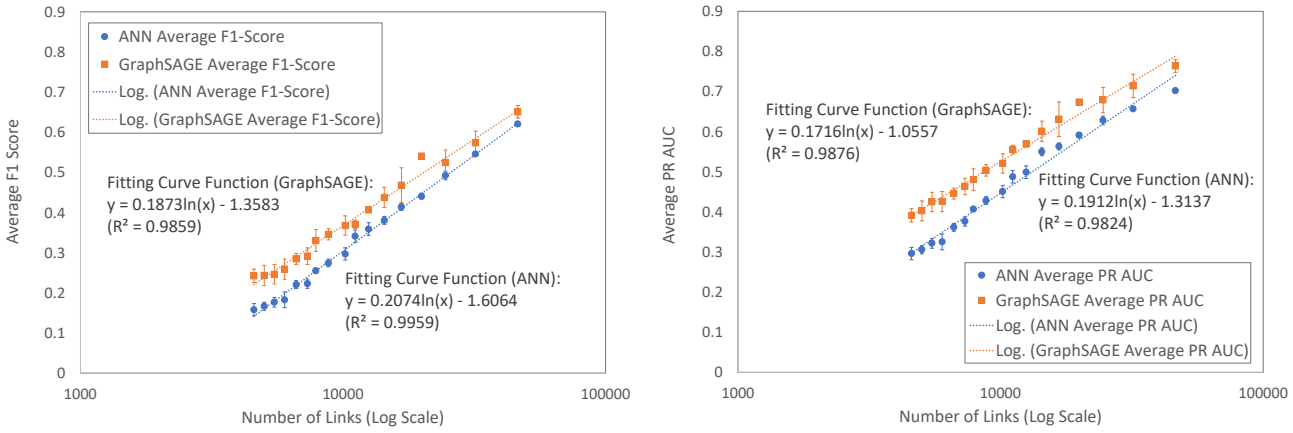
Lastly, to further validate that the decreased performance of GraphSAGE is irrelevant to the network topology, we conducted an experiment by using a subset of 2016 training data (showing totally different network topology) and 2017 testing data to test the model's predictive power. These synthetic data are generated by identifying the top 166 popular Divvy Bike stations in 2016 and the trips that occurred among these stations in 2016 and 2017. Although the trip networks constructed by these stations and trips exhibit distinct typologies in terms of their degree distributions compared to the real trip networks, the same decreased trend of predictive performance is still observed. Furthermore, the decreasing rate of the performance of both the synthetic network and the real network is highly correlated with the shrinking speed of the network size, thereby again demonstrating that the primary reason for poor prediction accuracy comes from the imbalance issue of sparse networks, rather than from network typologies.

5.4 Systems Design Decision Support

In this section, the proposed GraphSAGE model is utilized to assist BSS designers or other stakeholders in predicting system performance after a design strategy is proposed. We first formulate a design case and then evaluate the influence of the design decisions on users' trips and compare the prediction accuracy of GraphSAGE by comparing it with the baseline ANN model.

5.4.1 Divvy Bike design case

There are two levels of design decisions in this system. The first is *capacity-level design decision* that a designer should determine, i.e., the stations that need to be expanded or contracted and the number of docks that each station needs to add or remove. The second is *station-level design decision*, that is, a designer needs to decide (at a certain time point) which existing stations need to be removed and where new stations shall be built. By comparing the data from the Divvy bike station in May 2017 with the data from the Divvy bike station in May 2016, we assume that a decision maker proposed the following two-level design decisions at the end of May 2016 and would like to estimate their influence by predicting the connections of these key stations in May 2017.



(a) Average F1-Score (with error bars, probability threshold = 0.5)

(b) Average PR AUC (with error bars)

Fig. 7: F1-Scores and PR AUCs change with the number of links. The rightmost points in the plots correspond to 46,352 links when the cutoff value is equal to 0. We notice that the average F1-Scores and PR AUCs of both GraphSAGE and ANN models decrease logarithmically with the shrink of the network sizes, and the GraphSAGE model consistently has higher values than the ANN model.

1) *The capacity-level design decision*: stations in the set $S_1 = \{341, 195, 97, 72\}$ are planned to expand by adding 16 docks; stations in the set $S_2 = \{444, 496, 2, 445, 400, 489, 412, 407\}$ are planned to shrink by removing 8 docks. The locations of the stations in sets S_1 and S_2 are marked with dark blue and light blue pins, respectively, in Figure 8.
2) *The station-level design decision*: station in set $S_3 = \{372\}$ is planned to remove; 48 new stations in set $S_4 = \{524, 578, 522, 622, 550, 531, 517, 581, 575, 585, 523, 584, 580, 525, 520, 576, 619, 590, 591, 592, 623, 589, 586, 526, 620, 515, 579, 582, 514, 588, 573, 583, 577, 571, 574, 587, 595, 405, 527, 519, 602, 603, 598, 604, 605, 599, 606, 612\}$ are planned to construct. The locations of the stations in sets S_3 and S_4 are marked with red and green pins, respectively, in Figure 8.

We update the May 2016 network by applying these proposed design decisions. For example, the capacity and location information designed for the new stations in the set S_4 is added to the station list of 2016. Regarding their POI data, we adopted the approach described in Section 4.1 to count the number of each type of POIs around these designed stations in 2016. With the updated 2016 network, we used the trained GraphSAGE and ANN models to predict the connections of these critical stations within sets S_1, S_2, and S_4, as well as evaluate the prediction by comparing them with real connections in 2017.

5.4.2 Capacity-level station connection prediction

Taking the expansion station set S_1 as an example, we assume that the stations in S_1 are connected with n out of the N stations in Period Two in reality. For example, these stations connect $n = 149$ stations in 2017 and Divvy Bike had

Fig. 8: The geographical locations of stations in sets S_1, S_2, S_3 and S_4.

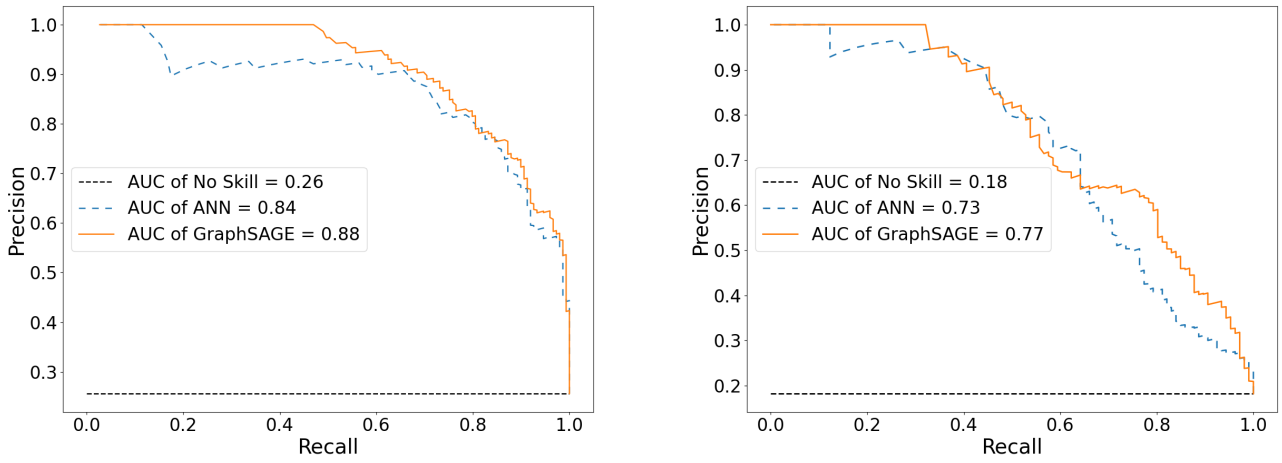
$N = 582$ in total that year. The GraphSAGE model predicts that these stations connect with M stations when the probability threshold is p, and m stations are correctly predicted. Therefore, the true positive (TP) equals m, the false negative (FN) equals $n - m$, the false positive (FP) equals $M - m$, and the true negative (TN) equals $N + m - n - M$. The PR curves correspondingly obtained across all probability thresholds from 0 to 1.

Table 5: Confusion matrices of expansion station trip network connections via ANN and GraphSAGE predictive model (probability threshold = 0.50). "Not Connection" denotes stations that were not connected to the stations in the set S_1 by trips in 2017 and vice versa for the "Connection" term. Similar definitions apply to Table 6 and Table 7.

	ANN Prediction		**GraphSAGE Prediction**	
	Not Connection	Connection	Not Connection	Connection
Not Connection	279±12 (TNR 64.53%±2.81%)	154±12 (FPR 35.47%±2.81%)	365±3 (TNR 84.20%±0.69%)	68±3 (FPR 15.80%±0.69%)
Connection	2 (FNR 1.07%±0.33%)	147 (TPR 98.93%±0.33%)	9±2 (FNR 6.17%±1.61%)	140±2 (TPR 93.83%±1.61%)
F1-Score	0.66±0.02		0.78±0.01	

Table 6: Confusion matrices of contraction station trip network Connections via ANN and GraphSAGE predictive model (probability threshold = 0.50).

	ANN Prediction		**GraphSAGE Prediction**	
	Not Connection	Connection	Not Connection	Connection
Not Connection	316±21 (TNR 64.84%±4.21%)	172±21 (FPR 35.16%±4.21%)	331±8 (TNR 67.75%±1.62%)	157±8 (FPR 32.25%±1.62%)
Connection	7±1 (FNR 7.23%±1.56%)	87±1 (TPR 92.77%±1.56%)	4±1 (FNR 4.04%±1.04%)	90±1 (TPR 95.96%±1.04%)
F1-Score	0.50±0.03		0.53±0.01	



(a) Expansion Case

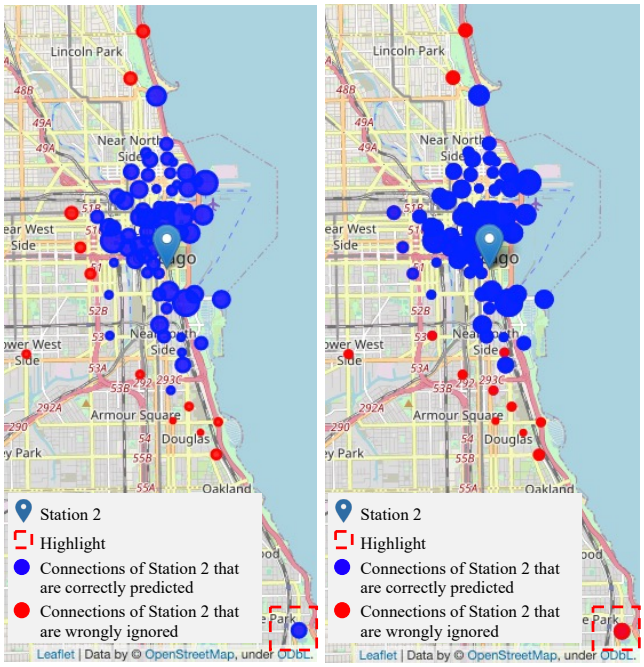
(b) Contraction Case

Fig. 9: PR curve example of *capacity-level design decision* evaluation through four-fold ANN and GraphSAGE trained models by predicting the network connections of key stations. We observe that the AUCs of the GraphSAGE model are 3% $\sim$ 5% higher than the ANN model in both expansion and contraction cases. For the expansion case, the average PR AUC of the Graphsage model is 0.88, which is 3% higher than that of the ANN model, equal to 0.85 ± 0.01. In terms of the contraction case, the average PR AUCs of the Graphsage and ANN models are, respectively, 0.78 ± 0.01 and 0.73 ± 0.02.

The confusion matrices and F1 scores of the expansion stations in S_1 and contraction stations in S_2 when the probability threshold is 0.50 are presented in Table 5 and Table 6. The results in Table 5 indicate that ANN more accurately predicts expansion stations' connections with higher TPR (98.93%) than GraphSAGE (TPR=93.83%), but greatly

sacrifices TNR (64.53%). In the contraction case, t-tests are conducted, comparing the means of ANN and GraphSAGE in terms of their TNRs, TPRs, and F1 scores. The null hypothesis is that there is no difference. The resulting *p-value*, 0.01, denotes a significant difference between the means of the ANN and the GraphSAGE TPRs, indicating that the GraphSAGE TPR (95.96%) is higher than that of the ANN (92.77%). However, the *p-values* of 0.23 and 0.07 in the tests of TNR and F1-Score imply that ANN and GraphSAGE perform an identical predictive power in the contraction case when the probability threshold = 0.50.

Overall, the F1-Scores of GraphSAGE on both expansion and contraction cases show its superiority, implying the important role of neighborhood information in influencing users' behaviors in BSS. This conclusion is further validated by the PR curves shown in Figure 9. That being said, when designers are to evaluate their proposed capacity-level design strategies, the GNN-based model is more reliable.



(a) GraphSAGE (b) ANN

Fig. 10: Link prediction of contracted design case, Station 2, using the GraphSAGE and the ANN predictive model. The size of the dots depicts the capacity of the station. (a) is the GraphSAGE predicted result when the probability threshold is equal to 0.78, where 0.78 is the optimal threshold for the GraphSAGE PR curve in Figure 9 (b). 68 of the 80 connections (85.00%) of Station 2 are correctly identified. (b) is the ANN predicted result when the probability threshold is equal to 0.88, where 0.88 is the optimal threshold for the ANN PR curve in Figure 9 (b). 67 of 80 connections (83.75%) of Station 2 are correctly predicted. For the selection of optimal thresholds, please refer to our previous work [50].

To visually demonstrate the models' predictive performance in this regard, we take Station 2 in S_2 as an example, as shown in Figure 10. The graphs show a decent predictive performance where over 80% connections of Station 2 are correctly predicted. Furthermore, GraphSAGE, which considers neighborhood information, slightly improves the prediction accuracy and correctly predicts the geographically farthest connection of Station 2, which is located in the lower right corner of the plots (highlighted by the red dashed square).

5.4.3 Station-level station connection prediction

With regard to predicting the connections of the newly built stations in S_4, we tested one ANN model and two different GNN models, as shown in Figure 11. The distinction between GraphSAGE and GraphSAGE-GroundTruth is that GraphSAGE kept the newly built stations independent and did not include their neighborhood information in the embedding. GraphSAGE-GroundTruth, instead, took the real neighborhood information from the new stations in May 2017 into the construction of network embeddings to test the best scenario that GraphSAGE prediction can reach. The AUC results indicate that GraphSAGE shows no better predictive power than ANN for these isolated new stations when there is no neighborhood information input. This is validated by the higher AUC of the GraphSAGE-GroundTruth model and its F1-Score in Table 7.

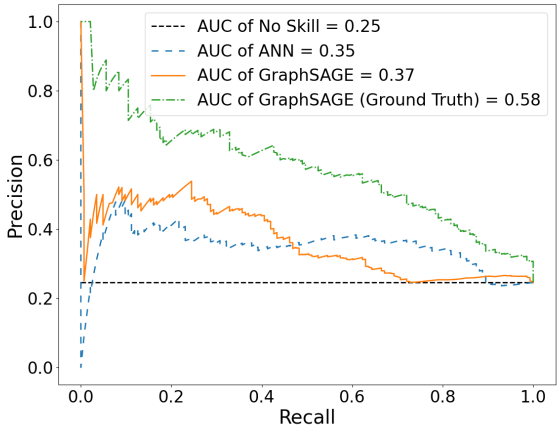


Fig. 11: PR curve example of *station-level design decision* evaluation via the fourth fold ANN and GraphSAGE trained models by predicting the network connections of the key stations. The average PR AUCs of ANN, GraphSAGE, and GraphSAGE (Ground Truth) by five folds are 0.33 ± 0.02, 0.36 ± 0.02, and 0.55 ± 0.04.

6 Limitations

There are a few limitations of the proposed GNN-based design decision support model in the current study that can lead to future investigations. First, the relatively lower predictive performance for the BSS network with a smaller size, having a worse imbalance issue, shows that our proposed model is better suited for link prediction of a denser network. It

Table 7: Confusion matrices of newly built station trip network connections via ANN and GraphSAGE predictive model with ground truth (probability threshold = 0.50).

	ANN Prediction		**GraphSAGE Prediction** (Ground Truth)	
	Not Connection	Connection	Not Connection	Connection
Not Connection	163±14 (TNR 37.13%±3.29%)	276±14 (FPR 62.87%±3.29%)	245±7 (TNR 55.72%±1.57%)	194±7 (FPR 44.28%±1.57%)
Connection	24±5 (FNR 16.92%±3.47%)	119±5 (TPR 83.08%±3.47%)	18±2 (FNR 12.73%±1.28%)	125±2 (TPR 87.27%±1.28%)
F1-Score	0.44±0.01		0.54±0.01	

should be worthwhile to study methods that can address this imbalance challenge and thus broaden the application of the model.

Second, a few assumptions made in this work could potentially impede the model from capturing reality, thus weakening its validity. For example, in the design case study given in Section 5.4, we assumed that all design decisions are made at the same time, e.g., at the end of May 2016. However, in reality, the decisions could be scattered across different months, so a dynamic model is preferred in this scenario to predict the station linkage in the short term. Also, we observed that it was rare that a user rented a bike and then returned it to the same location. But, our model is not designed to predict rare self-loop links. For example, in a mini-experiment, we selected a station (station 30) at random and created two duplicate stations (1030 and 1031) with identical attributes, such as the same dock numbers, locations, and surrounding POIs, as well as the same network neighbors. We then used the trained GraphSAGE model to predict the link probabilities of stations 30, 1030, and 1031 connecting to other stations separately. According to the results, we observed that there is a high probability of forming connections among the three duplicated stations, leading to the dominance of self-loop trips. Therefore, when using the model in a situation where there are duplicated nodes, it is better to assume that self-loop trips are not allowed. Another strategy is to combine all duplicated nodes in one node by stacking the number of docks of each node.

Third, the accuracy in predicting the newly added stations is relatively low, as shown in Section 5.4.3. To make this model applicable to the new stations, efforts are required to test more adjacency matrix approximation approaches and, subsequently, better estimates of the network neighborhood information of newly introduced stations. When applying the proposed model for decision support, it is more suitable to predict connections and travel demand between stations that have already been on the network.

Finally, in this study, only a few node features (station capacity, geographic coordinates, and surrounding POIs) are considered. However, there could be other factors influencing the accuracy of predictions, such as unserved travel demands (e.g., instances where people attempted to rent a bike at an empty station). One key reason for missing this information is the availability and accessibility of the data. For example, obtaining data on failed rental attempts at an empty station, which is essential to capture unserved travel demands, is not readily available in the current dataset. However, our proposed model is adaptable and generalizable to incorporate additional features. In our future study, a potential way to address this data limitation is to conduct surveys or statistics from BSS service Apps to estimate unserved travel demands.

7 Conclusion

In this study, we present a complex network-based approach to predict whether two stations in a shared mobility network would have sufficient travel demand to form a connection over a long timescale. The utility of the proposed approach in supporting system decisions in shared mobility networks is investigated and validated. In particular, we examine whether local network information impacts link formation using GNN models. In a case study of Divvy Bike in Chicago, two-hop neighborhood information is used to generate node embeddings for link prediction. The results show that the GNN model with local network information outperforms the one without, revealing the important role of local network structures in the formation of trip networks at the system level. We also test the model performance using local network information by changing the link strength from weak to strong, corresponding to the network size from large to small. The results indicate that the GNN model has maintained better performance than the ANN model regardless of the imbalanced data issue.

Finally, we present a design case study to illustrate how to apply the GNN-based predictive model to assist system designers to gain insights into their proposed design decisions. Using the trained GNN model to predict the network neighbors of the expansion stations, the contraction stations, as well as the newly built stations, we demonstrate the applicability of the predictive model in helping system designers make an initial assessment of the network connections of these critical stations. However, despite the fact that the current prediction of new stations is not satisfactory, the GroudTruth result validates that improvement can still be

achieved once a better approximation of the adjacency matrix is obtained.

In future work, as mentioned above, we plan to extend the current model to the link prediction of weighted trip networks. We also would like to conduct more tests to find additional factors to which the decreased performance in Figure 7 could be attributed. Lastly, more exploration of the adjacency matrix approximation approach will also be carried out to generate a more powerful predictive model for the evaluation of *station-level design decisions*. For example, one potential approach is to treat the edge prediction task as an iterative process. In this approach, the predicted result from the GraphSAGE model described in this paper serves as input for another identical GraphSAGE model. The iterative process continues until a satisfactory level of accuracy is achieved.

References

[1] Baxter, G., and Sommerville, I., 2011. "Socio-technical systems: From design methods to systems engineering". *Interacting with computers,* **23**(1), pp. 4–17.

[2] Wang, W., Chen, J., Zhang, Y., Gong, Z., Kumar, N., and Wei, W., 2021. "A multi-graph convolutional network framework for tourist flow prediction". *ACM Transactions on Internet Technology (TOIT),* **21**(4), pp. 1–13.

[3] Xiao, Y., and Sha, Z., 2022. "Robust design of complex socio-technical systems against seasonal effects: a network motif-based approach". *Design Science,* **8**.

[4] Schuijbroek, J., Hampshire, R. C., and Van Hoeve, W.-J., 2017. "Inventory rebalancing and vehicle routing in bike sharing systems". *European Journal of Operational Research,* **257**(3), pp. 992–1004.

[5] Yi, P., Huang, F., and Peng, J., 2019. "A rebalancing strategy for the imbalance problem in bike-sharing systems". *Energies,* **12**(13), p. 2578.

[6] Duan, Y., and Wu, J., 2019. "Optimizing rebalance scheme for dock-less bike sharing systems with adaptive user incentive". In 2019 20th IEEE International Conference on Mobile Data Management (MDM), IEEE, pp. 176–181.

[7] Fricker, C., and Gast, N., 2016. "Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity". *Euro journal on transportation and logistics,* **5**(3), pp. 261–291.

[8] Çelebi, D., Yörüsün, A., and Işık, H., 2018. "Bicycle sharing system design with capacity allocations". *Transportation research part B: methodological,* **114**, pp. 86–98.

[9] Lin, J.-R., Yang, T.-H., and Chang, Y.-C., 2013. "A hub location inventory model for bicycle sharing system design: Formulation and solution". *Computers & Industrial Engineering,* **65**(1), pp. 77–86.

[10] Si, H., Shi, J.-g., Wu, G., Chen, J., and Zhao, X., 2019. "Mapping the bike sharing research published from 2010 to 2018: A scientometric review". *Journal of cleaner production,* **213**, pp. 415–427.

[11] DeMaio, P., 2009. "Bike-sharing: History, impacts, models of provision, and future". *Journal of public transportation,* **12**(4), pp. 41–56.

[12] Borgnat, P., Abry, P., Flandrin, P., Robardet, C., Rouquier, J.-B., and Fleury, E., 2011. "Shared bicycles in a city: A signal processing and data analysis perspective". *Advances in Complex Systems,* **14**(03), pp. 415–438.

[13] Froehlich, J., Neumann, J., Oliver, N., et al., 2009. "Sensing and predicting the pulse of the city through shared bicycling.". In IJCAI, Vol. 9, pp. 1420–1426.

[14] Kaltenbrunner, A., Meza, R., Grivolla, J., Codina, J., and Banchs, R., 2010. "Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system". *Pervasive and Mobile Computing,* **6**(4), pp. 455–466.

[15] Yoon, J. W., Pinelli, F., and Calabrese, F., 2012. "Cityride: a predictive bike sharing journey advisor". In 2012 IEEE 13th international conference on mobile data management, IEEE, pp. 306–311.

[16] Ashqar, H. I., Elhenawy, M., Rakha, H. A., Almannaa, M., and House, L., 2021. "Network and station-level bike-sharing system prediction: A san francisco bay area case study". *Journal of Intelligent Transportation Systems*, pp. 1–11.

[17] Lin, L., He, Z., and 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**, pp. 258–276.

[18] He, S., and Shin, K. G., 2020. "Towards fine-grained flow forecasting: A graph attention approach for bike sharing systems". In Proceedings of The Web Conference 2020, pp. 88–98.

[19] Liu, J., Sun, L., Li, Q., Ming, J., Liu, Y., and Xiong, H., 2017. "Functional zone based hierarchical demand prediction for bike system expansion". In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 957–966.

[20] Singhvi, D., Singhvi, S., Frazier, P., Henderson, S. G., O'Mahony, E., Shmoys, D. B., and Woodard, D. B., 2015. "Predicting bike usage for new york city's bike sharing system". In AAAI Workshop: Computational Sustainability.

[21] Tran, T. D., Ovtracht, N., and d'Arcier, B. F., 2015. "Modeling bike sharing system using built environment factors". *Procedia Cirp,* **30**, pp. 293–298.

[22] Faghih-Imani, A., Hampshire, R., Marla, L., and Eluru, N., 2017. "An empirical analysis of bike sharing usage and rebalancing: Evidence from barcelona and seville". *Transportation Research Part A: Policy and Practice,* **97**, pp. 177–191.

[23] Chen, X., and Jiang, H., 2020. "Detecting the demand changes of bike sharing: A bayesian hierarchical approach". *IEEE Transactions on Intelligent Transportation Systems*.

[24] Gast, N., Massonnet, G., Reijsbergen, D., and Tribastone, M., 2015. "Probabilistic forecasts of bike-sharing

systems for journey planning". In Proceedings of the 24th ACM international on conference on information and knowledge management, pp. 703–712.

[25] Jiang, W., 2022. "Bike sharing usage prediction with deep learning: a survey". *Neural Computing and Applications,* **34**(18), pp. 15369–15385.

[26] Zell, A., 1994. *Simulation neuronaler netze*, Vol. 1. Addison-Wesley Bonn.

[27] Wu, F., Hong, S., Zhao, W., Wang, X., Shao, X., Wang, X., and Zheng, X., 2021. "Neural networks with improved extreme learning machine for demand prediction of bike-sharing". *Mobile Networks and Applications*, pp. 1–11.

[28] Wang, B., and Kim, I., 2018. "Short-term prediction for bike-sharing service using machine learning". *Transportation research procedia,* **34**, pp. 171–178.

[29] Chen, P.-C., Hsieh, H.-Y., Su, K.-W., Sigalingging, X. K., Chen, Y.-R., and Leu, J.-S., 2020. "Predicting station level demand in a bike-sharing system using recurrent neural networks". *IET Intelligent Transport Systems,* **14**(6), pp. 554–561.

[30] Lee, S.-H., and Ku, H.-C., 2022. "A dual attention-based recurrent neural network for short-term bike sharing usage demand prediction". *IEEE Transactions on Intelligent Transportation Systems*.

[31] Yang, H., Xie, K., Ozbay, K., Ma, Y., and Wang, Z., 2018. "Use of deep learning to predict daily usage of bike sharing systems". *Transportation research record,* **2672**(36), pp. 92–102.

[32] Li, X., Xu, Y., Zhang, X., Shi, W., Yue, Y., and Li, Q., 2023. "Improving short-term bike sharing demand forecast through an irregular convolutional neural network". *Transportation Research Part C: Emerging Technologies,* **147**, p. 103984.

[33] Rathkopf, C., 2018. "Network representation and complex systems". *Synthese,* **195**(1), pp. 55–78.

[34] Cui, Y., Ahmed, F., Sha, Z., Wang, L., Fu, Y., and Chen, W., 2020. "A weighted network modeling approach for analyzing product competition". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 84003, American Society of Mechanical Engineers, p. V11AT11A036.

[35] Sha, Z., and Panchal, J. H., 2016. "A degree-based decision-centric model for complex networked systems". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 50084, American Society of Mechanical Engineers, p. V01BT02A016.

[36] Barabási, A.-L., 2012. "The network takeover". *Nature Physics,* **8**(1), pp. 14–16.

[37] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G., 2008. "The graph neural network model". *IEEE transactions on neural networks,* **20**(1), pp. 61–80.

[38] Song, B., McComb, C., and Ahmed, F., 2022. "Assessing machine learnability of image and graph representations for drone performance prediction". *Proceedings of the Design Society,* **2**, pp. 1777–1786.

[39] Ferrero, V., DuPont, B., Hassani, K., and Grandi, D., 2022. "Classifying component function in product assemblies with graph neural networks". *Journal of Mechanical Design,* **144**(2).

[40] Ahmed, F., Cui, Y., Fu, Y., and Chen, W., 2021. "A graph neural network approach for product relationship prediction". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 85383, American Society of Mechanical Engineers, p. V03AT03A036.

[41] Liu, J., Ong, G. P., and Chen, X., 2020. "Graphsage-based traffic speed forecasting for segment network with sparse data". *IEEE Transactions on Intelligent Transportation Systems,* **23**(3), pp. 1755–1766.

[42] Zhang, J., Chen, F., Guo, Y., and Li, X., 2020. "Multi-graph convolutional network for short-term passenger flow forecasting in urban rail transit". *IET Intelligent Transport Systems,* **14**(10), pp. 1210–1217.

[43] Yoshida, A., Yatsushiro, Y., Hata, N., Higurashi, T., Tateiwa, N., Wakamatsu, T., Tanaka, A., Nagamatsu, K., and Fujisawa, K., 2019. "Practical end-to-end repositioning algorithm for managing bike-sharing system". In 2019 IEEE International Conference on Big Data (Big Data), IEEE, pp. 1251–1258.

[44] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M., 2020. "Graph neural networks: A review of methods and applications". *AI Open,* **1**, pp. 57–81.

[45] Hamilton, W. L., Ying, R., and Leskovec, J., 2017. "Inductive representation learning on large graphs". In Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 1025–1035.

[46] Perozzi, B., Al-Rfou, R., and Skiena, S., 2014. "Deepwalk: Online learning of social representations". In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 701–710.

[47] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q., 2015. "Line: Large-scale information network embedding". In Proceedings of the 24th international conference on world wide web, pp. 1067–1077.

[48] Chen, H., Perozzi, B., Al-Rfou, R., and Skiena, S., 2018. "A tutorial on network embeddings". *arXiv preprint arXiv:1808.02590*.

[49] Xiao, Y., and Sha, Z., 2020. "Towards engineering complex socio-technical systems using network motifs: A case study on bike-sharing systems". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 84003, American Society of Mechanical Engineers, p. V11AT11A045.

[50] Yinshuang Xiao, Faez Ahmed, Z. S., 2022. "Travel links prediction in shared mobility networks using graph neural network models". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Amer-

ican Society of Mechanical Engineers.
[51] Wiki, O., 2022. Overpass turbo — openstreetmap wiki,. [Online; accessed 4-February-2022].
[52] Yang, Y., and Diez-Roux, A. V., 2012. "Walking distance by trip purpose and population subgroups". *American journal of preventive medicine,* **43**(1), pp. 11–19.
[53] Ahmed, F., Cui, Y., Fu, Y., and Chen, W., 2022. "Product Competition Prediction in Engineering Design Using Graph Neural Networks". *ASME Open Journal of Engineering,* **1**, 05. 011020.
[54] Brownlee, J., 2020. *Imbalanced classification with python: Better metrics, balance skewed classes, cost-sensitive learning*. Machine Learning Mastery.
[55] Divvy_Bike, 2020. Divvy system data. Last accessed 8 February 2022.
[56] Bengio, Y., and Grandvalet, Y., 2003. "No unbiased estimator of the variance of k-fold cross-validation". *Advances in Neural Information Processing Systems,* **16**.
[57] Nogueira, F., 2014–. Bayesian Optimization: Open source constrained global optimization tool for Python.