applied sciences

MDPI

Article

Scatter-GNN: A Scatter Graph Neural Network for Prediction of High-Speed Railway Station—A Case Study of Yinchuan–Chongqing HSR

Manfu Ma, Yiding Zhang *, Yong Li, Xiaoxue Li and Yiping Liu

College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China
* Correspondence: easonzhangcs@outlook.com; Tel.: +86-199-1646-8220

Featured Application: This paper takes the Yinchuan–Chongqing high-speed railway (HSR) as an example and proposes an auxiliary strategy for the traditional line planning scheme. With the increasing density of China's high-speed railway line network, this method can be applied to railway station selection, line design, and other similar practices. It may also be applied to other line scenarios such as highways in the future.

Abstract: The Yinchuan–Chongqing high-speed railway (HSR) is one of the "ten vertical and ten horizontal" comprehensive transportation channels in the National 13th Five-Year Plan for Mid- and Long-Term Railway Network. However, the choice of node stations on this line is controversial. In this paper, the problem of high-speed railway station selection is transformed into a classification problem under the edge graph structure in complex networks, and a Scatter-GNN model is proposed to predict stations. The article first uses the Node2vec algorithm to perform a biased random walk on the railway network to generate the vector representation of each station. Secondly, an adaptive method is proposed, which derives the critical value of edge stations through the pinching rule, and then effectively identifies the edge stations in the high-speed railway network. Next, the calculation method of Hadamard product is used to represent the potential neighbors of edge sites, and then the attention mechanism is used to predict the link between all potential neighbors and their corresponding edge sites. After the link prediction, the final high-speed railway network is obtained, and it is input into the GNN classifier together with the line label to complete the station prediction. Experiments show that: Baoji and Hanzhong are more likely to become node stations in this north–south railway trunk line. The Scatter-GNN classifier optimizes the site selection strategy by calculating the connection probabilities between two or more candidate routes and comparing their results. This may reduce manual selection costs and ease geographic evaluation burdens. The model proposed in this paper can be used as an auxiliary strategy for the traditional route planning scheme, which may become a new way of thinking to study such problems in the future.

Keywords: high-speed railway; route planning; site selection; graph neural network; scatter-GNN; link prediction

check for updates

Citation: Ma, M.; Zhang, Y.; Li, Y.; Li, X.; Liu, Y. Scatter-GNN: A Scatter Graph Neural Network for Prediction of High-Speed Railway Station—A Case Study of Yinchuan–Chongqing HSR. *Appl. Sci.* **2023**, *13*, 150. https://doi.org/10.3390/app13010150

Academic Editors: Maxim A. Dulebenets and Krzysztof Koszela

Received: 2 November 2022
Revised: 13 December 2022
Accepted: 20 December 2022
Published: 22 December 2022

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Railway transportation is one of the most convenient, safest, most environmentally friendly and economically cheap transportation methods in the world [1–3]. It can not only promote the economic development of countries in the world and increase the exchanges between countries, but also help to enhance the overall image of the country and highlight the level of modern development [4–6]. At the same time, it also has a close connection and important influence on a country's regions and cities. At present, the demand for railway transportation in countries around the world continues to increase, resulting in a gradually dense railway network. Entering the era of high-speed rail, this influence has intensified

the flow speed and scale of the entire society, and promoted the rapid development of production. Some studies have shown that high-speed rail transportation has a greater impact on underdeveloped areas and areas far away from regional central cities [7,8]. Therefore, the development and reform commissions of various cities are actively striving for the entry of high-speed rail, which makes a reasonable station selection strategy in a new line extremely important. In addition, the complexity of the railway network also brings great challenges to traditional line planning and train operation scheduling [9–12]. This paper takes the Yinchuan–Chongqing high-speed railway (HSR) section as an example, expounds in detail the key significance of this line planning, and proposes a station selection method, which is applicable to the planning of all railway lines.

The Yinchuan–Chongqing high-speed railway is a north–south railway line starting from Yinchuan, the capital of Ningxia Autonomous Region, through Wuzhong and Guyuan, Pingliang of Gansu Province, Baoji of Shaanxi Province, Guangyuan, Nanchong of Sichuan Province, and Chongqing. This opens up the longitudinal dead-end road between the northwest and southwest regions, and forms a new trunk line from the Guanzhong-Tianshui Economic Zone to the Chengdu-Chongqing Economic Zone and the Yangtze River Economic Belt, which is conducive to promoting the coordinated economic development of Shaanxi-Gansu-Ningxia old revolutionary areas and Qinba mountainous areas. However, although the Yinchuan–Guyuan–Baoji–Hanzhong–Nachong section has long been expected to invest, the choice of the two parallel lines of Tianshui–Guangyuan and Baoji–Hanzhong is controversial, and has not yet been approved by the Development and Reform Commission.

Usually, the factors considered for the route selection scheme include population flow, environmental protection assessment, geological survey, project cost, and station route selection [13], etc., and the comprehensive route selection research method is used to select among many control factors through technical and economic comparison [14]. Moreover, the balance of interests to choose a suitable route that can meet the interests of all parties. For example, literatures [15,16] proposed that biodiversity is an important aspect of protecting the ecological environment. Routes that affect the development of ecosystems and biodiversity are therefore rejected during environmental impact assessments during development site selection. In [17], a geographic information system (GIS)-based network analysis and analytic hierarchy process (AHP) approach created a new hybrid route considering economic and environmental criteria and it was compared with three different routes from pre-construction studies. These traditional methods usually only complete the first step for route planning. Most of them expounded the objective influencing factors on route selection, but did not formulate a reasonable route selection strategy. Moreover, in many practical situations, this kind of evaluation index can only filter out the general route of travel, but no in-depth research has been carried out on the specific cities or stations to be passed. Numerous studies have shown that GNN's excellent ability to deal with unstructured data has made new breakthroughs in data analysis, recommendation systems, physical modeling, natural language processing, and graph combination optimization problems. Therefore, it can also be applied to site prediction of traditional high-speed rail networks. However, the number of high-speed railway lines in the eastern region of China far exceeds that in the western region. Analogous to the network on the graph, the paper defines this line structure as an edge graph structure network. Among them, the edge graph structure refers to the existence of some edge nodes in the graph, that is, the unbalanced structure formed by the degree of some nodes in the graph is much smaller than other nodes. This indicates that some sites in the network of high-speed rail lines have much larger degrees than others. For example, Beijing station has far more adjacent stations than Bijie station. In the case of extremely unbalanced site degree, directly training the GNN classifier makes the vector representation ability of the site with more neighbors stronger, and the loss function calculation is accurate. The sites with fewer neighbors are marginalized, and the vector representation capabilities of these edge sites are relatively insufficient, and the loss calculation error is large. For edge sites, their neighbors are

much more important than those of large sites, and the neighbors of edge sites are always under-sampled compared with other sites during the vector aggregation process, which leads to suboptimal prediction performance.

In order to make up for the shortcomings of the above research methods, this paper uses the graph neural network [18–23] to model the main national high-speed railway line network, aiming at predicting the passing stations of the Yinchuan–Chongqing high-speed railway line. The model first uses the Node2vec algorithm to generate the vector representation of each high-speed rail station, and then proposes an adaptive function to effectively identify the edge stations in the high-speed railway network, and then uses the graph attention mechanism [20] to complete the potential neighbors of the edge stations.

This ensures that during the embedding process of each site, the difference in the number of their neighbor sites is as small as possible. Finally, the connection probability between possible sites is calculated. It should be noted that due to the extremely limited number of railway stations, the prediction effect of the graph deep learning model has certain limitations. The scatter graph neural network (Scatter-GNN) model proposed in this paper is only used as an auxiliary strategy for the traditional route planning scheme, which may become a new way of thinking to study such problems in the future research. The main results of the article are summarized as follows:

(1) The Scatter-GNN model was used to model the main national high-speed railway lines, and calculate the priority of two possible planned lines in the Yinchuan–Chongqing high-speed railway, which may become a new scenario for applying the GNN classifier;
(2) Compared with the graph representation learning method GAT, the Scatter-GNN model improves the accuracy by 0.03%.

2. Related Work

2.1. High-Speed Railway Line Planning Based on Mathematical Model

At present, high-speed railway line planning based on mathematical methods can be roughly divided into two categories: one is the heuristic algorithm based on multi-objective optimization, and the other is the algorithm based on the line pool. For example, the literatures [24,25] established a multi-objective linear programming model of statistical optimization, and predicted line direction by means of fuzzy mathematical programming. The literature [26] proposes four different objective functions, considering the cost of all transportation systems as well as external costs, and proposes heuristic and meta-heuristic solving algorithms to evaluate line planning strategies. Although this type of method can provide a qualitative judgment on line selection. However, in practice, if the planned route is long, the multi-step reasoning significantly reduces the confidence, which is obviously not conducive to the design of all routes. The literature [27] proposes an iterative method combined with a Mixed Integer Linear Programming (MILP) model for maximizing operator profits during route planning. The iterative approach aims to optimize the frequency settings of the route based on the expected route passengers will take. Although this type of method provides specific solutions, there is no corresponding planned route as a verification, and the universality is not high. Most of the model parameters are based on ideal settings, which are difficult to apply to specific route planning. Another type is the planning scheme based on the line pool. The literature [28–30] has studied the optimization method of train operation plan with the shortest travel time and the least amount of passenger transfers. Based on the idea of alternative route planning, the shortest path selection behavior of passengers in the "line break" network is described. This kind of method has a single level of consideration, and the railway transportation cost and geographical environment assessment needs to be considered. The literature [31] studied the HSR line planning theory based on line pools. It establishes some reasonable criteria for all trains that may be included in the set and combines the route planning features with the multi-commodity flow problem to establish an integer programming model and a nonlinear mixed integer programming model, using Lagrangian relaxation inspiration

algorithm to solve the model. Although this type of method takes into account the actual situation of passengers traveling by car, the amount of statistical data is large, and the number of passengers is time-varying. If it is not combined with other strategies, it lacks accuracy in practical applications.

2.2. High-Speed Railway Line Planning Based on Machine Learning Model

The route planning of a high-speed railway network based on a machine learning model provides a better solution to the time-varying demand problem. This method can be roughly divided into two categories: one is the route planning scheme based on branch search strategy, and the other is the route planning scheme based on convolutional neural network. For example, the literature [32] built a two-level planning model based on Stackelberg game theory, incorporated passenger flow distribution into route planning, and obtained cost- and customer-oriented route planning. It was expanded by introducing an "estimated start time" to capture timing information and assess the degree of fit between route plans and travel demand. Although this type of model has strong robustness, the convergence speed is slow, the running time is too long, and the performance of the algorithm mostly depends on the initial value, which leads to low efficiency in actual route planning. The literature [33] proposed a two-layer optimization model within a simulation framework to deal with the high-speed railway route planning problem. In this model, the top layer is designed to implement an optimal set of stopping schedules with service frequencies, and is formulated as a nonlinear program, which is then solved by a genetic algorithm. The above-mentioned methods are all scheduling schemes designed for high-speed trains to avoid space–time conflicts between trains after running. They are all based on the existing lines and dynamically adjusted on this basis of a midway emergency strategy. However, that does not apply to predictions for new lines. Another category is the planning scheme based on convolutional neural network. For example, [34] utilizes a temporal graph convolutional network (T-GCN) for circuit planning, which combines a graph convolutional network (GCN) and a gated recurrent unit (GRU). GCNs are used to learn complex topologies to capture spatial correlations, while gated recurrent units are used to learn the dynamics of traffic data to capture temporal correlations. This type of method can be applied to ordinary network structures. However, for a huge high-speed railway network, with the gradual increase in model stations and routes, the non-parallel computing problem of GRU is magnified. At the same time, it cannot solve the problem of gradient disappearance faced by the model, which makes the prediction results unsatisfactory. The literature [35] proposed a network architecture consisting of a deep trust network (DBN) and a regression model, and verified that the network can capture random features from multiple traffic data. However, this method is currently only in theoretical research and has not been applied to actual route planning. Although the models in the above literature are highly time-varying, they are not suitable for unplanned sites. In this paper, the Scatter-GNN model is used to predict the passing stations of the Yinchuan–Chongqing high-speed railway line, which may become a new idea for studying such problems in the future.

2.3. Graph Neural Network Model Based on Edge Graph Structure

A series of studies [36–38] have confirmed that the performance of a classifier is mainly determined by the majority class [39]. However, there is another aspect of imbalance in graph-structured data, which is the imbalance of topological structure on the graph, which is also called edge graph structure. Considering the generality of this problem, the literature [40] proposed a ReNode framework for solving the problem of edge graph structure. It first reweights nodes based on how close each labeled node is to its class boundary [41–44]. The weights of training nodes that are close to the category boundary and are likely to cause decision boundary shifts are reduced, while the weights of training nodes close to the center of the category are increased. This makes the node's influence boundary and the real category boundary more coincident, reducing the decision boundary

offset problem caused by the imbalanced topology. However, this type of method needs to measure the distance from the node to the class boundary. This index has no specific value and is not easy to control. Moreover, the model has poor portability and is difficult to apply to graph-related tasks other than node classification. The literature [45] noted that although graph neural networks can learn node representations, they treat all nodes uniformly, without paying attention to a large number of tail nodes, which have less structural information, resulting in poor performance. It is based on the assumption that all nodes have similar labels to their neighbors, and use the transformation operation to transfer the head node to the tail node to simulate the missing neighborhood information. By modeling the relationship between a node and its neighbors, the transformation vector is finally used to calculate the missing neighborhood expressions of all tail nodes, so as to improve the robustness of tail node embedding. Although this method completes the neighbors of the long-tail nodes in the graph, it ignores that the real graph data do not all follow the long-tail distribution, and there are some nodes that fail to reach the transfer index and are ignored, which is obviously not conducive to feature representations for all nodes. This paper proposes an adaptive method to discover all edge nodes in the graph, which may better play the performance of the GNN classifier.

3. Route Planning Scheme Based on Scatter-GNN

3.1. Data Selection

This paper selects the "four vertical and four horizontal" and "eight vertical and eight horizontal" passenger routes and some intercity railways in the national "medium and long-term railway network planning" of the Chinese Ministry of Railways as the basic lines of the model. The paper focuses on collecting 137 national hub stations and regional node stations. The Table 1 below shows some of the line data for some stations in the high-speed rail network. It is worth noting that if a city has multiple high-speed rail stations, all station names are recorded as the city name. In addition, only the station names of prefecture-level cities and above are recorded in the table, and other stations are not displayed. Statements regarding input data requirements are listed in Appendix A.

Table 1. Example of datasets.

Order	Route
1	Beijing–Tianjin–Jinan–Xuzhou–Bengbu–Nanjing–Shanghai
2	Beijing–Shijiazhuang–Zhengzhou–Wuhan–Changsha–Guangzhou–Shenzhen–Jiulong
3	Beijing–Chengde–Chaoyang–Fuxin–Shenyang–Tieling–Siping–Changchun–Harbin
4	Shenyang–Anshan–Yingkou–Dalian
5	Hangzhou–Ningbo–Taizhou–Wenzhou–Fuzhou–Xiamen–Shenzhen
6	Nanjing–Hefei–Wuhan–Chongqing–Chengdu
7	Lianyungang–Xuzhou–Shangqiu–Zhengzhou–Luoyang–Xi'an–Baoji–Lanzhou–Xining–Wulumuqi
8	Shanghai–Hangzhou–Nanchang–Changsha–Guiyang–Kunming
9	Qingdao–Jinan–Dezhou–Shijiazhuang–Taiyuan
...	...

3.2. Problem Definition

Since the number of high-speed railway lines in eastern China far exceeds that in western China, this paper defines this railway network as an edge graph structure network. We propose a method for solving station prediction in edge graph networks. The edge graph refers to the existence of some edge nodes in the graph. The specific definition is as follows: Given a railway network G, the number of stations V in G is n, the connecting edges between two stations are called routes, and E is the number of routes. Then when G is neither fully connected nor empty, there are:

$$0 < E < \frac{n(n-1)}{2} \tag{1}$$

where a possible value of E is $nf(n)$, then the edge site of network G can be determined as:

$$V = F(e) \tag{2}$$

where e represents the number of neighbors around any site, and F is a logistic function that represents the true state of the result when e satisfies the following conditions:

$$F(e) = \begin{cases} 1, e \leq f(n) \\ 0, e > f(n) \end{cases} \tag{3}$$

the f in the above formula is an adaptive function, denoted as the mapping of f to F:

$$f \to F \tag{4}$$

when n tends to infinity, if the value of V is only related to n, and F' indicates that the node degree is less than the independent variable, the edge graph node is finally defined as:

$$V = F'(f(n)) \tag{5}$$

In Section 3.3, this paper focuses on deriving the solution process of the adaptive function. The direction of some high-speed railway lines and main stations can be described as shown in Figure 1 below. Among them, the black dotted line connects the possible passing stations of the Yinchuan–Chongqing high-speed railway. The black solid line represents the line segment that this paper focuses on, and the gray lines are other lines that actually exist.

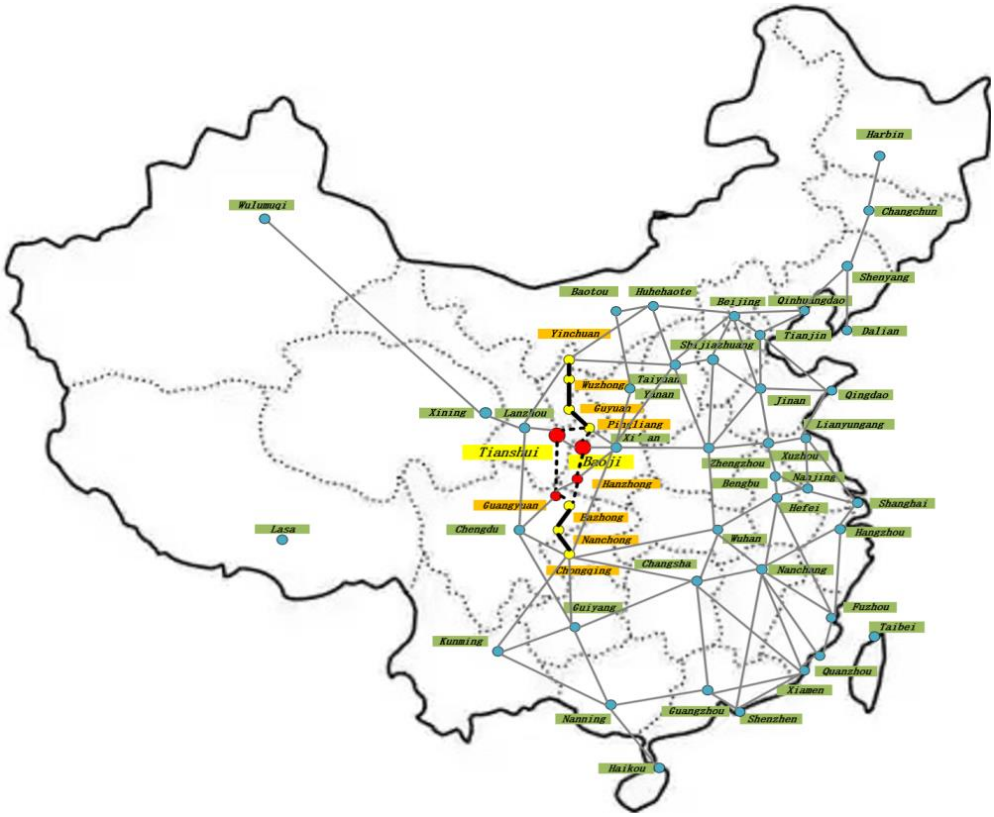


Figure 1. Schematic diagram of the forecast of the Yinchuan–Chongqing high-speed railway (HSR) station.

3.3. Adaptive Function Calculation and Edge Site Location Search

When the number of stations n tends to infinity, this paper makes the following derivation for the possible existence interval of the number of routes E:

$$0 < \frac{(1+n)\sqrt{(n+m)}}{2} < \frac{n(n-1)}{2} \tag{6}$$

where m is an arbitrary constant, then the possible values of E are the left and right neighborhoods of $\frac{(1+n)\sqrt{(n+m)}}{2}$. Further, using the scaling method has:

$$\frac{n(1+n)\sqrt{n}}{2\sqrt{(n^2+\frac{n}{2})}} < \frac{(1+n)\sqrt{n}}{2} < \frac{(1+n)\sqrt{(n+m)}}{2} < \frac{(1+n)\sqrt{(n+m^2)}}{2} < \frac{n\sqrt{n}\sum_{n=1}^{n}\frac{n}{m^2+n}}{\sum_{n=1}^{n}\frac{1}{\sqrt{m^2+n}}} \tag{7}$$

The above formula continues to divide the possible existence interval of the number of edges E, and its possible values are divided into six areas at this time.

For formula $\frac{n(1+n)\sqrt{n}}{2\sqrt{(n^2+\frac{n}{2})}}$ deformation, we obtain:

$$\frac{n(1+n)\sqrt{n}}{2\sqrt{(n^2+\frac{n}{2})}} = \frac{n^2(1+n)\sqrt{n}}{2(n^2+\frac{n}{2})} * \frac{\sqrt{n^2+\frac{n}{2}}}{n} = \frac{n\sqrt{n}\cdot\frac{n(1+n)}{2(n^2+\frac{n}{2})}}{\frac{n}{\sqrt{(n^2+\frac{n}{2})}}} \tag{8}$$

when $n \to \infty$:

$$\lim_{n\to\infty}\frac{n(1+n)}{2(n^2+\frac{n}{2})} = \frac{1}{2}$$

$$\lim_{n\to\infty}\frac{n}{\sqrt{(n^2+\frac{n}{2})}} = 1$$

therefore:

$$\lim_{n\to\infty}\frac{n(1+n)\sqrt{n}}{2\sqrt{(n^2+\frac{n}{2})}} = \frac{n\sqrt{n}}{2} \tag{9}$$

expand the formula $\sum_{n=1}^{n}\frac{n}{m^2+n}$ and $\sum_{n=1}^{n}\frac{1}{\sqrt{m^2+n}}$, let $m = n$ obtain:

$$\sum_{n=1}^{n}\frac{n}{m^2+n} = \frac{1}{n^2+1} + \frac{2}{n^2+2} + \cdots + \frac{n}{n^2+n} \tag{10}$$

$$\sum_{n=1}^{n}\frac{1}{\sqrt{m^2+n}} = \frac{1}{\sqrt{n^2+1}} + \frac{1}{\sqrt{n^2+2}} + \cdots + \frac{1}{\sqrt{n^2+n}} \tag{11}$$

when $n \to \infty$:

$$\lim_{n\to\infty}\frac{n(1+n)}{2(n^2+n)} < \lim_{n\to\infty}\sum_{n=1}^{n}\frac{n}{m^2+n} < \lim_{n\to\infty}\frac{n(1+n)}{2(n^2+1)}$$

$$\lim_{n\to\infty}\frac{n}{\sqrt{(n^2+n)}} < \lim_{n\to\infty}\sum_{n=1}^{n}\frac{1}{\sqrt{m^2+n}} < \lim_{n\to\infty}\frac{n}{\sqrt{(n^2+1)}}$$

and:

$$\lim_{n\to\infty}\frac{n(1+n)}{2(n^2+n)} = \lim_{n\to\infty}\frac{n(1+n)}{2(n^2+1)} = \frac{1}{2}$$

$$\lim_{n\to\infty}\frac{n}{\sqrt{(n^2+n)}} = \lim_{n\to\infty}\frac{n}{\sqrt{(n^2+1)}} = 1$$

from the pinch theorem we obtain:

$$\lim_{n\to\infty}\frac{n(1+n)\sqrt{n}}{2\sqrt{\left(n^2+\frac{n}{2}\right)}}=\lim_{n\to\infty}\frac{n\sqrt{n}\sum_{n=1}^{n}\frac{n}{m^2+n}}{\sum_{n=1}^{n}\frac{1}{\sqrt{m^2+n}}}=\lim_{n\to\infty}nf(n)=\frac{n\sqrt{n}}{2} \tag{12}$$

that is when $n\to\infty$:

$$f(n)=\frac{\sqrt{n}}{2} \tag{13}$$

The above formula shows that when the number of stations in the network is large enough, the sum of the number of all routes tend to $\frac{n\sqrt{n}}{2}$. Then site V in the network is defined as an edge site as:

$$V=F'\left(\frac{\sqrt{n}}{2}\right) \tag{14}$$

Substitute 137 stations in the high-speed railway line into this function to calculate: $f(137)=\frac{\sqrt{137}}{2}<6$. Then the edge site is recorded as a site whose number of adjacent sites is less than 6. The process of location search and potential neighbor feature representation is described as the Algorithm 1:

Algorithm 1: Calculate Undirected Node Degree and Neighbor Vector.

Require:
import networkx as nx
Graph $G=(V,E)$
Ensure:
G = nx.random_graphs.barabasi_albert_graph (Graph)
Return G.degree(g.degree < $\frac{\sqrt{n}}{2}$)
For n = 1, 2, . . . ,N **do**
    Generate neighborhood node lists N_1, N_2, . . . , N_K
    for k = 1, 2, . . . , K **do**
        $\vec{v'}=\vec{v_n}\odot\vec{N_k}$
        g.add_edges_from(v',N_k)
output $G=(V,E)$

Let the vector representation of edge site V be denoted as $\vec{v}$, and its neighbor sites are: V_i, V_j, V_k, etc. Then the potential neighbor features of V can be expressed as:

$$\begin{aligned}\vec{v'}&=\vec{v}\odot\vec{v_i}\\ \vec{v''}&=\vec{v}\odot\vec{v_j}\\ \vec{v'''}&=\vec{v}\odot\vec{v_k}\\ &\cdots\cdots\end{aligned} \tag{15}$$

In order to prevent the cold-start [46–50] problem of link prediction, this paper connects the obtained potential neighbor sites with the neighbors of their corresponding edge sites. This is because the vector obtained after the Hadamard product represents the edge information between stations, which is associated with any station. This paper assumes that each potential neighbor site has only one neighbor, and uses this to predict the possibility of edge connections between it and the edge site. In addition, if the neighbors of the edge site are edge sites, then the obtained potential neighbor sites are respectively connected to the two. Appendix B gives explanations about some special mathematical symbols in the paper.

Figure 2 below shows an edge graph structure. The red part represents the edge site, the blue parts represent other sites, and the dotted line part represents the potential neighbor site of the edge site. A solid black line connects a potential neighbor site with its

corresponding edge site neighbors. Dashed lines indicate whether there is a route between the two stations that are predicted. If it exists, a link is added between them and the corresponding black link is deleted, otherwise its potential neighbors are deleted.

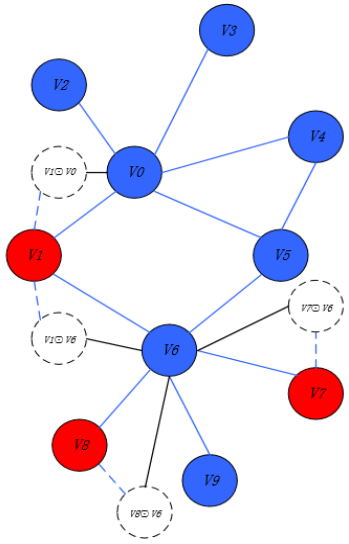


Figure 2. An example of potential neighbors of a site in the edge graph.

3.4. Completion of Edge Site Neighbors

In this paper, the graph attention mechanism is used to predict the link between each edge site and their corresponding potential neighbors, and then a new line network is generated. The completion process of edge site neighbors is shown in Figure 3 below:

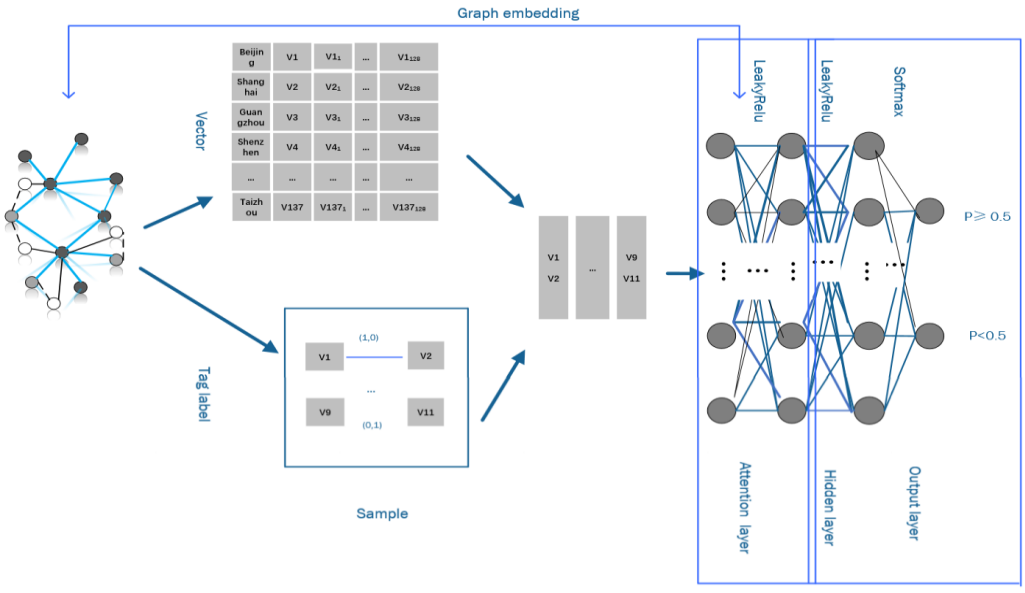


Figure 3. Edge site neighbor completion process.

In this paper, the Node2vec algorithm is used to generate the eigenvectors of each station in the railway network. The specific description is as follows:

Given the current site v, the probability of visiting the next site x is:

$$P(c_i = x | c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z}, & if(v,x) \in E \\ 0, & otherwise \end{cases} \quad (16)$$

where π_{vx} is the unnormalized transition probability between site v and site x. Z is a normalization constant.

Node2vec introduces two hyperparameters p and q to control the strategy of random walk, assuming that the current random walk reaches site v after passing edge (t, v). Let π_{vx} be the weight of the edge between site v and x:

$$\pi_{vx} = \alpha_{pq}(t, x) \cdot w_{vx} \tag{17}$$

$$\alpha_{pq}(t, x) = \begin{cases} \frac{1}{p}, & \text{if } d_{tx} = 0 \\ 1, & \text{if } d_{tx} = 1 \\ \frac{1}{q}, & \text{if } d_{tx} = 2 \end{cases} \tag{18}$$

where d_{tx} is the shortest path distance between site t and site x. The effect of hyperparameters p and q on the walk policy is discussed below. The parameter p controls the probability of repeatedly visiting the site just visited. If the value of p is larger, the probability of visiting the site just visited becomes smaller, and vice versa. q controls whether the walk goes outward or inward. If $q > 1$, the random walk tends to visit sites close to t (biasing BFS); if $q < 1$, the random walk tends to visit sites far from t (biasing DFS). After the sequence sampling is completed, the algorithm learns the vector representation of each site according to Word2vec. The essence of Word2vec is to use contextual information to train neural networks. It mainly has two implementation methods, namely continuous bag of words (CBOWs) and Skip-grams. CBOW trains the neural network by predicting a word from contextual information, while Skip-grams trains the neural network by predicting the context given a word. Analogy in the graph structure, its site vector learning process is described as the Algorithm 2:

Algorithm 2: Learning Vector Representation.

Require:
Graph $G = (V, E, W)$, Dimensions d, Walks per node r. Walk length l, Context size k, Return p, In-out q
π = PreprocessModifiedWeights (G, p, q)
$G' = (V, E, \pi)$
Initialize walks to Empty
Ensure:
1. For iter = 1 to r do
2. for all nodes u ∈ V do
3. walk = node2vecWalk(G', u, l)
4. Append walk to walks
5. f = StochasticGradientDescent(k, d, walks)
6. Return f

Secondly, all routes in the network are added to the sample space as positive samples, and the same number of station pairs without routes are randomly selected and added to the sample space as negative samples; then, station pairs and their site vector representation is sent to the attention layer, and the input of each pair of sites is represented by a vector as:

$$\vec{v} = \left\{ \vec{v}_1, \vec{v}_2, \vec{v}_3, \ldots, \vec{v}_n \right\}, \ \vec{v}_i \in R^F \tag{19}$$

A new vector representation is obtained by aggregating the neighbor site information of each site:

$$\vec{v}'_i = \sigma\left(\sum_{j \in N_i} \alpha_{ij} W \vec{v}_j \right) \tag{20}$$

where W is the training weight, and site j is the first-order neighbor of site i. α_{ij} represents the ratio of the attention coefficient [10] of site j to site i to the attention coefficients of all neighbors of site i to site i. Among them, the attention coefficient e_{ij} can be defined as:

$$e_{ij} = F\left(W\vec{v}_i, W\vec{v}_j\right) \tag{21}$$

It represents the importance of site j to site i, regardless of graph structure information, and F is a self-defined function. In order to make the attention coefficient easy to understand and easy to compare, this paper introduces a softmax function to regularize all adjacent sites j of site i:

$$\alpha_{ij} = softmax_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k\in N_i}\exp(e_{ik})} \tag{22}$$

After adding the LeakyRelu function and substituting it into formula (21), the complete attention mechanism is expressed as follows:

$$\alpha_{ij} = \frac{\exp(\text{LeakyRelu}(\vec{F}^T\left[W\vec{v}_i||W\vec{v}_j\right]))}{\sum_{k\in N_i}\exp(\text{LeakyRelu}(\vec{F}^T\left[W\vec{v}_i||W\vec{v}_k\right]))} \tag{23}$$

where α_{ij} is called the normalized attention coefficient. The vector representation of site i changes from the original $\vec{v}_i$ to $\vec{v}'_i$ after passing through the attention layer, and then uses the multi-head attention mechanism to repeat the process K times to obtain the following formula:

$$\vec{v}'_i = \Big\Vert_{k=1}^{K}\sigma\left(\sum_{j\in N_i}\alpha_{ij}^k W^k\vec{v}_j\right) \tag{24}$$

where K represents the stacking layers of the attention layer, $||$ represents the row vector splicing operation, and N_i is the first-order neighbor set of the site $i \in V$. The Kth attention mechanism in the multi-head attention mechanism is α^k, and W^k is the linear transformation weight matrix of the input features under the Kth attention mechanism. The focus of multi-head attention is different, which makes the model compatibility better.

Finally, in order to make the prediction effect of the model more stable, the splicing vector is K-averaged to obtain the final feature of each site $\vec{v}'_i$, $\vec{v}'_i$ indicates that the output feature of each site is related to all its adjacent features, which is obtained after activation of their linear and nonlinear functions.

$$\vec{v}'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in N_i}\alpha_{ij}^k W^k\vec{v}_j\right) \tag{25}$$

Finally, the new vector representation for each pair of sites is:

$$\vec{v}' = \left\{\vec{v}'_1, \vec{v}'_2, \vec{v}'_3, \ldots, \vec{v}'_n\right\},\ \vec{v}'_i \in R^{F'} \tag{26}$$

After the aggregation operation of the K-layer graph attention mechanism, each pair of site vectors is represented by the Hadamard product. Taking sites i and j as an example, the route is expressed as follows:

$$\vec{e}_{ij} = \vec{v}'_i \odot \vec{v}'_j \tag{27}$$

Finally, the probability of the existence of a route between two stations can be calculated by activating by the following formula:

$$P_{ij} = Softmax\left(W\vec{e}_{ij} + b\right) \tag{28}$$

A binary cross-entropy loss function is used as supervised training:

$$\mathcal{L}_{cross\ entropy} = -\frac{1}{|\mathcal{P} \cup \mathcal{N}|} \sum_{(i,j) \in |\mathcal{P} \cup \mathcal{N}|} e_{ij} \log P_{ij=1} + (1 - e_{ij}) \log P_{ij=0} \tag{29}$$

where $|\mathcal{P} \cup \mathcal{N}|$ represents the sum of the number of positive samples and the number of negative samples in the sample space; if there is a route between stations i and j, then $e_{ij} = 1$, otherwise $e_{ij} = 0$; $P_{ij=1}$ represents the probability that the route e_{ij} exists in the line, $P_{ij=0}$ represents the probability that the route e_{ij} does not exist in the line. The operation of the link prediction process is described as the Algorithm 3:

Algorithm 3: Link Prediction.

Graph $G = (V, E, W)$
Forward propagation:
Def forward(self,h,adj):
calculate the attention coefficient e,adjacent nodes and obtain:
$\vec{v}'_i = \sigma(\frac{1}{K} \sum_{k=1}^{K} \sum_{n \in N_i} \alpha_{in}^k W^k \vec{v}_n)$
$\vec{v}'_j = \sigma(\frac{1}{K} \sum_{k=1}^{K} \sum_{n \in N_j} \alpha_{jn}^k W^k \vec{v}_n)$
if self.concats is False
output layer does not perform linear transformation
Mosaic feature vector:
Def _prepare_attentional_mechanism_input(self, W·h):
For eigenvector, characteristic matrix in n **do**
all_combinations_matrix=w·h_...
return all_combinations_matrix.view
Attention Networks:
Def Link prediction(epoch,loss,P)
For i = 1 to epoch **do**
for edge = 1 to N **do**
Get node p and q by dividing edge:
$\vec{e_{ij}} = \vec{v}'_i \odot \vec{v}'_j$
for k=1 layer to K layer **do**
calculate loss = $-\frac{1}{|P \cup N|} \sum_{(i,j) \in |P \cup N|} e_{ij} \log P_{ij=1} + \left(1 - e_{ij}\right) \log P_{ij=0}$ and gradient
update parameter w and b
save model
Return P
Prediction:
For n = 1, 2, . . . ,N **do**
Generate neighborhood node lists N_1, N_2, . . . , N_K
for k = 1, 2, . . . , K **do**
input $\vec{v_n} \odot \vec{v'_k}$ to model
if $p \geq 0.5$
g.add_edges_from(v_n,v'_k)
g.delete_edges_from(v'_k,N_k)
output $G = (V, E)$

After the model training is over, this paper adds all routes whose existence probability is not less than 0.5 obtained through link prediction to the original network to construct a new route network.

3.5. Calculation of Existing Route Probability Based on Scatter-GNN

After completing the neighbors of the edge site to obtain a new line map, this paper uses the GNN classifier to calculate the probability of the existence of the two lines of

Pingliang–Tianshui–Guangyuan and Pingliang–Baoji–Hanzhong, respectively. The flow chart of the Scatter-GNN model is shown in Figure 4 below:

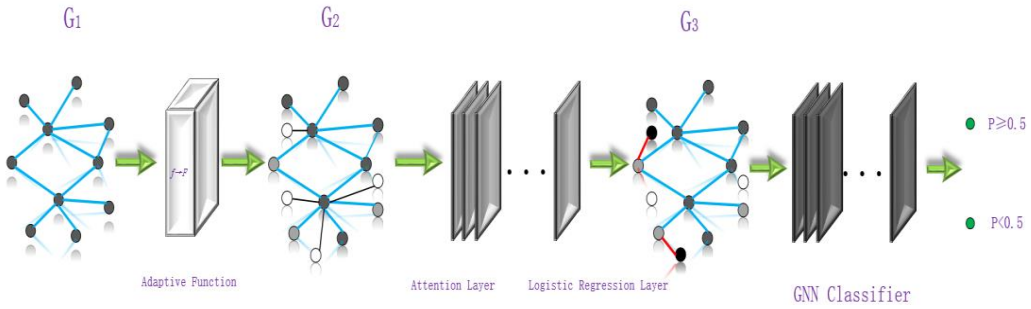


Figure 4. Flowchart of Scatter-GNN model.

Overall, this paper first defines the high-speed railway network as an edge graph structure network. Firstly, the position of the edge site is obtained after the calculation of the adaptive function. Then, the vector representation of the potential neighbors of the edge site is obtained by using the calculation method of the Hadamard product. Then through the attention layer and the logistic regression layer, the neighbors of the edge sites are completed. The way of completion is to predict the links between each edge site and its potential neighbors. Finally, the new line network is predicted by the GNN classifier to obtain the probability of the existence of the target line. The specific process can be described as the Algorithm 4:

Algorithm 4: Scatter-GNN.

Require:
Graph $G = (V, E, W)$,
Node Features Matrix F,
Neighborhood Sample Layer Num K,
Each Layer Neighborhood Sample Size $S_1, S_2, \ldots, S_K$,
Learning Rate for Adam: lr

Ensure:
Updated Node Features Matrix F

1. Initialize the parameters in W, b
2. **while** not converge **do**
3. **for** i = 1, 2, . . . , N **do**
4. Generate neighborhood node lists $N_1, N_2, \ldots, N_K$ by K layer neighborhood sampling according to $S_1, S_2, \ldots, S_K$
5. for k = 1, 2, . . . , K **do**
6. $h_i^k = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in N_i}\alpha_{ij}^k W^k \vec{h_j}\right)$ or $\ldots, h_i^k \in F$
7. Update h_i^k in F
8. Calculate loss = $-\frac{1}{|P\cup N|}\sum_{(i,j)\in|P\cup N|} e_{ij}\log P_{ij=1} + \left(1-e_{ij}\right)\log P_{ij=0}$ and gradient of ∂W and ∂b by Adam
9. Update the parameters:
$W = W - lr * \partial W$
$b = b - lr * \partial b$
10. **Return** Updated F

Prediction:
11. input $\vec{v_m} \odot \vec{v_n}$ to model
12. **Return** P

4. Experimental Verification

In this paper, the Scatter-GNN model is applied to the high-speed railway network and compared with other recently proposed baselines. The final results show the predicted probabilities of the Scatter-GNN model for various possible planned routes.

4.1. Baseline Algorithm

In this paper, four baselines proposed in recent years are selected for comparison experiments with Scatter-GNN.

- DeepWalk: DeepWalk [51] obtains the node sequence by random walk in the graph, then uses the Word2vec algorithm to obtain the vector representation of the node, and finally uses the logistic regression classifier for prediction. The parameters of the logistic regression classifier are all consistent with the Scatter-GNN;
- Node2vec: Node2vec [52] is an improved algorithm for DeepWalk. It obtains the node sequence by performing a biased random walk in the graph, and then obtains the vector representation of the node through the Word2vec algorithm. Similar to the DeepWalk algorithm, the vector representation of the obtained node is input into the logistic regression classifier for prediction, and the classifier parameters are consistent with the Scatter-GNN;
- GCN: GCN [7] uses the convolution kernel to extract the structural features of the graph, and then combines the node features to train on the whole graph;
- GAT: GAT [8] enhances the vector representation of the target node by assigning different weights to each neighbor, and it is one of the most advanced GNN models in graph representation learning so far.

4.2. Parameter Setting

In order to facilitate comparative analysis, the dimension of the node vector generated by all algorithms in the experiment is set to 64, and other parameters follow the original setting values. The specific parameters are set as follows: Node2vec algorithm random walk step size is 40, random walk step number is 80, parameter b is 0.25, parameter d is 2, and the window size is 10. The GAT model has 32 attention layers and 64 hidden layer nodes. All GNN models have an Epoch of 100 and a learning rate of 1×10^{-5}. All module parameters in the Scatter-GNN model are consistent with the GAT model. All algorithms are trained on 90% of the data and tested on 10% of the data.

4.3. Experimental Analysis

Classification accuracy (ACC) and F1-Score are important indicators for evaluating the performance of classifiers, which have been widely used in the industry in recent years. Therefore, this paper uses these two evaluation indicators to evaluate the model. For processing edge graph structures, the Macro-F metric is more comprehensive in evaluating the model.

Table 2 presents the experimental results of some commonly used baseline algorithms and Scatter-GNN models on this dataset. As can be seen from the table, the traditional graph embedding method has a large gap in prediction performance compared with the deep learning model. However, Scatter-GNN and other graph neural network models have no significant gap in accuracy and F1-Score. This is because the model has always been based on the assumption of the number of sites $n \to \infty$ in the calculation of the adaptive function. Moreover, when the number of sites is small, the function value is small, the number of edge sites is generally small, and the number of potential neighbors is also small. This results in a small difference between the completed line network structure and the original network, which can be approximately treated as the same type of GNN model. Nevertheless, Scatter-GNN has an improvement in accuracy compared with the GAT model.

Table 2. Evaluation on edge graph link prediction (%) using GNN as the base model.

Methods	ACC	Macro-F
DeepWalk	47.6003	43.0754
Node2vec	48.4542	45.2780
GCN	57.35 ± 1.4	54.02 ± 1.2
Scatter-GCN	57.49 ± 1.3	54.05 ± 1.2
GAT	60.21 ± 1.4	58.18 ± 1.6
Scatter-GAT	60.24 ± 1.5	58.20 ± 1.6

In addition, this paper also uses the GNN model to calculate the existence probability of Pingliang–Tianshui–Guangyuan and Pingliang–Baoji–Hanzhong lines, as shown in Figure 5 below:

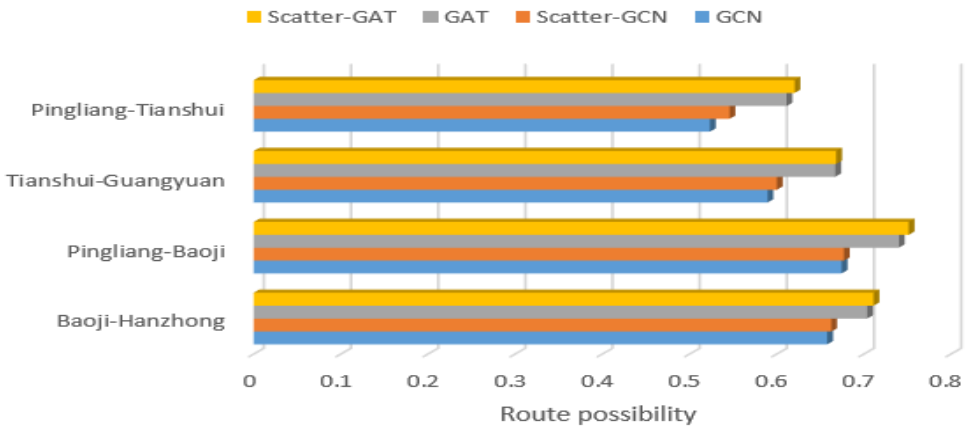


Figure 5. Line existence probability statistics.

The Scatter-GNN classifier calculates the connection probabilities between two or more candidate lines and compares their results. Several routes with higher total probabilities are included in the route planning to improve the site selection strategy.

As can be seen from the figure above, there is a certain gap in the probability calculation values of the four routes for various GNN models. However, their judgments on the final result are completely consistent. Among them, the Scatter-GAT model is used to calculate the existence probabilities of the four routes of Pingliang–Tianshui, Tianshui–Guangyuan, Pingliang–Baoji, Baoji–Hanzhong are about 62.06%, 66.84%, 75.15%, and 71.11%, respectively. This also shows that Baoji and Hanzhong are more likely to become node stations in this north–south railway trunk line. In the same way, this method is also applicable to the planning of other high-speed rail lines. With the gradual increase in known routes, the prediction of the model is more convincing.

5. Conclusions

In this paper, a graph neural network is used as the basic model, and a classification model for solving the edge network structure is proposed. China's high-speed railway line network is a typical edge graph structure network, so the model proposed in this paper can be applied to traditional line planning. Experiments show that the line of Pingliang–Baoji–Hanzhong section has a higher probability of existence, and it is more likely to become a node station on this north–south railway trunk line. Due to the extremely limited number of railway stations, the prediction accuracy of the graph deep learning model is not high. In addition, the calculation of the adaptive function of the model is based on the assumption that the number of sites $n \to \infty$ makes Scatter-GNN and other graph neural network models have no significant difference in accuracy and F1-Score. Therefore, the Scatter-GNN model proposed in this paper is only used as an auxiliary strategy for traditional route

planning schemes. Nevertheless, it shows some improvement in accuracy compared with the GAT model.

With the continuous increase in high-speed railway lines in China, the adjacent stations of each station continue to increase, and the prediction of the Scatter-GNN model is more accurate. It can not only assist manual line selection in actual line planning, ease the burden of investigation, but also re-evaluate existing operating lines. In the end, it can be applied to other routing scenarios such as roads, flight lines, etc. This is also likely to become a new way of thinking for future research on line planning, design, construction, and other similar issues. At the same time, the Scatter-GNN model also has the following limitations in practical applications:

(1) Since the Scatter-GNN model needs to constantly judge whether to add sites and routes before formal training, this makes it less efficient than traditional GNN classifiers;
(2) Due to the extremely limited number of railway stations, the prediction accuracy of the graph deep learning model is not high;
(3) The adaptive function defined in the Scatter-GNN model is based on the assumption $n \to \infty$. However, the stations in the railway line network eventually become saturated, which limits the model prediction performance.

In view of the above deficiencies, this paper intends to focus on the following issues in future research:

(1) How to divide the scope of the adaptive function more carefully, so as to further improve the prediction performance of the model;
(2) Explore ways to limit the number of potential neighbor sites to improve prediction efficiency.

These will be a meaningful challenges for future research.

Author Contributions: Conceptualization, M.M.; methodology, M.M. and Y.Z.; software, M.M. and Y.Z.; validation, Y.Z.; formal analysis, M.M. and Y.Z.; investigation, M.M.; resources, M.M.; data curation, M.M.; writing—original draft preparation, M.M. and Y.Z.; writing—review and editing, M.M., Y.Z. and Y.L. (Yong Li); supervision, M.M. and Y.Z.; project administration, X.L. and Y.L. (Yiping Liu); funding acquisition, Y.L. (Yong Li). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (Grant no. 72161034, 61863032) and The Major Scientific Research Projects of Northwest Normal University (Grant no. NWNU-LKZD2021-06).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In Section 3.1, this paper makes the following statement on the input data requirements of the Scatter-GNN model:

1. The input datasets in the model are all structured data, which should be a graph data structure. For example, transportation network, Weibo social network, urban population map, epidemic transmission network, and terrain structure map, etc. They should all conform to the definition of graph: Graph = (V,E). Among them, node V represents a city, an individual, and a website. The edge E represents the relationship between them, such as the social relationship between known people, the existing route between roads or railways, the mutual concern between two sites, etc. If there is only a set of nodes but no edges, it does not meet the input requirements of the model. It should be noted that the existing set of connected edges should have practical significance.

2. The acceptable data format for the model is: tabular form (excel) or adjacency matrix form. After importing the data file, use Networkx to build all the data into a graph, and add nodes and edges to it respectively. The specific method can be found in Algorithm 1 in the paper.

Ultimately, the output of the model can be treated as a regression problem.

Appendix B

Table A1. Interpretation of special mathematical symbols in the paper.

Order	Symbol	Illustration
1	$\odot$	Multiply the corresponding positions of two vectors
2	σ	Ordinary activation function
3	exp	Exponential function based on the natural constant e
4	$\|\|$	Concatenation operation of two vectors
5	softmax	Activation functions in multi-classification and normalization
6	LeakyRelu	Leaky Rectified Linear Units, linear activation function
7	R^F	Vector space

References

1. Singh, P.; Elmi, Z.; Meriga, V.K.; Pasha, J.; Dulebenets, M.A. Internet of Things for sustainable railway transportation: Past, present, and future. *Clean. Logist. Supply Chain* **2022**, *4*, 100065. [CrossRef]
2. Khan, M.Z.; Khan, F.N. Estimating the demand for rail freight transport in Pakistan: A time series analysis. *J. Rail Transp. Plan. Manag.* **2020**, *14*, 100176. [CrossRef]
3. Beuthe, M.; Jourquin, B.; Geerts, J.-F.; Ha, C.K.N. Freight transportation demand elasticities: A geographic multimodal transportation network analysis. *Transp. Res. Part E Logist. Transp. Rev.* **2001**, *37*, 253–266. [CrossRef]
4. Atack, J.; Margo, R.A. The impact of access to rail transportation on agricultural improvement: The American Midwest as a test case, 1850–1860. *J. Transp. Land Use* **2011**, *4*, 5–18. [CrossRef]
5. Ghofrani, F.; He, Q.; Goverde, R.M.; Liu, X. Recent applications of big data analytics in railway transportation systems: A survey. *Transp. Res. Part C Emerg. Technol.* **2018**, *90*, 226–246. [CrossRef]
6. Gao, M.; Cong, J.; Xiao, J.; He, Q.; Li, S.; Wang, Y.; Yao, Y.; Chen, R.; Wang, P. Dynamic modeling and experimental investigation of self-powered sensor nodes for freight rail transport. *Appl. Energy* **2020**, *257*, 113969. [CrossRef]
7. Liang, Y.; Zhou, K.; Li, X.; Zhou, Z.; Sun, W.; Zeng, J. Effectiveness of high-speed railway on regional economic growth for less developed areas. *J. Transp. Geogr.* **2020**, *82*, 102621. [CrossRef]
8. Yin, M.; Bertolini, L.; Duan, J. The effects of the high-speed railway on urban development: International experience and potential implications for China. *Prog. Plan.* **2015**, *98*, 1–52. [CrossRef]
9. Bešinović, N. Resilience in railway transport systems: A literature review and research agenda. *Transp. Rev.* **2020**, *40*, 457–478. [CrossRef]
10. Singh, P.; Dulebenets, M.A.; Pasha, J.; Gonzalez, E.D.R.S.; Lau, Y.-Y.; Kampmann, R. Deployment of autonomous trains in rail transportation: Current trends and existing challenges. *IEEE Access* **2021**, *9*, 91427–91461. [CrossRef]
11. Singh, P.; Pasha, J.; Khorram-Manesh, A.; Goniewicz, K.; Roshani, A.; Dulebenets, M. A Holistic Analysis of Train-Vehicle Accidents at Highway-Rail Grade Crossings in Florida. *Sustainability* **2021**, *13*, 8842. [CrossRef]
12. Grechi, D.; Maggi, E. The importance of punctuality in rail transport service: An empirical investigation on the delay determinants. *Eur. Transp.-Trasp. Eur.* **2018**, *70*, 1–23.
13. Yan, C. Research on Route Planning of Xi'an to Hancheng Intercity Railway. *Railw. Stand. Des.* **2016**, *11*, 1562.
14. Assad, A.A. Modelling of rail networks: Toward a routing/makeup model. *Transp. Res. Part B Methodol.* **1980**, *14*, 101–114. [CrossRef]
15. Lee, S.D. Strategic environment assessment and biological diversity conservation in the Korean high-speed railway project. *J. Environ. Assess. Policy Manag.* **2005**, *7*, 287–298. [CrossRef]
16. He, G.; Lu, Y. Public protests against the Beijing–Shenyang high-speed railway in China. *Transp. Res. Part D Transp. Environ.* **2016**, *43*, 1–16. [CrossRef]
17. Yildirim, V.; Bediroglu, S. A geographic information system-based model for economical and eco-friendly high-speed railway route determination using analytic hierarchy process and least-cost-path analysis. *Expert Syst.* **2019**, *36*, e12376. [CrossRef]
18. Will, H.; Ying, Z.T. Inductive representation learning on large graphs. *NeurIPS* **2017**, *2*, 1024–1034.
19. Thomas, N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the ICLR, Toulon, France, 24–26 April 2017.
20. Petar, V.; Guillem, C.; Arantxa, C.; Adriana, R.; Pietro, L.; Yoshua, B. Graph attention networks. In Proceedings of the ICLR, Vancouver, BC, Canada, 30 April 30–May 2018.

21. Xu, K.; Hu, W.H.; Jure, L.; Stefanie, J. How powerful are graph neural networks? In Proceedings of the ICLR, New Orleans, LA, USA, 6–9 May 2019.
22. Liu, Z.M.; Fang, Y.; Liu, C.H.; Steven, C.H. Node-wise localization of graph neural networks. In Proceedings of the IJCAI, Montreal, QC, Canada, 19–27 August 2021.
23. Pei, H.B.; Wei, B.Z.; Kevin, C.; Chang, C.; Lei, Y.; Yang, B. Geom-GCN: Geometric graph convolutional networks. In Proceedings of the ICLR, Ababa, Ethiopia, 30 April 2020.
24. Chang, Y.H.; Yeh, C.H.; Shen, C.C. A multi objective model for passenger train services planning: Application to Taiwan's high-speed rail line. *Transp. Res. Part B* **2000**, *34*, 91–106. [CrossRef]
25. Yin, Y. Multi objective bilevel optimization for transportation planning and management problems. *J. Adv. Transp.* **2002**, *36*, 93–105. [CrossRef]
26. Gallo, M.; Montella, B.; Acierno, L.D. The transit network design problem with elastic demand and internalization of external costs: An application to rail frequency optimization. *Transp. Res. Part C* **2011**, *19*, 1276–1305. [CrossRef]
27. Liu, D.; Javier, D.M.; Lu, G.U.; Peng, Q.Y.; Ning, J.; Pieter, V. A Matheuristic Iterative Approach for Profit-Oriented Line Planning Applied to the Chinese High-Speed Railway Network. *J. Adv. Transp.* **2020**, *18*, 4294195. [CrossRef]
28. Scholl, S. *Customer-Oriented Line Planning*; University of Kaiserslautern: Kaiserslautern, Germany, 2005; pp. 23–56.
29. Schöbel, B.; Scholl, S. Line planning with minimal transfers. In Proceedings of the 5th Workshop on Algorithmic Methods and Models for Optimization of Railways, Dagstuhl, Germany, 1 October 2005.
30. Schöbel, A.; Scholl, S. *Line Planning with Minimal Traveling Time*; 5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS'05); Schloss Dagstuhl-Leibniz-Zentrum für Informatik: Mallorca, Spain, 2006; pp. 1–16.
31. Fu, H.L. *Research on Theory and Methods of Line Planning for High-Speed Railways*; Beijing Jiaotong University: Beijing, China, 2010. (In Chinese)
32. Zhao, S.; Wu, R.; Shi, F. A line planning approach for high-speed railway network with time-varying demand. *Comput. Ind. Eng.* **2021**, *160*, 107547. [CrossRef]
33. Wang, L.; Jia, L.; Qin, Y.; Li, H. A two-layer optimization model for high-speed railway line planning. *J. Zhejiang Univ. Sci. A* **2011**, *12*, 902–912. [CrossRef]
34. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3848–3858. [CrossRef]
35. Huang, W.; Song, G.; Hong, H.; Xie, K. Deep architecture for traffic flow prediction: Deep belief networks with multitask learning. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2191–2201. [CrossRef]
36. Shi, M.; Tang, Y.F.; Zhu, X.Q.; Wilson, D.A.; Liu, J.X. Multi-Class Imbalanced Graph Convolutional Network Learning. In Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI, Virtual, 19–26 August 2020; pp. 2879–2885.
37. Ghorbani, M.; Kazi, A.; Baghshah, M.S.; Rabiee, H.R.; Navab, N. RA-GCN: Graph Convolutional Network for Disease Prediction Problems with Imbalanced Data. *arXiv* **2021**, arXiv:2103.00221. [CrossRef]
38. Patel, H.; Singh, R.D.; Thippa, R.G.; Xie, K. A review on classification of imbalanced data for wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2020**, *16*, 1550147720916404. [CrossRef]
39. Yang, Y.Z.; Xu, Z. Rethinking the value of labels for improving class-imbalanced learning. In Proceedings of the Advances in Neural Information Processing Systems 33, Annual Conference on Neural Information Processing Systems, Virtual, 6–12 December 2020.
40. Chen, D.L.; Lin, Y.K.; Zhao, G.X.; Ren, X.C.; Li, P.; Zhou, J.; Sun, X. Topology-Imbalance Learning for Semi-Supervised Node Classification. In Proceedings of the NeurIPS, Virtual, 6–14 December 2021; pp. 29885–29897.
41. Huang, C.; Li, Y.N.; Loy, C.C.; Tang, X. Learning Deep Representation for Imbalanced Classification. In Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Las Vegas, NV, USA, 27–30 June 2016; pp. 5375–5384.
42. Ren, M.Y.; Zeng, W.Y.; Yang, B.; Urtasun, B. Learning to Reweight Examples for Robust Deep Learning. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 4334–4343.
43. Cui, Y.; Li, M.; Jia, T.; Lin, Y.; Song, Y.; Belongie, S.J. Class-Balanced Loss Based on Effective Number of Samples. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Long Beach, CA, USA, 16–20 June 2019; pp. 9268–9277.
44. Cao, K.D.; Wei, C.; Gaidon, A.; Aréchiga, N.; Ma, T.Y. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. In Proceedings of the Advances in Neural Information Processing Systems 32, Annual Conference on Neural Information Processing Systems 2019, NeurIPS, Vancouver, BC, Canada, 8–14 December 2019; pp. 1565–1576.
45. Liu, Z.M.; Nguyen, T.K.; Fang, Y. Tail-GNN: Tail-Node Graph Neural Networks. In Proceedings of the KDD, Virtual, 14–18 August 2021; pp. 1109–1119.
46. Leroy, V.; Cambazoglu, B.B.; Bonchi, F. Cold start link prediction. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–28 July 2010; pp. 393–402.
47. Wang, Z.; Liang, J.; Li, R.; Qian, Y. An approach to cold-start link prediction: Establishing connections between non-topological and topological information. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 2857–2870. [CrossRef]
48. Wang, H.; Zhang, F.; Hou, M.; Qi, Y. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Angeles, CA, USA, 2 June 2018; pp. 592–600.

49. Ge, L.; Zhang, A. Pseudo cold start link prediction with multiple sources in social networks. In Proceedings of the 2012 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics 2012, Anaheim, CA, USA, 26–28 April 2012; pp. 768–779.
50. Tang, M.; Wang, W. Cold-start link prediction integrating community information via multi-nonnegative matrix factorization. *Chaos Solitons Fractals* **2022**, *162*, 112421. [CrossRef]
51. Bryan, P.; Rami, A.; Steven, S. DeepWalk: Online learning of social representations. In Proceedings of the KDD, New York, NY, USA, 24–27 August 2014; pp. 701–710.
52. Grover, A.; Leskovec, J. node2vec: Scalable Feature Learning for Networks. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.