

Prediction of Emergency Mobility Under Diverse IoT Availability

Bin Sun¹, Renkang Geng¹, Yuan Xu^{1,*}, and Tao Shen^{1,2}

¹School of Electrical Engineering, University of Jinan, Jinan 250022, Shandong, China.

²HIT Robot Group Co., Ltd, Xingkai Road, Pingfang, Harbin 150001, China.

Abstract

INTRODUCTION: Prediction of emergency mobility needs to consider more scenarios as Internet of Things (IoT) develops at a high speed, which influences the quality and quantity of data, manageable resources and algorithms.

OBJECTIVES: This work investigates differences in dynamic emergency mobility prediction when facing dynamic temporal IoT data with different quality and quantity considering diverse computing resources and algorithm availability.

METHODS: A node construction scheme under a small range of traffic networks is adopted in this work, which can effectively convert the road to graph network structure data which has been proved to be feasible and used for the small-scale traffic network data here. Besides, two different datasets are formed using public large scale traffic network data. Representative widely used and proven algorithms from typical types of methods are selected respectively with different datasets to conduct experiments.

RESULTS: The experimental results show that the graphed data and neural network algorithm can deal with the dynamic time series data with complex nodes and edges in a better way, while the non-neural network algorithm can predict the with a simple graph network structure.

CONCLUSION: Our proposed graph construction with graph neural network improves dynamic emergency mobility prediction. The prediction should consider the scenarios of availability of computing resources, quantity and quality of data among other IoT features to improve the results. Later, automation and data enrichment should be improved.

Keywords: Emergency Traffic, IoT Time Series, Data Quantity, Data Quality, Machine Learning Prediction.

Received on 04 April 2022, accepted on 07 September 2022, published on 13 September 2022

Copyright © 2022 Bin Sun *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/), which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetpht.v8i4.274

1. Introduction

Emergency medical service is a fundamental requirement in modern society. Dynamic relocation of emergency response resources is required during emergency transit. Conventionally, all transport data is sent to a computer centre to calculate traffic status and plan routes as the edge terminals lack related capabilities and qualified data. The situation is changing. With the rapid development of modern society, technology has penetrated everyone's daily life, ranging from national satellites and aviation to smartphones and smart homes in every household. The progress of science and technology has enriched people's material life[1]. Internet of

Things (IoT) attracted much attention in recent years[2]. The IoT system mainly includes a centre and sub-device terminal[3]. The sub-device terminal connects to the centre and other devices through Bluetooth or wireless LAN for data sharing[4]. The centre is responsible for collecting and processing the running data of each sub-device and adjusting the working state of the device. The sub-device terminal is responsible for receiving the centre signal, performing the corresponding work according to the purpose, and outputting the corresponding device information to the central processor[5]. The Internet of everything has brought great convenience to people in the new era, but at the same time, people have more and more requirements for the devices of the IoT. For a variety of complex work situations, how to

* Corresponding author: Email: cse_xuy@ujn.edu.cn

coordinate the devices to serve people more effectively has always been the focus of relevant personnel[6]. The IoT generates a large amount of time series data. Artificial intelligence can be used to mine the properties and rules of the data itself and accurately detect and predict the data, which can greatly improve people's living efficiency and reduce people's accident risk[7].

Taking dynamic road traffic as an example, traffic data is a kind of dynamic time series data with large volatility, spatial heterogeneity, fast update and huge volume[8]. With the establishment of intelligent transportation systems in first-tier cities and the arrival of the era of the "Internet of vehicles" in the background, relevant traffic data is getting bigger and bigger[9]. Getting more cars on the road in the same amount of time has always been the ultimate goal in dealing with traffic congestion[10]. At the same time, most cities are using the traditional traffic signal system, manually setting the time of passage[11]. Therefore, if the traffic data can be processed and predicted accurately and quickly, urban traffic congestion can be greatly improved. Relevant traffic departments can effectively dredge traffic by adjusting signal lights, adjusting transition lanes and carrying out traffic control, so as to facilitate people's travel route planning and improve people's living efficiency[12].

Early studies in related fields mainly used mathematical modelling methods to fit the data, such as linear model[13], decision tree[14], Auto Regressive Integrated Moving Average (ARIMA) model[15], etc. After that, neural network-based methods[16] especially deep learning algorithms gradually stepped onto the historical stage. By stimulating neurons, it can better refine and aggregate data information and is widely used in the recognition of image, speech and medical information among other domains[17]. However, as the complexity of the neural network algorithm model is high, with a large amount of data and large memory consumption, the computing capacity of computers at that time was insufficient, and the development of deep learning algorithms was stuck in a bottleneck for a period of time[18]. With the breakthrough of computer hardware technology in the past decade, the computing power, memory and other indicators of computers have been greatly improved. Machine learning algorithms based on graph neural networks have shown good results in the field of data prediction[19].

This work aims to investigate differences in dynamic emergency mobility prediction effect when facing different dynamic temporal IoT data with diverse computing resources and algorithm availability then provide suitable data and algorithm advices for dynamic emergency mobility prediction considering recent IoT achievements.

2. Background and Related Work

In this work, the current commonly used and proven temporal data prediction algorithms are considered. Conventional and neural network machine learning algorithms are compared. Non-neural network machine learning algorithms are represented by XGBoost[20] and LightGBM[21], which have

the following characteristics: The algorithm model is small in size, easy to construct, and convenient to implement, with low requirements for computing power and high speed. Neural network machine learning algorithms, represented by the convolutional neural network[22], recurrent neural network [23] and graph neural network algorithm[19], have the following characteristics: good prediction effect, high accuracy, complex model architecture, large requirements for computing power, suitable for large and complex data. Next, the two algorithms are introduced in detail respectively.

2.1. Ensemble Learning

The algorithm composed of a single learner has limited application scope. In the face of complex and highly fluctuating data, it is often unable to effectively extract data features for prediction. Therefore, the concept of the integrated learning algorithm is to put forward, and the flow chart of the integrated learning algorithm is shown in Figure 1.

One ensemble learning algorithm constructs and combines multiple weak learners to form strong learners to complete the learning task and often achieves more remarkable generalization performance than single learners. According to different combination methods, they can be classified into Bagging, Boosting and Stacking[24].

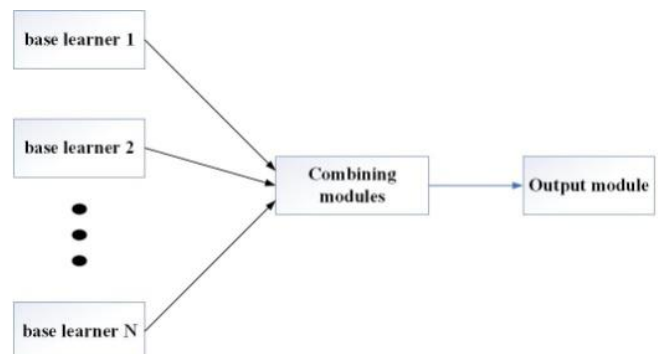


Figure 1. The schematic diagram of integrated learning is shown in the figure, which consists of several weak base learners and forms a better learner for classification or regression prediction.

XGBoost Algorithm

XGBoost is a kind of non-neural network machine learning algorithm proposed in recent years, and has been proved to have a good prediction effect on tabular data. It is widely used in medicine, finance, home, robot control and other fields, and is often used by researchers to predict all kinds of time series data, which is representative to a certain degree. Next, the architecture of XGBoost algorithm is described. XGBoost

algorithm is one of the ensemble learning algorithms built with Boosting. The base learner in this algorithm is essentially CART (Classification and Regression Tree)[25].

CART's main principle is to recursively divide each region in the input space, where the training dataset is stored, into two sub-regions and determine the output value of each sub-region. Depending on the type of tree, the criteria for splitting molecular regions differ. For CART, the squared error minimization standard is commonly used, and the loss function is provided in the formula below.

$$\sum_{x_i \in R_m} (y_i - f(x_i))^2 \quad (1)$$

The target function of XGBoost is as follows:

$$L(\phi) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \quad (2)$$

The objective function of the XGBoost algorithm is composed of the loss function of CART and the regularization term. The loss function represents the existing deviation of the model, and the regularization term represents the existing variance of the model. By adding the regularization term, the number of samples in the fitting function can be reduced as much as possible to prevent the occurrence of over-fitting. The algorithm can be used for both regression and classification prediction.

2.2. Graph Neural Network

Many researchers consider graph neural network algorithms as five types including Graph Convolution Networks (GCN) [26], Graph Attention Networks [27], Graph auto-encoder [28], Graph Generative Networks[29], Graph spatial-temporal Networks[30].

Graph neural network model mainly studies Graph node representation (Graph Embedding), Graph edge structure prediction and Graph classification, the latter two tasks are also based on Graph Embedding expansion. The types of graphs processed are mainly as follows: heterogeneous graphs, bipartite graphs, multidimensional graphs, symbolic graphs, hypergraphs and dynamic graphs. The type of graph constructed in this work is the dynamic graph.

Dynamic graphs add a whole new dimension: time. The dynamic graph neural network algorithm considers both spatial coherence and temporal continuity of nodes in the graph. When a dynamic graph neural network algorithm is dealing with problems, it is in essence to arrange a dynamic graph into a series of slice graphs according to the same time interval, and realize the prediction of nodes in the graph by extracting the features of each slice graph and aggregating them. For dynamic graph neural network algorithms, many current methods use GCN (Graph Convolutional Network)[31] and GAT (Graph Attention Network)[32] to capture the node dependence in slice graph, and RNN(Recurrent Neural Network)[33], LSTM (Long short-

term memory)[34] and GRU (Gated Recurrent Unit)[35] to capture the time dependence.

Graph Convolutional Network

GCN mainly obtains the embedding of node characteristics through the method of matrix operation, The core formula of the algorithm is shown below:

$$H^{(l+1)} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \quad (3)$$

Matrix A refers to the adjacency matrix of the graph, and matrix A^* is represented by the adjacency matrix plus the identity matrix, that is, the self-connection of nodes is added to the original graph. Matrix D is the row sum of matrix A^* , which represents the degree of the node. Matrix $H^{(l)}$ is the original feature of each node, and W is the learning parameter, which is randomly generated and optimized and adjusted according to the results. GCN algorithm takes the node itself into account through self-connection and then takes the information of the connected node into account to normalize the node. Matrix $H^{(l+1)}$ is the feature representation of nodes after embedding by the GCN method. The dimension of the vector depends on the dimension of the w function. GCN method can effectively extract, aggregate, and reduce the features of the original node, that is, obtain the spatial characteristics in the graph.

Gated Recurrent Unit (GRU)

GRU can be considered a simplified version of LSTM and can also handle long-term dependency using gates. It can be understood that a time axis is added between the originally parallel and independent sub-modules to connect them and interact with each other. The logic diagram and detailed flow chart of the algorithm are shown in Figure 2.

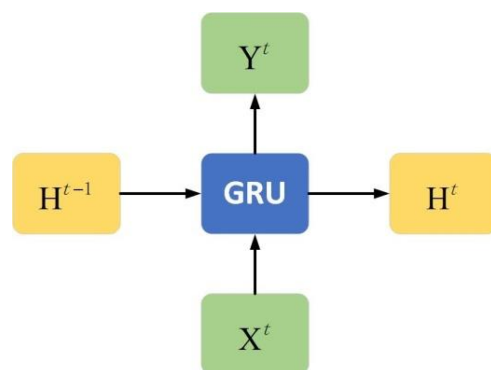


Figure 2. The figure shows the input and output of the GRU module. The input is the information of the current node and the feature input of the last time.

There are two inputs and two outputs in the model. x_t is the input of the node at the current moment, h_{t-1} is the input of node features at the last moment, Y^t is the output of the node at the current moment, h_t represents the aggregation of

node features at the current moment and all previous moments, and the same operation will be continued to the next node. z_t and r_t in the figure represent update and reset gates respectively. The reset gate determines how new inputs are combined with previous memories, and the update gate determines how much previous memories come into play. The update gate controls how much previous state information is taken into the current state while the reset one controls how much previous information is taken into the current candidate set. h_t . A smaller reset gate leads to less information taken from the previous state.

Temporal Graph Convolutional Network (TGCN)

TGCN is a recently proposed spatiotemporal data prediction algorithm based on graph network. The prediction accuracy is high and can be applied to a wide range. It is widely used in transportation, medicine, education and other fields, and has been proved to have a good prediction effect on all kinds of data containing both time and space dimensions. In the neural network algorithm has a certain representative. Next, the concrete architecture of the TGCN algorithm is described. TGCN algorithm[36] uses this method, and its flow chart is shown in the Figure 3 below:

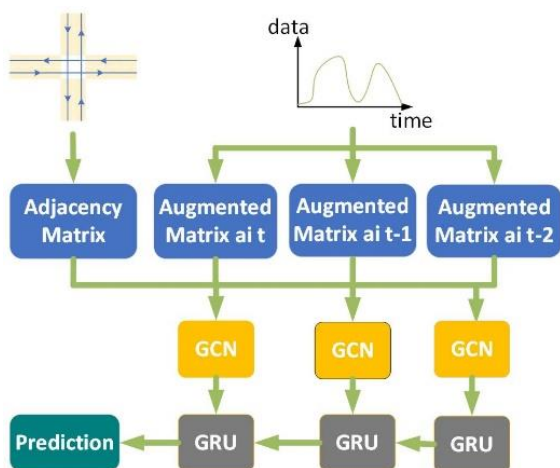


Figure 3. The figure shows the core principle of the TGCN algorithm, which processes road network and traffic data into matrices, extracts spatial features through the GCN algorithm, then extracts time features through the GRU algorithm, and finally outputs prediction results.

TGCN algorithm, the road network data is processed into an adjacency matrix, and the sequential traffic data is processed into a series of feature matrices according to fixed time intervals. The adjacency matrix and feature matrix are combined to obtain data spatial attributes in the GCN model, and then the h_t obtained is added to the GRU model to obtain data time attributes. Finally, the corresponding prediction results are output.

Therefore, in this experiment, XGBoost algorithm is used as the baseline of non-neural network machine learning

algorithm, and TGCN algorithm is used as the baseline of neural network machine learning algorithm. But both algorithms have some limitations. XGBoost algorithm has a simple model architecture, which tends to have low prediction accuracy in the face of large amount of data, complex data content, and high physical correlation data, while TGCN algorithm has some problems, such as complex model architecture, low interpretability, slow algorithm iteration speed, and high computing power requirements.

3. Methodology

The overall flow chart of this work is shown in Figure 4. First, the original data is obtained from multiple sensor devices, followed by preprocessing to ensure accuracy and simplicity. Later, according to the physical location and correlation of devices, nodes and edges are designed to form a graph network. All related data between different devices are connected to relevant nodes based on the graph network. By setting the time interval, data is generally processed into node features, which are output along with the adjacency matrix. Finally, the results are predicted using different algorithms to guide the further work of the equipment.

Next, the dynamic road traffic conditions are taken as an example to explain the experimental method in detail.



Figure 4. The figure shows the flow chart of this work.

The raw data is read first, then the data is preprocessed, the processed data is given to the algorithm for training, and the prediction results are evaluated at last.

The content of this work mainly includes four aspects. Firstly, data collection and acquisition are carried out. Public urban road traffic data can be obtained through network channels, and unpublicized urban traffic data can also be obtained through cooperation with relevant transportation departments. After that, the obtained data are preprocessed to remove the problematic data, and the road nodes and corresponding node features are established. Then the algorithm is built and optimized, and the data is given to the algorithm for prediction. After obtaining the predicted results, evaluate and analyze them. The experiments are done using Python programming language.

3.1. Data Acquisition

Many ways can be used to acquire traffic data, such as visual tracking[37] or loop detection. The experiments using traffic data mainly include two kinds, one is from the related work[36], derived from the network public thousands of road of Shenzhen city road network data, containing the adjacency matrix of all urban roads within one month and the average speed of the vehicle. The average speed is taken as the main node feature. In this work, it is called large-scale urban road network data. As most of the urban traffic data sources are common intersection monitoring probes, did not establish intelligent traffic networks, so in this work, we chose second - and third-tier Chinese cities with the same time number of coherent intersection traffic monitoring data. The data include license plate number, vehicle type, through time, passage direction, etc. In this work, it is called small-scale urban road network data. The original example data are shown in Table 1 below.

Table 1. Part of the original data is excerpted in the table for display, including Time, Vehicle ID, Crossroad, Direction, Lane and other features.

Time	Vehicle ID	Crossroad	Direction	Lane
2021-08-12 12:47:24	7619	1	North to south	1
2021-08-12 12:47:30	6673	1	East to West	4
2021-08-12 12:47:35	5931	1	West to East	3
2021-08-12 12:47:39	0810	1	West to East	4
2021-08-12 12:47:46	0197	1	South to North	2

3.2. Graphed Node Construction

Preprocessing is needed as the collected data can be contaminated by equipment failure or other factors. For example, traffic data often have a large number of errors caused by non-motor vehicle driving. These issues affect the analysis of motor vehicle data and reduce the prediction accuracy of future data. Thus, we first preprocess the original data to ensure the cleanliness.

First a number of coherent intersections are selected and each intersection contains four lanes. Lane one is for left turning, lane 2 and 3 for going straight, lane 4 for right turning respectively. This can accurately distinguish paths for specific vehicles and match the driveway data to the corresponding road nodes. According to the above connections, we set up road nodes and adjacency matrix, as shown in Figure 5.

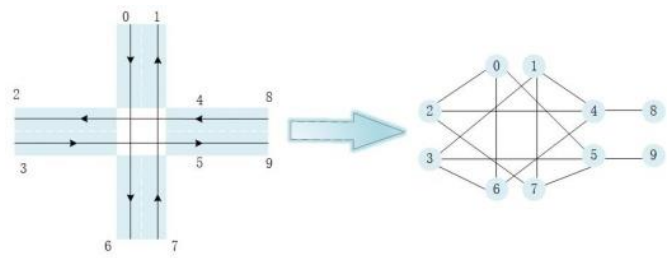


Figure 5. The figure shows the method of processing the traditional intersection nodes into a graph network structure, connecting corresponding road nodes according to different travel directions.

For example, node 0 is connected to node 2 through the right-turning lane, so the data of the right-turning lane in node 0 is matched to node 2; meanwhile, node 4 is also connected to node 2 through a straight-going lane, so the data of the right-turning lane in node 4 is matched to node 2. After the above operations, we will process the processed traffic data into traffic flow data at a fixed time interval (the time interval available in this work is five minutes), and take traffic flow data as the main feature of node prediction. The processed example data are shown in Table 2 below.

Table 2. The table shows a small piece of data after processing, which contains the traffic flow of each node within 5 minutes.

Time	Node 0	Node 1	Node 2	Node 3
2021-08-12 12:35:00-12:40:00	35	42	26	32
2021-08-12 12:40:00-12:45:00	42	45	52	55
2021-08-12 12:45:00-12:50:00	34	20	23	25
2021-08-12 12:50:00-12:55:00	23	12	27	19
2021-08-12 12:55:00-13:00:00	15	21	19	9

3.3. Lagged Prediction

For ensemble algorithms, in order to enrich the time features of our experiment, we processed lag on the traffic dataset and treated traffic data with a five-minute time delay as a lag feature, that is, the traffic flow at the intersection in the past time as a feature of model learning and as the time feature of predicting the traffic flow data at the intersection at the current time. Following that, the XGBoost method model was built, the data was put into the model for prediction, and the results were obtained. The algorithm flow chart is shown in Figure 6. Firstly, the data is read, then the relevant features are constructed, and then the data is fed to the XGBoost algorithm. Finally, the best prediction results are obtained by adjusting the algorithm parameters for evaluation.

For the TGCN algorithm, the adjacency matrix of road nodes processed previously and corresponding timing flow data of road nodes are added to the algorithm. Appropriate training set proportion, num_{walk} and other parameters are selected for modelling prediction, and the prediction data of the final output nodes are evaluated accordingly.

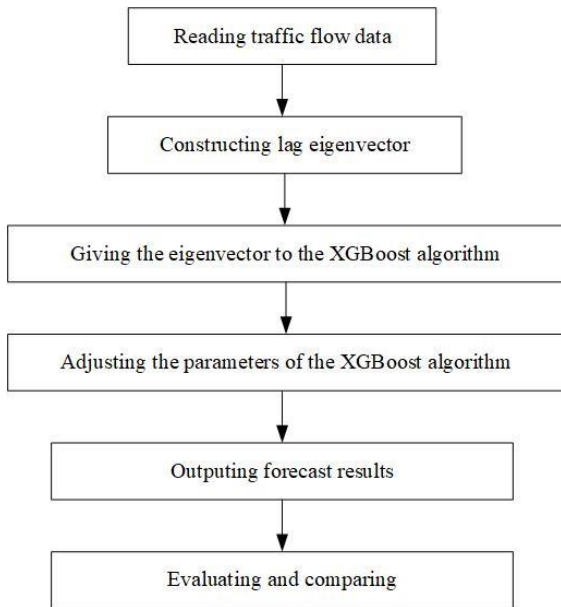


Figure 6. The flow chart of prediction using the XGBoost algorithm in this work.

3.4. Evaluation Metrics

The prediction of traffic data is a typical machine learning regression prediction, so we use the three most commonly used indicators in regression prediction to measure the prediction effect. Including (1) Mean Absolute Error (MAE) (2) Root Mean Squared Error (RMSE) (3) R Squared (R²). Assuming that y_i is the real data and \hat{y}_i is the predicted data, the definition formula of indicators is as follows:

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2} \quad (4)$$

$$MAE = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i| \quad (5)$$

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\hat{y}_i - \bar{y})^2} \quad (6)$$

4. Results

In this work, the traffic data of a small area road network were added to the two algorithms, and the experimental results were obtained as shown in Table 3.

Table 3. Comparison of prediction algorithms in the case of small-scale road network data.

Metrics	XGBoost	TGCN
MAE	5.2557	2.0711
RMSE	8.2477	3.4833
R-squared	0.8241	0.7175

Through the above table, it can be found that, for the prediction of small-range road network flow data, the MAE and RMSE of the XGBoost algorithm are 5.2557 and 8.2477, generally higher than the MAE and RMSE of the TGCN algorithm. However, the R² of the XGBoost algorithm is 0.8240. In the TGCN algorithm, R² is 0.7175. In this respect, the XGBoost algorithm has a better prediction effect than the TGCN algorithm. Later, in order to explore the reasons for this situation, we conducted a further analysis of traffic data.

Table 4. Comparison of prediction algorithms in the case of large-scale road network data.

Metrics	XGBoost	TGCN
MAE	3.2777	2.746
RMSE	8.6005	4.0696
R-squared	0.7845	0.8388

We guess that the traffic data is greatly affected by time, and the travel demand of people at night and during working hours is much less than that of rush hour, so the value range of a large part of the traffic data is between 0 and 10. TGCN algorithm has high accuracy in predicting this situation, so MAE and RMSE values are small. The XGBoost algorithm responds better when the data starts climbing or falling rapidly (i.e., when the fluctuation becomes more severe), so R² is larger. At the same time, it also indicates that the traffic flow data of a small road network pay more attention to the traffic characteristics of its nodes in the past period of time.

The experimental results of large-scale road network speed data are shown in Table 4. It is found that the RMSE value of the TGCN algorithm is 4.0696 and the MAE value is 2.7460, while the RMSE value of the XGBoost algorithm is 8.6005 and the MAE value is 3.2777. TGCN has better performance. Meanwhile, the R² value of TGCN reaches 0.8388, which is closer to 1 than the R² value of the XGBoost algorithm. Experimental results show that the TGCN algorithm has a better prediction effect.

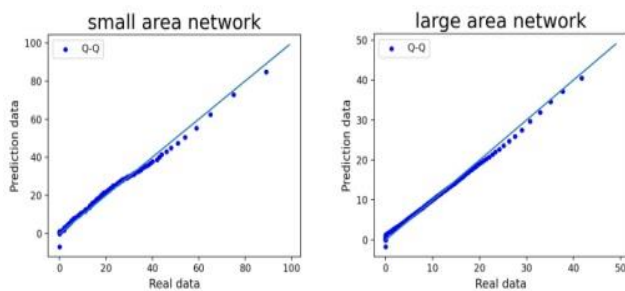


Figure 7. Non-neural network prediction.

The figure shows the actual-vs-predicted plot of the prediction results of the XGBoost algorithm under two kinds of data. On the whole, the prediction data of the algorithm is relatively close to the real data. When the value of the real data is low, the prediction result is slightly higher, while when the real data is gradually rising, the corresponding prediction result is slightly lower.

5. Analysis

Through the analysis of the experimental results, in a wide range of road network data, there are more connections between road nodes, and the correlation and heterogeneity between road nodes are stronger. The importance of main roads is different from that of ordinary roads, so the weight of edges between nodes is also different. TGCN algorithm can better weight allocation and aggregation of information between nodes and neighbouring nodes, while the XGBoost algorithm cannot define the connection of nodes and cannot aggregate node information well. Therefore, the TGCN algorithm can show better performance in this kind of dataset. When the road network covered by traffic data is larger, the graph neural network algorithm is more suitable for prediction. When the road network data covered by traffic data has fewer nodes, the integrated learning algorithm based on the XGBoost can focus more on the characteristics of nodes themselves, and the model is easy to build and the prediction speed is fast, so it can show better results. This method improves universality and scalability, especially for processing and predicting time series data of Internet of Things. The data scope of this method includes the data provided by a single device and the multi-dimensional data between multiple devices, especially the data sharing and analysis among multiple devices with physical correlation. It can be used in transportation, medicine, home and other fields.

6. Conclusion

There are two main contributions in this work. First, a graphed node construction is done for intersection traffic data. This adopts a node construction scheme under a small

traffic network with small data, which can effectively treat the road as a graph network structure. Second, two representative algorithms and distinctive IoT datasets are used to evaluate the prediction effect and explore the influence of possible time-varying environments applicable to the non-neural network algorithm and neural network algorithm for dynamic emergency mobility prediction.

The results prove that graphed node construction is feasible and show that the traffic data from a wide range of urban road networks with sufficient nodes and edges gives a stronger and clearer correlation between nodes within neural networks which better distinguishes weights of edges among nodes, thus a better information fusion of adjacent nodes. This leads to more accurate neural network predictions than non-neural ones but with more computing resources requirements. However, for a small range of road nodes, the connections among nodes are not very sufficient. As the non-neural modelling method pays more attention to the characteristics of nodes themselves, it shows better predictions than neural ones. The results give a guide that the quality and quantity of IoT data characteristics should be considered together with available computing resources on edges like ambulances or rode-side equipment to decide on processing methods.

7. Discussion

There are some limitations should be considered. First, the targeted data should be enriched as other related information may increase the prediction. Second, data cleaning and data characterizing can be more automatic. Third, parameters and network struct can be optimized in general.

Acknowledgement

The authors would like to thank reviewers and editors for their constructive comments. This work is supported by National Key R&D Program of China 2018AAA0101703, Shandong Key R&D Program 2019JZZY021005 and Natural Science Foundation of Shandong ZR2020MF067.

References

- [1] Joan Solomon. *Teaching Science, Technology and Society. Developing Science and Technology Series*. ERIC, 1993.
- [2] Shuai Liu, Shuai Wang, Xinyu Liu, Jianhua Dai, Khan Muhammad, Amir H Gandomi, Weiping Ding, Mohammad Hijji, and Victor Hugo C de Albuquerque. Human inertial thinking strategy: A novel fuzzy reasoning mechanism for iot-assisted visual monitoring. *IEEE Internet of Things Journal*, 2022.
- [3] Luigi Atzori, Antonio Iera, and Giacomo Morabito. Internet of Things: A survey. *Comput. Netw.*, 54(15):2787–2805, 2010.
- [4] Xiaoyi Cui. Internet of Things. In *Ethical Ripples of Creativity and Innovation*, pages 61–68. Springer, 2016.
- [5] Somayya Madakam, Vihar Lake, Vihar Lake, Vihar Lake, and others. Internet of Things (IoT): A literature review. *J. Comput. Commun.*, 3(05):164, 2015.

- [6] Mahdi H Miraz, Maaruf Ali, Peter S Excell, and Rich Picking. A review on Internet of Things (IoT), Internet of everything (IoE) and Internet of nano things (IoNT). In *2015 Internet Technologies and Applications (ITA)*, pages 219–224. IEEE, 2015.
- [7] Ganapathy Mahalakshmi, S Sridevi, and Shyamsundar Rajaram. A survey on forecasting of time series data. In *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, pages 1–8. IEEE, 2016.
- [8] Guillaume Leduc and others. Road traffic data: Collection methods and applications. *Work. Pap. Energy Transp. Clim. Change*, 1(55):1–55, 2008.
- [9] Beying Deng and Xufeng Zhang. Car networking application in vehicle safety. In *2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA)*, pages 834–837. IEEE, 2014.
- [10] Zehang Sun, George Bebis, and Ronald Miller. On-road vehicle detection: A review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(5):694–711, 2006.
- [11] Yizhe Wang, Xiaoguang Yang, Hailun Liang, and Yangdong Liu. A review of the self-adaptive traffic signal control system based on future traffic environment. *J. Adv. Transp.*, 2018, 2018.
- [12] Luo Qi. Research on intelligent transportation system technologies and applications. In *2008 Workshop on Power Electronics and Intelligent Transportation System*, pages 529–531. IEEE, 2008.
- [13] Dennis V Lindley and Adrian FM Smith. Bayes estimates for the linear model. *J. R. Stat. Soc. Ser. B Methodol.*, 34(1):1–18, 1972.
- [14] Anthony J Myles, Robert N Feudale, Yang Liu, Nathaniel A Woody, and Steven D Brown. An introduction to decision tree modeling. *J. Chemom. J. Chemom. Soc.*, 18(6):275–285, 2004.
- [15] Siu Lau Ho and Min Xie. The use of ARIMA models for reliability forecasting and analysis. *Comput. Ind. Eng.*, 35(1-2):213–216, 1998.
- [16] Shuai Liu, Shuai Wang, Xinyu Liu, Amir H Gandomi, Mahmoud Daneshmand, Khan Muhammad, and Victor Hugo C De Albuquerque. Human memory update strategy: a multi-layer template update mechanism for remote visual monitoring. *IEEE Transactions on Multimedia*, 23:2188–2198, 2021.
- [17] Bo K Wong, Thomas A Bodnovich, and Yakup Selvi. Neural network applications in business: A review and analysis of the literature (1988–1995). *Decis. Support Syst.*, 19(4):301–320, 1997.
- [18] Yu-chen Wu and Jun-wen Feng. Development and application of artificial neural network. *Wirel. Pers. Commun.*, 102(2):1645–1656, 2018.
- [19] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Trans. Neural Netw.*, 20(1):61–80, 2008.
- [20] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.
- [21] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.*, 30, 2017.
- [22] Saad Albawi, Tareq Abed Mohammed, and Saad AlZawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. Ieee, 2017.
- [23] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari, 2010.
- [24] Xiaohu Shi, Yanwen Li, Haijun Li, Renchu Guan, Liupu Wang, and Yanchun Liang. An integrated algorithm based on artificial bee colony and particle swarm optimization. In *2010 Sixth International Conference on Natural Computation*, volume 5, pages 2586–2590. IEEE, 2010.
- [25] Roger J Lewis. An introduction to classification and regression tree (CART) analysis. In *Annual Meeting of the Society for Academic Emergency Medicine in San Francisco, California*, volume 14. Citeseer, 2000.
- [26] Yifei Shen, Yongji Wu, Yao Zhang, Caihua Shan, Jun Zhang, B Khaled Letaief, and Dongsheng Li. How Powerful is Graph Convolution for Recommendation? In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1619–1629, 2021.
- [27] Petar Velicković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *ArXiv Prepr. ArXiv171010903*, 2017.
- [28] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *ArXiv Prepr. ArXiv161107308*, 2016.
- [29] Wengong Jin, Kevin Yang, Regina Barzilay, and Tommi Jaakkola. Learning multimodal graph-to-graph translation for molecular optimization. *ArXiv Prepr. ArXiv181201070*, 2018.
- [30] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 922–929, 2019.
- [31] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters—improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4353–4361, 2017.
- [32] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *stat*, 1050:20, 2017.
- [33] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *ArXiv Prepr. ArXiv14092329*, 2014.
- [34] Ralf C Staudemeyer and Eric Rothstein Morris. Understanding LSTM—a tutorial into long shortterm memory recurrent neural networks. *ArXiv Prepr. ArXiv190909586*, 2019.
- [35] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *ArXiv Prepr. ArXiv14061078*, 2014.
- [36] Jiawei Zhu, Qiongjie Wang, Chao Tao, Hanhan Deng, Ling Zhao, and Haifeng Li. AST-GCN: Attributeaugmented

spatiotemporal graph convolutional network for traffic forecasting. *IEEE Access*, 9:35973–35983, 2021.

- [37] Shuai Liu, Shuai Wang, Xinyu Liu, Chin-Teng Lin, and Zhihan Lv. Fuzzy detection aided real-time and robust visual tracking under complex environments. *IEEE Transactions on Fuzzy Systems*, 29(1):90–102, 2020.