Proceedings of the ASME 2022
International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC-CIE2022
August 14-17, 2022, St. Louis, Missouri

DETC2022-90694

TRAVEL LINKS PREDICTION IN SHARED MOBILITY NETWORKS USING GRAPH NEURAL NETWORK MODELS

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

ABSTRACT

The emerging sharing mobility systems are gaining increasing popularity because of the significant economical and environmental benefits. To facilitate the operation of sharing mobility systems, many studies are conducted to analyze and predict users' travel behaviors. However, most research focuses on investigating every station's usage and demand; therefore, insight into the user behavior and travel demand between stations from origin to destination is little known. Aiming to better understand the factors that would influence origin-destination travel demand, we present a complex network-based approach to predicting the travel demand between stations (e.g., whether two stations have sufficient trips to form a strong connection in a month) in sharing mobility systems. Particularly, in this study, we are interested in knowing whether local network information (e.g., the 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 in it. To answer this question, we adopt Graph Neural Network (GNN), in which the concept of network embedding can capture and quantify the effect of local network structures. The results are compared with the regular artificial neural network (ANN) model without network embedding. This study is demonstrated using the bike sharing system, Divvy Bike in Chicago, as an example. We observe that the GNN prediction gains up to 9% higher performance than that of the ANN model. This implies that the local network information contributes to the formation of sharing mobility network. Moreover, it is found that when predicting the following year's network, the model that employs the node embedding obtained from the previous year's network outperforms the model with the node embedding obtained from the ANN predicted networks.

Keywords: Shared mobility network; Socio-technical Systems; Complex networks; Graph neural network; GraphSAGE.

1 BACKGROUND AND INTRODUCTION

1.1 MOTIVATION

Shared mobility system is one typical complex socio-technical system that its functionality and complexity highly relate to social behaviors [1]. This emerging system has experienced rapid growth in the recent decade because of its sustainable and environment-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 it 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 due to imbalanced demands for points of interest at different locations [2] or suboptimal system design decisions, e.g., an imbalanced dock distribution in bike sharing systems (BSS) [3]. Effective solutions to these problems are essential for the success of system operation, high rate of customer retention, and long-term quality service.

*Corresponding author.

To this end, attempts have been made in the existing literature to develop both 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 local service systems of a BSS network for enhanced robustness against seasonal effects [3]. To validate the design approach and demonstrate the utility of the resulting rebalancing strategies for a shared mobility system, it is necessary to have a dedicated predictive model to forecast the travel demand after rebalancing. The benefits of having a high-performance predictive model for the system are twofold: 1) since real-world testing with any rebalancing strategy in a large-scale system is costly, a predictive model can be a surrogate model to allow stakeholders to simulate and test different methods and configurations more efficiently. 2) A predictive model can be used to forecast the usage of a system's capacity and thus guide stakeholders in making more effective decisions on the system's operation, maintenance, and future development.

The recent advancement in machine-learning literature has greatly promoted the research on various predictive models of user travel patterns and helped gain better insights into shared mobility. Common factors influencing user travel behaviors include peak hours, holiday impacts, climate changes, surrounding Point of Interest (POI), and spatial dependencies among serving stations, etc. [10–13]. These models can be put into two categories: parametric statistical models and machine-learning models [10]. Some popular statistical models include linear regression models [14–16], Bayesian model [17], and Markov queuing model [18]. One advantage of these statistical models is their interpretability. For example, in [16], 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 statistical models is that they are usually built with many assumptions, resulting in low model validity and predictive performance. Machine-learning models typically outperform statistical models in terms of predictive power, but at the expense of losing interpretability. Some representative models include Convolutional Neural Network (CNN) [19] for traffic demand prediction considering the spatiotemporal characteristics of demand distribution and environmental factors, Long Short-Term Memory (LSTM) [20] unit for short-term (e.g, 10 min) prediction of bike availability, and Recurrent Neural Networks (RNNs) [21] for station-level prediction of real-time rental and return demands.

In existing models, stations in a shared mobility system are often treated independently, so they only predict station-level rental and return demands but do not tell where the return comes from and the rental goes. That means insights into the interconnections between stations are lacking.

In our previous works [3, 22], we have demonstrated that complex networks could be a useful tool for shared mobility system research and identified that some important local network structures (called network motifs) in the formation of the shared mobility network. The advantages of taking a network perspective in shared mobility systems research are twofold. First, 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. Second, network models can provide means to investigating the interactions between local system structures (e.g., travel patterns among three stations in a local area) and global system performance (e.g., the network robustness), which is essential to answer the following **research question**: whether and to what extent does the local network information (e.g, structure and node features) play a role in the formation of shared mobility network. So, in this study, the **objective** is to develop a complex network-based approach to predicting travel demand between stations in shared mobility systems based on local network information. The travel demand refers to the number of trips occurring from one station to another in a period of time.

1.2 PROBLEM FORMULATION

This study only focuses on docked shared mobility systems, i.e., the system includes fixed service stations with limited docking capacity. Our goal is to predict the existence of a strong connection (with a defined number of trips) between any two stations in the next year based on the previous year's trip record. We assume that at the end of the previous year, we already know the infrastructure setup of the next year, e.g., the geographical location of new stations, the number of added/removed docks in each station, and surrounding POIs of each station. To avoid seasonal effects, we split 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 the month i ($i = 1, ..., 12$) of next year with the trip data of month i in the previous year.

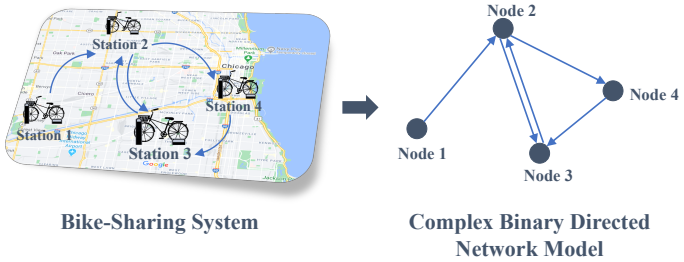


FIGURE 1: Illustration of modeling a shared mobility system by complex network.

As shown in Figure 1, a network is used to model the travel demand within a month in a BSS, where the nodes represent the stations and node attributes indicate station information such as geographic coordinates, capacity, and the number of surrounding POIs. Since we are only interested in strong connections, only if a link with the number of trips exceeds a defined threshold will be kept in the network. Therefore, the network under investigation is a directed unweighted network. 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 the 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 challenges for machine-learning algorithms.

In this paper, we adopt Graph Neural Network (GNN) model [23, 24] based on GraphSAGE for its capability of capturing and quantifying the effect of local network structures through network embedding – node representation by sampling and aggregating features from its network neighborhood [25]. More details of GNN are given in Section 2. The **contribution** 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. This approach can support system designers to test and experiment their design strategies with different configurations. This approach can also be easily generalized to other complex networked systems, e.g., air transportation networks and supply-chain networks.
2. By comparing to regular Artificial Neural Network (ANN) models, we revealed the important role of neighboring nodes' information in predicting travel demand in shared mobility systems. When two-hop neighbors' information of a station are included, the model's predictive performance is 9% higher than that without neighbors' information.
3. Specifically for shared mobility systems, we found that when taking the previous year's network structure to approximate the node embedding in predicting a link existence of next year, the GraphSAGE model outperforms the model that use the node embedding obtained from the ANN predicted networks.

The rest of the paper is organized as below. In Section 2, we introduce the knowledge background about GNN and GraphSAGE algorithm. Then, the proposed complex network-based approach and the associated methods for model analysis and evaluation are presented in Section 3. Section 4 takes Divvy Bike in Chicago as an example to demonstrate the utility of our approach. Finally, the paper is concluded in Section 5 with future work and closing thoughts.

2 KNOWLEDGE BACKGROUND

Graphs are an important representation for complex systems [26–28] that not only do they model interconnection and interrelation between system elements, but also the leverage of complex network theories [29]. 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) [30] was proposed in 2008, and due to its outstanding performance, GNN has been widely used across domains since then. For example, Ahmed *et al.* [23] developed a GNN-based method to predict the competition relationships between different car models in a vehicle co-consideration network. The model provided great insights into the key engineering attributes that promote the formation of car competitions.

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, graph match, etc. could be accomplished [24]. In recent years, many variants of GNN have been developed, each based on a different node embedding strategy [25, 31, 32]. For example, the well-known DeepWalk algorithm [31] 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 [33].

GraphSAGE is another remarkable variant of GNN in that it is a general inductive framework. Varied from 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 [25]. 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 [25].

3 METHODOLOGY

An overview of the complex network-based approach to predicting travel demand in shared mobility systems is shown in Figure 2. In this approach, we start by modeling a shared mobility

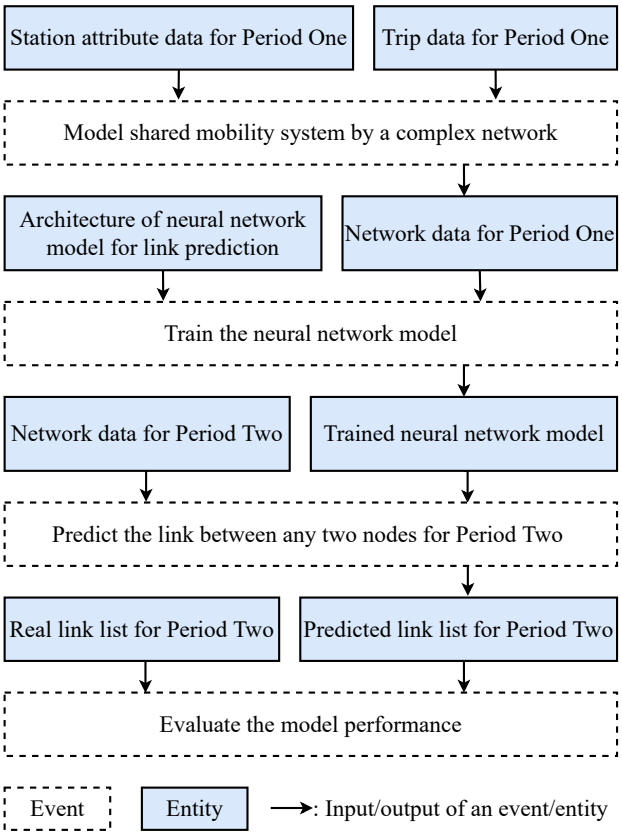


FIGURE 2: Complex network-based predicting framework for shared mobility system with Neural Network (**Period One**: Month i in year Y, **Period Two**: Month i in year $Y+1$, $i = 1,...12$)

system as a complex network using historical system data, i.e., Period One data, including station attributes and trip data. After obtaining the network model, we utilize Period One network data to train the predictive models, including the ANN model in Section 3.2 and the GraphSAGE-based model in Section 3.3. The trained model is then employed to predict the Period Two links based on the updated node 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 3.4.

3.1 NODE ATTRIBUTES

The node attributes considered in this study include the station geographic coordinates, the number of docks, as well as station surrounding POIs. Geographic coordinates can be used to calculate the distance between two stations, and the number of docks in a station determines the maximal number of bikes that users can rent/return from that station. Current research indicates that shared mobility systems exist evident travel patterns

TABLE 1: POI classification

Category Name	POI Names In Each Group
Financial	Bank
Education	Library, School
Recreational&Tourism	Cinema, Theatre, Hotel, Museum
Residential	Apartment, Hairdresser, Supermarket
Sustenance	Restaurant
Healthcare	Dentist, Hospital
Transportation	Public Transport Stops

between certain city functional zones due to users' specific commute purposes [12, 13]. In the study [12], for example, He and Shin divided POIs 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 that between stations in recreational and commercial areas.

In this study, the POI data is collected by Overpass turbo [34] including the name of each POI and its geographic coordinates. We first classify the POIs into seven categories, as shown in Table 1. Then, in Figure 3, we draw a circle of radius R with the target station at 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. Taking Station 2 as an example, its POI attribute vector is $[0,1,0,2,1,1,1]$, indicating that there is one school, two supermarkets, one restaurant, one hospital, and one public transportation stop within the circle of radius R. Regarding the value of the radius, R, we learned from reference [13] that a person would prefer to take a bike when the origin-to-destination distance is between 1.5 miles and 2.7 miles. This means that 1.5 miles are the longest distance that people are willing to commute by walking. Therefore, POIs within 1.5 miles from a station provide the best representation of the station's surroundings.

3.2 BASELINE: ANN LINK PREDICTING MODEL

In this study, we take ANN as a baseline model. As shown in Figure 4, the architecture of a simple ANN model consists of an input layer, one hidden layer, and an output layer. Training a model starts from formulating link features. In a shared mobility network, the link features are determined by the two connecting nodes. Accordingly, we use the concatenation of start and end node features with size N to represent the directed link features with size $2N$. To improve the training stability, max-min normal-

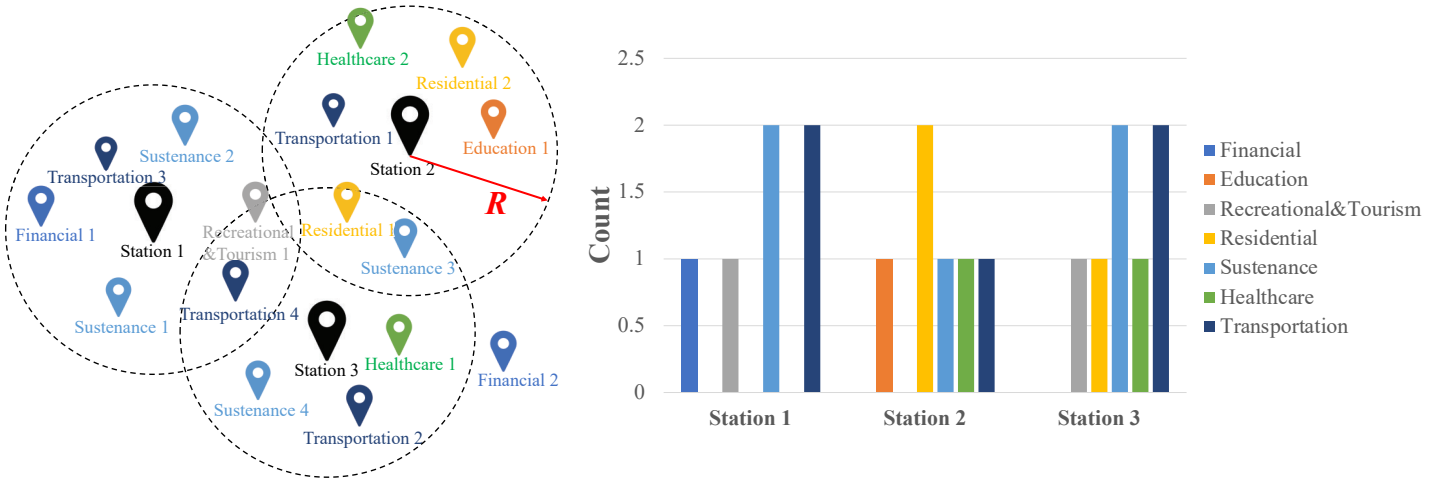


FIGURE 3: An example of transferring POIs to station attributes.

ization 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 has 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 a link exists 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 the 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.

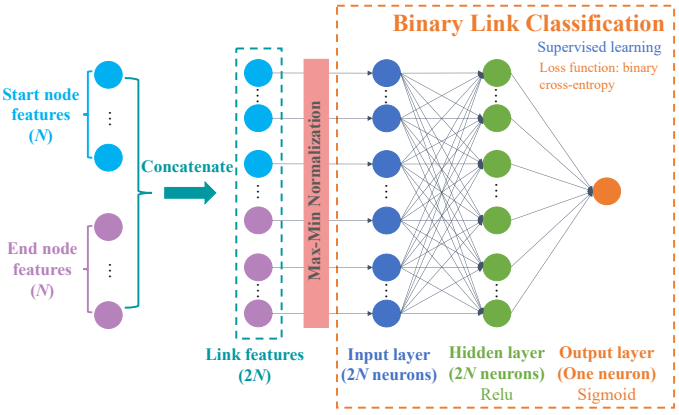


FIGURE 4: Architecture of the ANN link predicting model.

3.3 GNN-BASED LINK PREDICTING MODEL

Model architecture As illustrated in Figure 5, 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.

Similar to the ANN model, the learned node embeddings of the start node and end node are concatenated to represent the link embedding with the size of $2M$. Because the data has 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. 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.

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

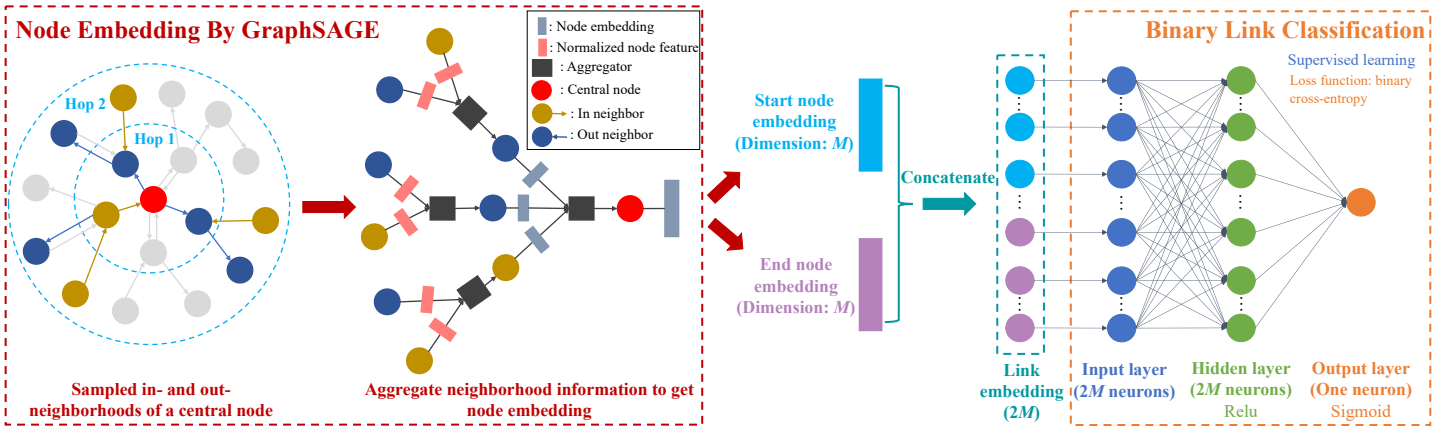


FIGURE 5: Architecture of the GraphSAGE link predicting model.

as class 0. The network data is for GraphSAGE 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 [23].

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

Methods for adjacency matrix prediction 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 from the same domain and have similar characteristics. It is worth noting that since isolated nodes do not have neighbors, GraphSAGE can learn nothing out of it. In this study, given that the training and testing shared mobility networks are from the same month but different years, they share similar characteristics. However, the testing network of a future year is unknown. To get the embedding of the testing nodes, an approximate adjacency matrix must be obtained to estimate their neighbors.

According to the study [23], there are several approaches for predicting the adjacency matrix, including directly using the training network or building a separate machine learning model for such an approximation. Three methods are tested in this paper.

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

3.4 LINK PREDICTION EVALUATION

Given that link prediction is the same as binary classification, several commonly used metrics for binary classification are chosen, including the confusion matrix, F1-Score, receiver operating characteristic (ROC) curve, precision-recall (PR) curve, and area under the ROC and PR curves (a,k.a. AUC, area under curves). The confusion matrix describes the performance of a classifier at a given probability threshold, which is typically 0.5. F1-Score is the harmonic mean of precision and recall in this given threshold such that 1.0 is the best and 0 is the worst. ROC curve plots the true positive rate (TPR) vs. the false positive rate (FPR) across all probability thresholds from 0 to 1. PR curve is the plot of the precision against the recall at all possible thresholds. Both ROC AUC and PR AUC are aggregated measures of different models' performances. 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 a better predictive performance. For imbalanced classification problems where the majority of observations is negative case and the minority of observations is positive case, ROC analysis provides equal insights on the model's predictive

performance on both cases. PR analysis focuses more on the model's ability in predicting the minority case [35].

4 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 4.2, the GraphSAGE link predicting model is compared with the ANN model to test whether the local network information (i.e., the node embedding features) has any impact on the link prediction. In Section 4.3, we compare three methods to approximate the adjacency matrix to identify to what extent the local network information can impact the link prediction in shared mobility networks.

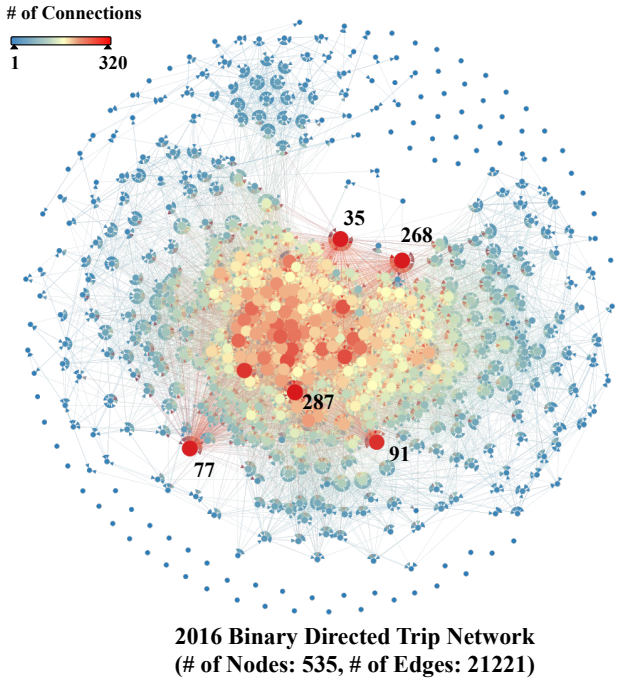


FIGURE 6: A visualisation of the Divvy Bike trip network in May 2016 after removing the links with less-occurred trips

4.1 DATA SOURCE

The Divvy Bike data is available to the public [36], 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, the 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 the binary directed trip networks by removing the links with less-occurred trips (i.e., those occurred no more than three times in a month). Taking the Period One network as an example, a visualization of this binary directed network is shown in Figure 6. The top hub stations' information for both two periods is listed in Table 2. It is observed that the top five hub stations in Period One are the hubs in Period Two despite the slight change of ranking. In terms of POI information, a total of 2,269 POIs in Period One and 2,403 POIs in Period Two are collected based on the method presented in Section 3.1. According to Section 3.1, each station has geographic coordinates (two features), the number of docks (one feature), and POIs (seven features), for a total of ten features.

4.2 GRAPHSAGE-BASED LINK PREDICTION

Data preparation for ANN-based link prediction In the ANN model, each candidate link within a trip network, represented by a node pair, is a data sample. Accordingly, there are 285,690 data samples in the Period One network, with 21,221 links classified into class 1 and 264,469 links into class 0. We split all class 1 links into 70% for training and 30% for validation. Meanwhile, to avoid imbalanced training, the same number of class 0 links are drawn at random from the class 0 sample pool and added to the training and validation sets. With these treatments, there are 14,855 class 1 samples and 14,855 class 0 samples in the training set. Given that Divvy Bike had 582 stations during Period Two (May 2017), the total number of potential links in the testing dataset is 338,142.

Data preparation for GraphSAGE predicting model The GraphSAGE model first requires network data to learn node embeddings. For the training model of node embedding, we feed it with the entire Period One network. To ensure a fair comparison, we stick to the same configuration in the second stage for training the link classification model as what has been adopted in the ANN model. For the approximate adjacency matrix obtained from the modified Period One network, those stations that are no longer operated in Period Two are removed and 48 new stations are added as isolated nodes.

Experiment settings We summarize the model settings as well as hyperparameter values in Table 3.

Results We first assess the performance of these two models using the confusion matrix and F1-Score, as shown in Table 4. The left-hand side shows the confusion matrix and F1-Score of the ANN model. The confusion matrix includes four different combinations of predicted and actual classes, where

TABLE 2: Top five hub stations information in trip network 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

there are 278,859 true negatives, 39,426 false positives, 1,242 false negatives, and 18,595 true positives. The true negative rate (TNR) and false positive rate (FPR) reveal that 87.61% of links in class 0 are predicted correctly, whereas 12.39% are not. Likewise, the true positive rate (TPR) and false negative rate (FNR) indicate that 93.64% of links in class 1 are predicted correctly, and 6.36% are not. Similar results of the GraphSAGE model are listed on the right-hand side. We observe from these two matrices that these two models share a similar predictive power in both categories when taking 0.5 as the probability threshold. The same conclusion can be reached by comparing their F1-Scores, which show that the difference is less than 0.005.

We then compare the ROC and PR curves of these two models in Figure 7. The inconspicuous differences between the two ROC curves, as well as their high AUC values (greater than 0.95), indicate that these two models show an identical and considerable performance when the predictions of majority class and minority class are treated equally important. However, the evident gap between the two PR curves 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 9% higher 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.

TABLE 3: Experiment parameter settings

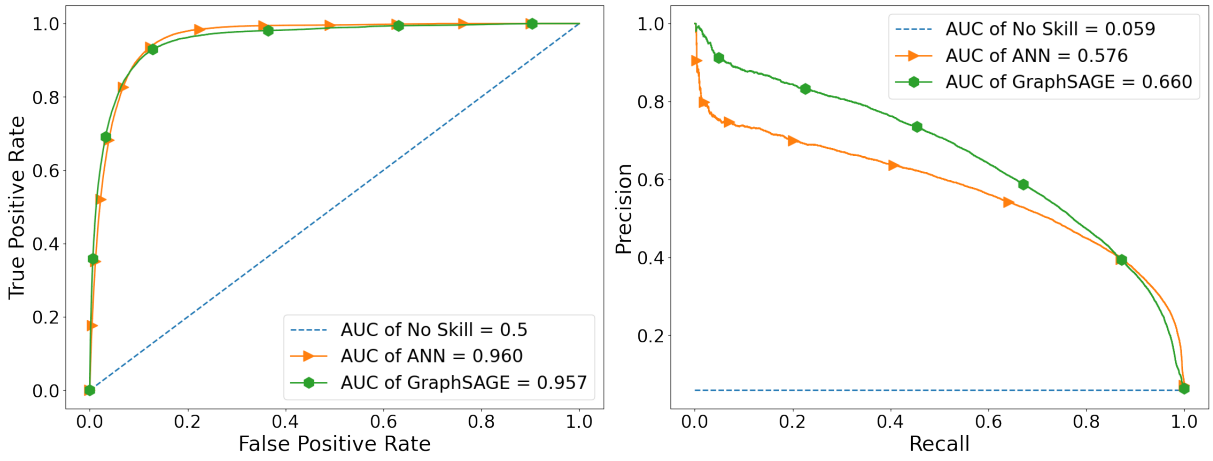
Setting Items	**Model Applied**	**Value**
Neighborhood search depth	GraphSAGE	2
# of Sampled in- and out-neighbors in two hops		10
Node embedding size		30
Input and hidden layer size for GraphSAGE		60
Input and hidden layer size for ANN	ANN	20
Minibatch size	GraphSAGE and ANN	192
Epoch		500
Learning rate		4e-4
Dropout		0

4.3 LINK PREDICTION BY GRAPHSAGE MODELS WITH DIFFERENT APPROXIMATE ADJACENCY MATRICES

Four different approximate adjacency matrices To probe into the effect of local information on the performance of GraphSAGE models, we investigate four different models with different approximate adjacency matrices for generating the node embedding of Period Two network. Both Model 1 and Model 2 use the ANN model to predict the probability of each candidate link, and then use the optimal ROC and PR thresholds to label the links. As shown in Figure 8 (a), the optimal ROC threshold, denoted by a red dot on the ROC curve, equals 0.485. To find this point, we first introduce the concept of Geometric Mean or $G-Mean$ in Equation 1, which is a metric used to seek a balance between false positive and true positive rates. In the ROC curve, the threshold with the largest $G-Mean$ leads to the optimal threshold. The optimal PR threshold is presented in Figure 8 (b) with a value of 0.837. Similar to the ROC threshold, we calculate the F1-Score with each threshold through Equation 2. The threshold with the largest F1-Score is the optimal one that produces the best balance between precision and recall [35]. With regard to Model 3, the approximate adjacency matrix of Period Two is obtained from the modified Period One network that is the same as the one adopted in Section 4.2. Model 4 employs the

TABLE 4: Confusion matrices of Period Two link prediction via ANN and GraphSAGE link predicting model (probability threshold = 0.5)

		ANN Link Prediction		GraphSAGE Link Prediction	
		0	1	0	1
Actual Class	0	278859 (TNR 87.61%)	39426 (FPR 12.39%)	278930 (TNR 87.64%)	39355 (FPR 12.36%)
	1	1262 (FNR 6.36%)	18595 (TPR 93.64%)	1475 (FNR 7.43%)	18382 (TPR 92.57%)
F1-Score		0.478		0.474	



(a) ROC Curve

(b) PR Curve

FIGURE 7: ROC and PR curve of Period Two link prediction via ANN and GraphSAGE predicting model. We notice that while the two models have similar AUC, the GraphSAGE model has higher corresponding precision values in the PR curve.

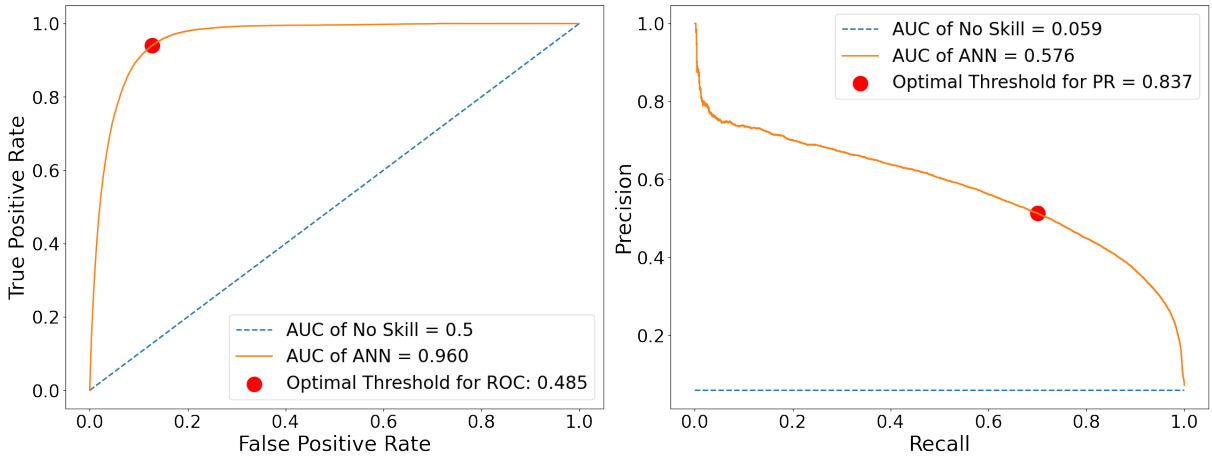
real Period Two network to generate the node embedding, which serves as the ground truth for the evaluation of the other three models.

$$G\,Mean = \sqrt{TPR * (1 - FPR)} \tag{1}$$

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2}$$

Results The confusion matrices and F1-Scores of all four models are summarized in Table 5. The results show that Model 2, Model 3, and Model 4 have similar TNRs that are 5% - 6% higher than that predicted by Model 1. However, when comparing the TPRs, Model 1 shows a comparable performance (94.84%) with the ground truth and achieves a higher prediction accuracy than Model 2 and Model 3. Model 2 has the lowest TPR (87.54%). The imbalanced performance of Model 1 can be explained by the fact that it is built upon a relatively lower probability threshold (0.485), resulting in a denser approximate network for embedding. This increases the likelihood that one link is classified as class 1, leading to a higher TPR (94.48%) and FPR (17.58%). Model 2, on the contrary, derived from a relatively higher probability threshold (0.837), has a sparser approximate network for embedding, thereby more links are predicted as class 0, making it has a lower TPR (87.54%) and FPR (11.13%). In general, F1-Scores depict that the overall performances of Model 2, Model 3, and Model 4 are similar and 8% higher than Model 1. If taking TNR, TPR, and F1-Score all into account, Model 3 yields the best performance.

To gain a more comprehensive insight into these four models' performance across all probability thresholds, we plot their ROC and PR curves in Figure 9. The four ROC curves closing to each other and seemingly following the same trend de-



(a) ROC Curve

(b) PR Curve

FIGURE 8: The optimal probability threshold for ROC and PR

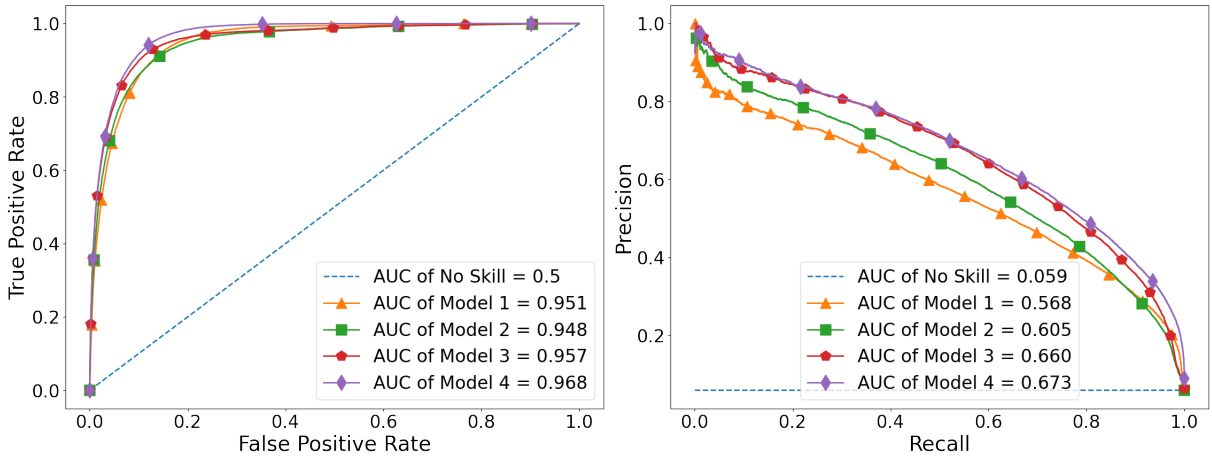
TABLE 5: Confusion matrices of period two link prediction via different approximate adjacency matrices (probability threshold = 0.5)

		Model 1: ANN Predicted Trip Network (With Optimal Threshold for ROC)		**Model 2:** ANN Predicted Trip Network (With Optimal Threshold for PR)	
		0	1	0	1
Actual Class	0	262343 (TNR 82.42%)	55942 (FPR 17.58%)	282863 (TNR 88.87%)	35422 (FPR 11.13%)
	1	1475 (FNR 5.52%)	18382 (TPR 94.48%)	2474 (FNR 12.46%)	17383 (TPR 87.54%)
F1-Score		0.397		0.478	
		Model 3: Modified May 2016 Trip Network		**Model 4:** May 2017 Real Trip Network	
		0	1	0	1
Actual Class	0	278930 (TNR 87.64%)	39355 (FPR 12.36%)	278203(TNR 87.41%)	40082 (FPR 12.59%)
	1	1475 (FNR 7.43%)	18382 (TPR 92.57%)	1042 (FNR 5.25%)	18815 (TPR 94.75%)
F1-Score		0.474		0.478	

notes that there is no evident variation in their predictive power when putting the spotlight on both the majority and minority classes. When more emphasis is placed on the prediction of minority class (positive links), the PR curves indicate that Model 3's performance is much closer to the ground truth (Model 4) and exhibits 10% and 6% improvements compared to Model 1 and Model 2. This further demonstrates the superiority of Model 3 over Model 1 and Model 2. Consequently, in this shared mobility network, we can conclude that utilizing the modified Period One network (May 2016 Divvy Bike network) to approximate the neighbors of nodes in Period Two network (May 2017 Divvy Bike network) is adequate to generate satisfactory predictions. One possible reason could be that users' travel patterns in a particular season do not differ too much from year to year if there are no significant changes in the city infrastructure and POI surrounding the stations. This results in relatively stable shared mobility networks with few changes in two consecutive years. For example, Table 2 shows that at least the top five hubs in Divvy Bike system appear in both periods from 2016 to 2017.

5 CONCLUSION

In this study, we present a complex network-based approach to predict if two stations in a shared mobility network would have



(a) ROC Curve

(b) PR Curve

FIGURE 9: ROC and PR curve of Period Two link prediction via GraphSAGE predicting models by using four different approximate adjacency matrices. We observe that Model 3 has higher AUCs than both Model 1 and Model 3. Model 1 has a marginally better ROC AUC and a marginally worse PR AUC than Model 2, which is consistent with the fact that Model 1 is built upon the optimal ROC threshold, whereas Model 2 is based the optimal PR threshold.

sufficient travel demand to yield a strong connection. Based on the proposed approach, we investigate whether local network information impacts the formation of a directed unweighted link in shared mobility networks using GNN models. In a case study on the Divvy Bike in Chicago, two-hop neighborhood information is used to generate node embeddings and further for link prediction. The results show that the model with the input of local network information outperforms the one without, revealing the important role of local network structure in the formation of global-level trip network topology. Finally, we compare multiple ways to approximate neighborhoods for future networks and show that in shared mobility networks, the predictive performance of the GraphSAGE model is the best when using the adjacency matrix achieved from the previous year's network. In future work, we plan to extend the current model to the link prediction of weighted trip networks. That means in addition to predict the existence of links, the model can also predict how many trips would occur from one station to another.

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] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] Du, B., Hu, X., Sun, L., Liu, J., Qiao, Y., and Lv, W., 2020. "Traffic demand prediction based on dynamic transition convolutional neural network". *IEEE Transactions on Intelligent Transportation Systems,* ***22***(2), pp. 1237–1247.

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

[21] 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.

[22] 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.

[23] Ahmed, F., Cui, Y., Fu, Y., and Chen, W., 2021. "A graph neural network approach for product relationship prediction". *arXiv preprint arXiv:2105.05881*.

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

[32] 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.

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

[34] Wiki, O., 2022. Overpass turbo — openstreetmap wiki,. [Online; accessed 4-February-2022].

[35] Brownlee, J., 2020. *Imbalanced classification with python: Better metrics, balance skewed classes, cost-sensitive learning*. Machine Learning Mastery.

[36] Divvy_Bike, 2020. Divvy system data. Last accessed 8 February 2022.