Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

Traffic prediction for diverse edge IoT data using graph network

Tao Shen[1], Lu Zhang[1], Renkang Geng[2], Shuai Li[1] and Bin Sun[1*]

Abstract

More researchers are proposing artificial intelligence algorithms for Internet of Things (IoT) devices and applying them to themes such as smart cities and smart transportation. In recent years, relevant research has mainly focused on data processing and algorithm modeling, and most have shown good prediction results. However, many algorithmic models often adjust parameters for the corresponding datasets, so the robustness of the models is weak. When different types of data face other model parameters, the prediction performance often varies a lot. Thus, this work starts from the perspective of data processing and algorithm models. Taking traffic data as an example, we first propose a new data processing method that processes traffic data with different attributes and characteristics into a dataset that is more common for most models. Then we will compare different types of datasets from the perspective of multiple model parameters, and further analyze the precautions and changing trends of different traffic data in machine learning. Finally, different types of data and ranges of model parameters are explored, together with possible reasons for fluctuations in forecast results when data parameters change.

Keywords Traffic time series prediction, Graph neural network, Edge data, IoT networking

Introduction

With the rapid progress of current science and technology and the arrival of the Internet of Things (IoT) era, the number of various intelligent devices used in people's daily lives has increased significantly. The huge number of smart devices generates a large amount of data, providing a large number of samples for the field of artificial intelligence (AI). Researchers have used AI algorithms in a variety of traditional fields, including industry and machinery [1]. For example, deep learning methods composed of different encoders and decoders can realize industrial system design [2], abnormal detection, mechanical fault diagnosis, and other functions [3]. Reasonable analysis and processing of data can monitor the operation status of equipment, predict the trend of equipment, improve the efficiency of citizens' work and life, and ensure the safety of people's property [4]. The Internet of Things makes a network between different devices by setting device nodes and edges and integrating and aggregating data features through node computing and edge computing [5]. Researchers can more accurately analyze the situation between the whole and the individual, clarify the correlation between different devices, and obtain the characteristics of devices by forming network interconnection data between devices for node computing and edge computing. Finally, better detection of the equipment can be achieved and the future operation of the equipment can be predicted. Under the current social background, this research has very practical value in many fields, such as transportation, weather, epidemic transmission, air quality, earthquake direction, etc. [6]. Many researchers combine edge computing with image processing, cloud computing, and other methods to monitor the operation of intelligent devices of the IoT

*Correspondence:
Bin Sun
cse_sunb@ujn.edu.cn
[1] School of EE, University of Jinan, Jinan 250022, China
[2] Shandong High-Speed Info. Group Co. LTD, Lvyou Rd. 11777, Jinan 250100, China

[7, 8]. Next, take the traffic situation as an example to explain in detail.

As the road network of modern urban traffic is highly complex, and the number of cars owned by citizens has increased significantly, the entire road network can no longer be effectively supervised and controlled manually. The work difficulty and workload of purely manual participation in traffic management are increasing [9]. Therefore, the concept of intelligent transportation was proposed by researchers [10]. It is a way to solve traffic problems and alleviate traffic congestion in recent years by integrating machines into traffic and borrowing machines to participate in the analysis, prediction, and future traffic decisions of traffic data [11]. Only when the future traffic situation is mastered, can the management decision-makers make regulations based on the basis, such as traffic police call, intersection patrol, signal time adjustment, and other actions [12]. With the increasingly improved national infrastructure, we can collect relatively complete traffic flow data through road monitoring, providing a large number of samples for the training of artificial intelligence algorithms [13]. On the other hand, in recent years, the concept of the "Internet of Vehicles" has been proposed, and a large number of smart cars, mainly new energy vehicles, have begun to come out and achieved good sales [14]. These vehicles can upload vehicle location information and time information via GPS. The Internet of vehicles can transmit terminal data and carry out edge computing through on-board applications [15]. This kind of floating car data has gradually become an important part of traffic data samples [16]. To make full use of this kind of data to analyze traffic conditions, researchers have proposed mobile edge computing and other related technologies [17].

In traffic management, the traffic flow and average speed of a specific section at a specific time are two very important traffic characteristics that describe the traffic conditions of the section [18]. Effective analysis of these two traffic characteristics can accurately study and judge road traffic conditions [19]. Then take corresponding traffic management actions, such as adjusting the time of traffic lights, setting variable lanes, etc. These measures can directly or indirectly regulate the traffic flow and improve the speed of traffic to ensure the smoothness of specific sections [20]. At present, the scientific research community conducts various dimensional analyses and processing on a large number of traffic data to obtain more ideal traffic conditions, facilitate the prediction of future traffic, greatly improve the operation efficiency of the urban road network, and indirectly improve the production and living efficiency and well-being of residents [21].

Traffic is an outdoor human activity. The transmission of traffic data mainly depends on the sensors of the vehicle itself and the fixed point sensors set on the road. Data transmission is carried out through wired, wireless networks, and Bluetooth devices [22]. This data collection method is prone to data loss at the production end and the transmission end [23]. When there is a certain degree of loss in the data, it will often have a greater impact on data processing and analysis [24]. If there is a big deviation between the prediction result and the actual situation, the decision made based on such prediction may not only not improve the operation efficiency of urban roads, but also cause new problems such as congestion and traffic accidents [25].

At present, the capture of original traffic data in most cities in the world mainly comes from two channels, namely, floating car GPS data and road sensor data [26]. Therefore, in this paper, we try to process the original traffic data through different methods to form a graph network and related traffic feature data and then use the graph neural network machine learning algorithm to predict. The purpose of this paper is to study and summarize the methods of making and collecting several different traffic datasets and set different parameters to predict the data through algorithms. Finally, from the perspective of several different important parameters, we can find the changing trend of algorithm prediction performance under different datasets. In the second part of this paper, we will introduce the current research background in this field and the algorithms related to this work. In the third part of this paper, the principle and process of the work will be introduced in detail. The fourth and fifth parts of this paper are the results of the paper and related analysis and discussion, respectively, describing the relevant results of this work.

The main contributions of this work are:

- proposing a novel graph-structure-based method of processing floating car data whose effects are verified;
- discovering the convergence speed of prediction algorithms to different characteristics of traffic datasets;
- exploring the influence of different lag time feature schemes on the prediction results when forecasting datasets.
- exploring the impact of missing value proportion and hidden factor dimension on different types of data.

Related work and background

Graph Neural Network Neural Algorithm (GNN)

Graph neural network algorithm [27] is an important branch of the machine learning field in recent years, which is characterized by accurate modeling and good prediction effect. The graph neural network algorithm modeling is complex, and the computer hardware is very high, so the development of this field encounters a bottleneck [28]. In recent years, with the rapid development of computer-related hardware capabilities and significant improvement of computer computing power, graph neural network algorithm has attracted more and more researchers' attention [29]. Graph neural network algorithms can effectively aggregate spatial characteristics of data based on graph network structure, such as Graph Convolutional Networks(GCN) [30], Graph Attention Networks(GAT) [31], GraphSAGE [32], etc. The early graph neural network algorithm can only deal with static graphs, while the large amount of data faced in daily life often includes an important dimension - time. Spatio-temporal data [33] is a very common data type in transportation, medicine, community management, and other fields Spatio-temporal data contains important time dimensions. The feature extraction of its data should consider both the time characteristics of nodes and the spatial characteristics of nodes. Therefore, researchers have proposed a Recurrent Neural Network(RNN) algorithm [34] to aggregate the time characteristics of nodes. It can also be combined with image processing, edge computing, and other fields [35, 36].

Recurrent Neural Network Algorithm (RNN)

The Recurrent Neural Network algorithm, which is an extension of traditional feedforward neural networks, can handle variable length sequence input. It learns variable length input sequences through internal cyclic hidden variables, and the output of the activation function of hidden variables at each time depends on the output of the cyclic hidden variable activation function at the previous time.

Given an input sequence $x = (x_1, x_2, ..., x_T)$, the cycle updating process of the RNN model is as follows:

$$h_t = g(Wx_t + Uh_{t-1}) \tag{1}$$

Where g is an activation function (such as a logistic sigmoid function or hyperbolic tangent function). W is the weight matrix of the hidden variable input to this moment. U is the weight matrix of the hidden variable from the previous time to this time. Given the currently hidden state h_t, RNN can be used to represent the joint probability distribution on the input sequence.

In the structure of the traditional fully connected neural network, neurons do not affect each other, there is no direct connection, and neurons are independent of each other. Like the traditional fully connected neural network, RNN is also composed of an input layer, a hidden layer, and an output layer. However, the relevant parameters of the hidden layer are not only related to the input signal but also related to the parameters of the previous neuron's hidden layer.

RNN is similar to adding a time axis to an ordinary fully connected neural network, connecting neurons that are not originally related to each other. RNN is characterized that, for each RNN neuron, its parameters are always shared, that is, for a text sequence, any input is processed the same to get an output.

On the other hand, the traditional fully connected neural network usually uses the horizontal arrangement for visualization, that is, each layer of neurons is arranged vertically. On the other hand, RNN is just the opposite. The visualization of neurons at each layer usually adopts a horizontal arrangement, which is convenient for effectively displaying the time correlation between neurons.

When facing long sequence data, the RNN unit can easily encounter gradient dispersion, which makes the RNN only have short-term memory, that is, the RNN can only obtain the information of the nearest sequence, while it has no memory function for the earlier sequence, thus losing important information. To improve the long-term memory function of the algorithm and avoid gradient explosion due to too long a data sequence, Long short-term memory recurrent neural network(LSTM) [37] and Gated Recurrent Unit(GRU) [38] algorithms are proposed. LSTM and GRU evolved from RNN. Next, the LSTM algorithm is introduced in detail.

LSTM algorithm

To make the model dependent for a long time, the LSTM algorithm [37] adds a gating structure, which makes LSTM add a transmission state compared with RNN to control the flow of features and screen them. The forgetting gate is the forgetting stage of the algorithm. Selectively forget the input of the previous node to ensure that important information is obtained while reducing the length of input time. It determines the unit state at the previous time and how much remains at the current time. The input gate, namely the memory stage of the algorithm, determines how much of the current network input is saved to the cell state. Output gate, that is, the output stage of the algorithm, the state of the control unit, and how many outputs are to the current output value of LSTM.

Graph network construction

Before using a graph signal processing algorithm, it is necessary to construct a graph network in an appropriate way [39]. For graph structures such as urban traffic road networks, abstract methods can be taken from the association of real physical locations, such as abstracting intersections as nodes in the graph structure and abstracting roads as edges between nodes. Or divide each road into small segments, abstract the small segments into graph-structured points, and abstract the connection between them into node edges [40]. The latter is more refined and often used in road network modeling based on road sensors (monitoring, etc.) and navigation modeling. This structure map network can be used not only in the processing of traffic data but also in the monitoring and prediction of a series of intelligent devices such as air temperature and air pollutants [41].

Matrix factorization

In the current era of big data, many data features need to be processed. Which features play an important role in machine learning and which features have a strong correlation have always been the focus of feature engineering. If only manual design is used, these problems are not only slow but also time-consuming (for example, Support vector machine(SVM) [42]). Therefore, the method of matrix factorization has become an important idea for researchers to solve this problem in recent years [28]. Matrix factorization means that the matrix is decomposed into two low-rank matrices. The formula is as follows:

$$Y \approx W\mathbf{X} \tag{2}$$

Where, $Y \in R^{N\times T}$ is the original matrix, $W \in R^{N\times r}$ represents the low rank spatial characteristic matrix, and $X \in R^{r\times T}$ represents the low rank temporal characteristic matrix. An important idea of machine learning matrix factorization is to allow the machine to select its features, and extract and aggregate the features, that is, it does not need to manually mine the features in the feature matrix, and it can be found through the machine's calculation. The matrix factorization method is often used to fill in missing values, but it cannot predict future data [43].

Spatio-temporal data includes two important data dimensions, namely spatial dimension, and time dimension. Therefore, extracting the features of Spatiotemporal data is essential to extract the features of the numbers in the matrix. Because the data itself is affected by various external factors, the data in the matrix is often complex and feature extraction is difficult. Therefore, researchers began to use the matrix decomposition method to extract features of complex and high-dimensional matrices. Next, a decomposition method of sequence matrix is introduced.

Temporal regularized matrix factorization

Different from the traditional matrix decomposition model, the Temporal Regularized Matrix Factorization(TRMF) [44] model learns and makes use of the time dependence existing in the time features when performing matrix decomposition, and each feature vector is closely related to the vectors of some previous time steps. TRMF describes time correlation as time regularization. The time regularizer is trained from the decomposed time characteristic matrix X. In turn, the time correlation learned by TRMF is used to regularize the matrix decomposition process so that better eigenmatrices can be learned.

The formula of TRMF is as follows:

$$\min_{W,x,\theta} \frac{1}{2} \sum_{(i,t)\in\Omega} \left(y_{it} - \mathbf{w}_i^T \mathbf{x}_t\right)^2 + \frac{\lambda_w \eta}{2} ||W||_F^2 + \frac{\lambda_x \eta}{2} ||X||_F^2 + \lambda_x \mathcal{R}_t(X|\theta) \tag{3}$$

In the formula, the first term is the objective function, the second and third terms are the regularization terms corresponding to the time and space characteristic matrices respectively, and the fourth term is the regularization term of the time series coefficient matrix.

LSTM and Graph Laplacian Regularized Matrix Factorization (LSTM-GL-ReMF)

In this paper, we use a sequence data prediction algorithm based on LSTM and Graph Laplacian(GL) [45] models in the work. The algorithm is called LSTM and Graph Laplacian Regularized Matrix Factorization(LSTM-GL-ReMF) [46], It decomposes multi-dimensional Spatio-temporal data into two parts: time matrix and space matrix (matrix decomposition) by using the time series matrix method. The W matrix represents the spatial data of the data and then uses the regularization of the Tulaplacian space to aggregate the data to obtain the spatial expression of the features. The X matrix represents the time data of the data, which is then sent to the LSTM module to capture the time rule of the data. Finally, they are restored to a set of Spatiotemporal data matrices to complete the prediction.

Method

Experimental design

The overall process and contents of this work are shown in the following Fig. 1.

Experimental dataset

To compare the performance differences of algorithms under completely different traffic data, to better fill and forecast the traffic data, we obtained four different types of traffic datasets. Among them, Seattle and Shenzhen datasets are traffic speed datasets. Chengdu dataset is a traffic flow dataset based on GPS data of floating cars. The small-scale road network data is the traffic flow dataset based on the road intersection monitoring system. Next, we will introduce it in detail.

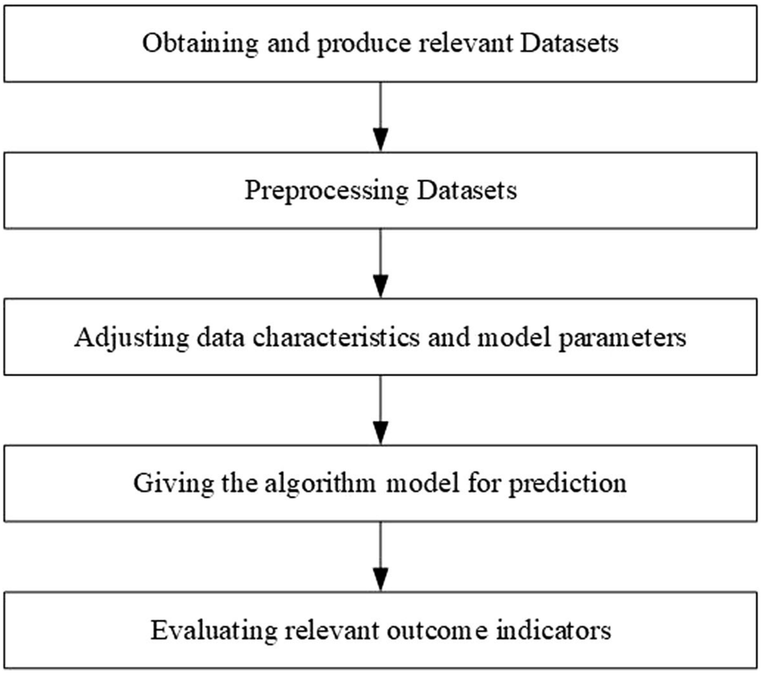
Obtaining and produce relevant Datasets
Preprocessing Datasets
Adjusting data characteristics and model parameters
Giving the algorithm model for prediction
Evaluating relevant outcome indicators

Fig. 1 The figure shows the specific process of this work. First, obtain and make relevant datasets, then preprocess all datasets, and then adjust the data characteristics of the dataset to build an algorithm model. After the above work is completed, the data will be sent to the algorithm model for prediction. Finally, the relevant outcome indicators are evaluated and analyzed

Seattle dataset

Seattle traffic speed dataset [9], published by Google, records the average traffic speed data collected by 323 sensors on the urban ring road in Seattle in 17568 time periods. The duration of each period is 5 minutes, and the average speed of all passing vehicles in this period is recorded. Therefore, the dataset is composed of a 323 * 323-dimensional adjacency matrix and a 323 * 17568 traffic speed characteristic matrix. Hereinafter, we refer to Dataset 1(D1).

Shenzhen dataset

For this work, we collected the traffic speed data of Shenzhen published by the transportation big data department, which is composed of a 156 * 156 adjacency matrix and a 156 * 2976 data matrix[30]. It includes the average speed of 156 roads in a certain area of Shenzhen in 2976 time periods from January 1, 2015, to January 31, 2015. Each period is 15 minutes long. In this data, each road is set as a node, and a total of 156 road nodes are formed. The adjacency matrix is formed by the connection of roads to represent the network structure of the graph. Hereinafter referred to as Dataset 2(D2).

Chengdu dataset

The floating car data used in this work is the public GPS data of taxis in Chengdu. This dataset covers more than 300 million GPS records of 14000 taxis in Chengdu, from 6 o'clock to 24 o'clock every day from August 17 to August 23, 2014. The data format includes vehicle ID, latitude, longitude, passenger status, and time. First, we read the GPS data and downloaded the urban road network data of Chengdu to determine the division of main functional areas in Chengdu, the location of urban trunk roads, and the geographical correlation between different regions. Then match the GPS data to the urban road network map to determine the main distribution position of the GPS data. After determining the distribution location, evaluate the density of the data location, and select areas with a high density of GPS data points, large traffic flow, large flow of people, and easy congestion in daily peak hours. Take the longitude and latitude of this area as the overall longitude and latitude range of this dataset. The thermal diagram of GPS data is shown in Fig. 2.

Next, we divide the overall scope into a series of regions. For the design of area size, each selected area is not easy to be too large or too small. The area is too small and the representativeness is reduced, which is not conducive to the overall analysis. The area is too large and the data is too general. When the data generates problems, such as traffic congestion, it cannot accurately reflect the location of traffic congestion. Regions should be adjacent to each other to ensure that data has both temporal and spatial continuity.

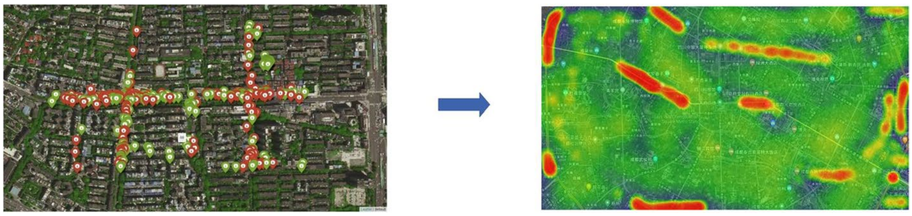

Fig. 2 The figure shows the results obtained by matching the GPS data with the road network, and the nodes in the figure represent vehicles. The left figure shows the location of vehicle volume in a small area, and the right figure shows the overall thermal map of Chengdu

In this work, 100 regions are designed for the Chengdu dataset, and each region contains 3-4 blocks. This can accurately reflect the regional traffic conditions, ensure the consistency of data, and facilitate the traffic police departments and relevant staff to carry out targeted traffic control and diversion processing for different regions.

Then we will match the GPS data with different urban areas. First, judge the location of the vehicle by longitude and latitude data, and then lock the time range of the vehicle when passing the area by time data (the time interval set in this work is five minutes. That is, the flow data is the statistics of the number of vehicles circulating within five minutes). Then the vehicle ID is given to the region. In this process, if the GPS data of a vehicle appears several times in the same period in the same area, it will not be calculated repeatedly. After matching the GPS data to the urban area through the above steps, we will count the vehicle IDs in different periods in all areas, so that we can obtain the traffic flow data of all areas at a fixed time interval. The dimension of this data is 100 * 1296. Hereinafter referred to as Dataset 3(D3).

Small range dataset

Globally, the traffic data sources of small and medium-sized cities are relatively single, and most of them are composed of a single traffic violation monitoring dataset at intersections. These monitoring systems often do not form a network to operate independently. To enrich the dataset types and verify the prediction universality of the algorithm, a small range dataset is made in this work. The dataset is collected from the intersection violation monitoring data of small and medium-sized cities in the third and fourth lines of China. The upstream, midstream, and downstream of the main urban trunk roads are selected, and the monitoring data of three consecutive traffic sections constitute the traffic network data.

The original data covers 31 days from August 1, 2021, to August 31, 2021. The characteristics of data include vehicle ID, passing time, lane, vehicle type, etc. To effectively draw the topological structure of the graph, the nodes and edges in the graph network are constructed, and the adjacency matrix of the dataset is obtained. We will set nodes in each direction of the three intersections in the upstream, midstream, and downstream according to the two-way lanes. Through the lanes, we can determine the direction of vehicles, to give the passing data to the corresponding nodes. The specific implementation method is shown in Fig. 3.

In the figure, nodes 1 and nodes 3, 5, and 8 have the relationship of a left turn, straight travel, and right turn

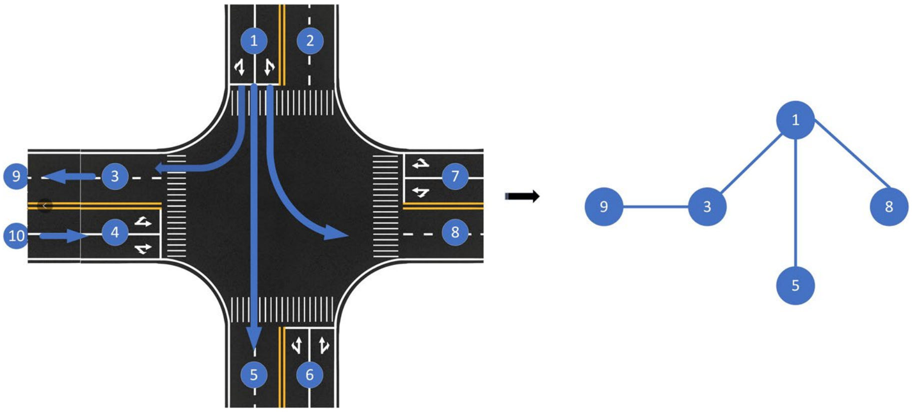

Fig. 3 The graph network construction method of road monitoring data is shown in the figure. The left figure shows the actual traffic situation and the right figure shows the network structure. The vehicles at node 1 in the figure flow to nodes 3, 5, and 8, so they are connected to the corresponding nodes

respectively, so they are connected as the edges of the graph. Nodes 4, 6, and 7 are connected to the corresponding nodes according to the same principle through three modes of transportation: straight, left turn, and right turn. Then the upstream, middle, and downstream nodes are connected in turn to form a graph network as a whole. Finally, determine 5 minutes as a fixed time interval, process the passing data into traffic data, and send it to the relevant nodes to form a 24 * 8928 traffic dataset. Hereinafter referred to as Dataset 4(D4).

Next, make statistics on the relevant characteristics of the four datasets, and the statistical results are shown in Table 1.

Through these four traffic data with different properties and characteristics, we can more comprehensively analyze the precautions and change trends of different traffic data in machine learning. In addition, it can also analyze the universality of the algorithm model and judge the rule of change together with several important parameters.

Data reading and pre-processing

The purpose of data preprocessing is to ensure the correctness and integrity of the data, to facilitate the subsequent feature extraction of the data, and the division of training set and test set. The preprocessing process includes: cleaning out duplicate records in data and removing null values and other abnormal records caused by equipment failures. When it is detected that there is no missing value or obvious abnormal value in the data, the preprocessing is completed.

Adjust data characteristics and model parameters

Set the number of iterations

The number of iterations is a very important parameter in the model training process. The number of iterations directly affects the prediction speed of the algorithm and the overall efficiency of the algorithm. Too many iterations may lead to overfitting, which will affect the accuracy of the algorithm. Therefore, it is necessary to balance prediction accuracy and prediction speed. While ensuring the accuracy of prediction, reduce the number of iterations as much as possible to improve the prediction speed.

Table 1 Summary of dataset characteristics

	D1	D2	D3	D4
Type	Speed	Speed	Flow	Flow
Range	Large	Large	Medium	Small
Node	Large	Medium	Medium	Small
Time	Large	Medium	Small	Large

To accurately understand the prediction effect of the model and evaluate the performance of the model. In this work, 40 iterations are set for the model under four datasets. Interpolate and predict the model parameters, time matrix, and space matrix generated by each iteration of the test set. After that, calculate the result index to judge the prediction effect of each iteration. Judge the change of prediction effect with the increase of iteration number. Finally, the goal of finding the most efficient iteration times and ensuring the prediction speed of the algorithm is achieved.

Set different missing value proportions

The proportion of missing values in all data directly affects the integrity of the data. Under different proportions of missing values, algorithms often show different effects on data analysis and prediction. When the missing value is high, the interpolation performance of the algorithm is required to be higher.

Therefore, this paper is designed for four datasets. Through non-random deletion, different percentages of missing values are set, including 10%, 20%, 30%, 40%, 50%, 60%, and 70%. Then divide the data into a training set and test set. The training set containing missing values is given to the model for iterative training. After obtaining the corresponding model parameters, input the test set to obtain the relevant results of prediction and interpolation under certain missing values. Finally, the change in algorithm prediction performance under different missing values is analyzed.

Lag time characteristics

The lag time characteristics represent the time-related topology, that is, based on what period in the past, the future time can be predicted. The wider the selected time range is, the more time features will be covered, but it will reduce the prediction speed, increase the complexity of the model, and may also cause gradient explosion or gradient disappearance, which will affect the prediction accuracy.

In this work, it is necessary to consider the selection of lag time characteristics, covering both the trend of past time and the characteristics of the current time. Because most of the sequence data have obvious daily trends, in this work, we set several different strategies with lag time characteristics, as shown in Fig. 4.

Scheme A represents a small period in the past. Scheme B represents a long time in the past. Scheme C represents the collection of Scheme A and yesterday at the same time. Scheme D represents the collection of Scheme A and the next historical period at the same time yesterday. After the test set is input, each time the lag time characteristics set in the model are given

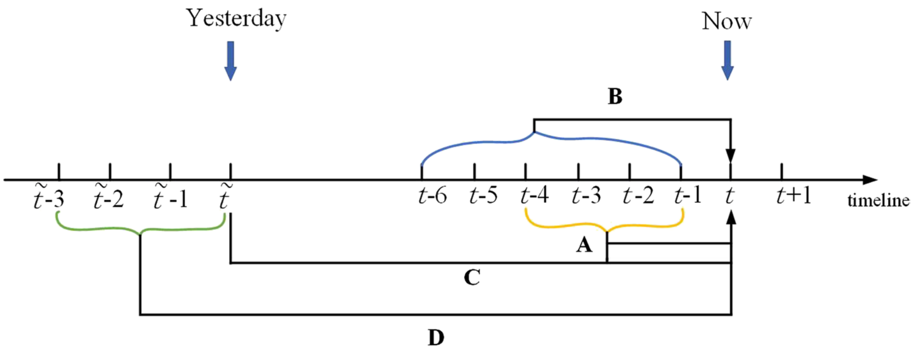

Fig. 4 The figure shows the four lag time characteristics set in this work, including the combination of the past period and the previous day. Scheme A represents a small period in the past. Scheme B represents a long time in the past

as a basis to predict the data results at the next time, and the final prediction and interpolation results are obtained by stepping.

Hidden factor dimension

The dimensions of hidden factors mainly affect the complexity of the model and the performance and efficiency of the algorithm and affect the sparsity of the matrix. The implicit factor is an important parameter in matrix decomposition, which is related to the grasp of spatial characteristics of Spatiotemporal data and the extraction of temporal characteristics. Selecting an appropriate hidden factor dimension can effectively extract features while reducing the dimensions of the input algorithm model, and effectively aggregate with LSTM and other related time feature extraction methods. Set the dimensions to 10, 20, 30, 40, and 50, and conduct experiments under four datasets to obtain relevant prediction results.

It should be noted that the selection of hidden factor dimension is generally much lower than the number of sensors and the number of periods. This can ensure the sparsity of the matrix, so that the matrix will not suffer from small signal fluctuations, resulting in large output differences. When the model is faced with data with high missing values, it can still be effectively trained and predicted.

Divide the training set and test set to train in the algorithm model

Through the above steps, the corresponding processed dataset is obtained. The dataset is divided into a training set and a test set in proportion, and the training set is given to the algorithm for training. After the training, input the test set into the trained algorithm model to obtain the corresponding prediction and interpolation results.

Evaluate relevant indicators and analyze the changing trend

Finally, by comparing the predicted results with the actual observations, the relevant metrics of the experiment are obtained, and the changing trend of the metrics is analyzed by plotting.

Evaluation metrics

Because the traffic data is a time series regression prediction. Therefore, some metrics often used by relevant researchers in regression prediction are selected in this work, including (1) Mean Absolute Error (MAE), (2) Mean Square Error (MSE), (3) Root Mean Square Error (RMSE), and (4) R Square (R2). Taking these indexes as the performance evaluation metrics of the prediction results of this work, to compare with other relevant research results in this field. The corresponding formula is as follows:

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|(y_i - \hat{y}_i)| \tag{4}$$

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \tag{5}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{6}$$

$$R^2 = 1 - \frac{\sum_i(\hat{y}_i - y_i)^2}{\sum_i(\hat{y}_i - \bar{y}_i)^2} \tag{7}$$

Results

This section describes data, practical procedures, and the way we conduct experiments.

Iteration times experimental results

The experimental results are obtained by iterating 30 times for the four datasets respectively and calculating the metrics for the prediction results obtained in each iteration, as shown in the Fig. 5 below.

The experimental results show that both D1 and D2 have significantly improved the prediction effect of the algorithm after the previous iterations, while after 5 iterations, the value of R2 starts to converge significantly. After 20 iterations, the R2 curves of the two datasets are based on smoothing, where the value of R2 in D1 converges to about 0.75, and the value of R2 in D2 converges to about 0.8. Therefore, the prediction speed of the algorithm for D1 and D2 is relatively fast and the prediction accuracy is good.

In D3, the value of R2 is affected by the number of iterations, which is a curve showing an obvious upward trend. In the first 30 times, with the increase of iteration times, the prediction effect of the algorithm is significantly improved. The value of R2 increased from about 0.1 to about 0.8. It shows that the iteration number of the algorithm is very sensitive to the data type of D3.

Finally, it can be found from the R2 value curve of D4 that the value of R2 has improved in the first five iterations, but after that, along with the increase in the number of iterations, the value of R2 has fluctuated around 0.5. Therefore, it can be judged that the number of iterations has not greatly affected the prediction effect of D4.

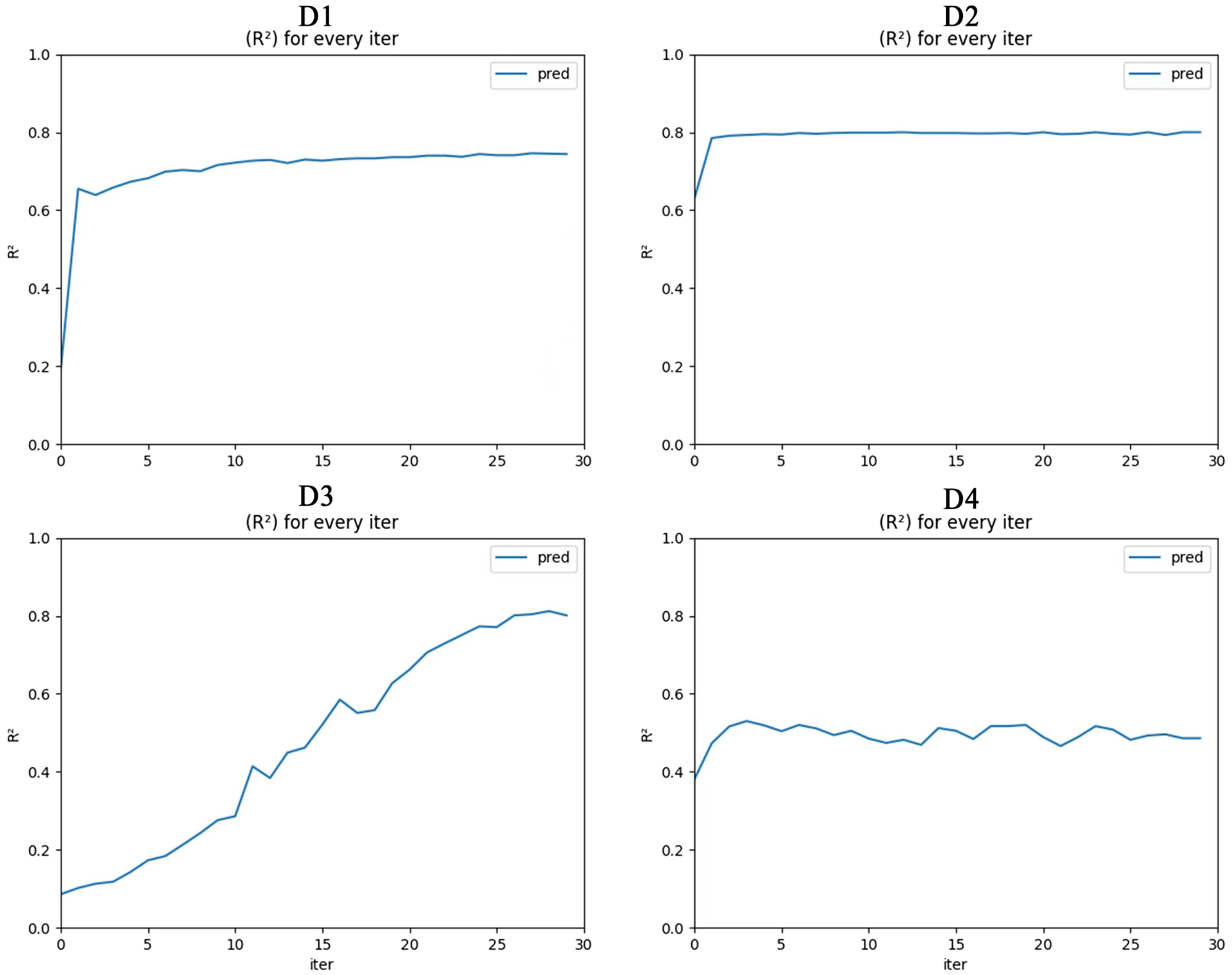

Fig. 5 The figure shows the experimental results of the iterative experiment. The four datasets lead to different performances. D1 and D2 give good predictions and fast convergence speed. The prediction of D3 is good, but the convergence speed is slow. The performance of D4 under this algorithm is normal compared to others

Table 2 Experimental results of missing value

	D1				D2				D3				D3				D4			
Missing	**MSE**	**MAE**	**RMSE**	**R2**	**MSE**	**MAE**	**RMSE**	**R2**	**MSE**	**MAE**	**RMSE**	**R2**	**MSE**	**MAE**	**RMSE**	**R2**	**MSE**	**MAE**	**RMSE**	**R2**
10%	27.6	3.67	5.25	0.691	20	3.24	4.47	0.802	552	17.8	23.5	0.9	552	17.8	23.5	0.9	21.5	3.02	4.64	0.527
20%	28.4	3.72	5.33	0.681	20.2	3.26	4.49	0.8	984	20.9	31.4	0.821	984	20.9	31.4	0.821	21.4	2.96	4.63	0.53
30%	30.1	3.74	5.49	0.662	20.1	3.26	4.49	0.8	757	20.2	27.5	0.862	757	20.2	27.5	0.862	22.1	2.96	4.70	0.514
40%	26.1	3.46	5.10	0.708	20.0	3.23	4.47	0.802	1866	30.6	43.2	0.661	1866	30.6	43.2	0.661	21.7	2.95	4.65	0.524
50%	26.2	3.50	5.12	0.706	20.9	3.36	4.57	0.793	3218	40.7	56.7	0.416	3218	40.7	56.7	0.416	22.6	3.01	4.76	0.502
60%	26.6	3.42	5.15	0.703	20.9	3.36	4.57	0.793	5147	48.3	71.7	0.065	5147	48.3	71.7	0.065	28.6	3.32	5.35	0.371
70%	33.2	3.76	5.77	0.628	21.4	3.42	4.63	0.787	6237	51.6	78.7	0.025	6237	51.6	78.7	0.025	33.3	3.46	5.77	0.269

Missing value proportion experimental results

The prediction indexes of the four datasets designed in this work under different missing value ratios are shown in the Table 2 below.

The overall effect of D1 is good. When the proportion of missing values is less than 70%, the results of all indicators are similar, and the value of R2 is about 0.7. When the proportion of missing values reaches 40%, the prediction effect is the best. When the proportion of missing values exceeds 60%, the prediction effect begins to decline.

The overall prediction result of D2 is also good. When the proportion of missing values reaches 10% and 40%, the prediction effect is the best. When the proportion of missing values is less than 40%, the value of R2 is more than 0.8, and when the proportion of missing values is more than 40%, it starts to decline slightly.

For D3, except for some fluctuations when the proportion of missing values is about 30%, the prediction effect decreases significantly with the increase of the proportion of missing values. The best prediction result is that there are 10% missing values, and the R2 value reaches 0.9. When the proportion of missing values exceeds 60%, the value of R2 decreases to less than 0.1.

For D4, with the increase in the proportion of missing values, the prediction effect tends to fluctuate and decline. After the proportion of missing values exceeds 40%, the prediction effect of the algorithm starts to decline significantly.

In a word, when there are few missing values, the algorithm has a good prediction effect for different datasets, while when the missing values are large, the effect under the traffic speed dataset is still good, and the traffic flow dataset is poor.

Lag time characteristic test results

The data under four different lag time strategies are given to the algorithm for prediction, and the experimental results are as follows in Table 3.

The prediction result of D1 is generally good. Scheme A has the best prediction effect, and R2 is 0.738. The second prediction effect of Scheme B is that R2 reaches 0.735. The results of each index of Scheme C and Scheme D are similar, and the value of R2 is about 0.71.

The prediction result of D2 is better than that of D1. Under the four schemes, the value of R2 reaches about 0.8, and the fluctuation is small. The prediction effect of Scheme C is the best, and the value of R2 is 0.799.

The related prediction result of D3 is the largest change in the four datasets. Under different schemes, the prediction results fluctuate greatly. Among them, the effect of Scheme B is the best, and the value of R2 reaches 0.825. Scheme C takes second place, and the value of R2 is 0.661. The effect of Scheme A and Scheme D is poor, and R2 values are 0.577 and 0.554 respectively.

The overall prediction effect of D4 is poor in the four datasets. Under the four schemes, the value of R2 always fluctuates at 0.5. Scheme A has the lowest R2 value, 0.492. Scheme C has the highest R2 value of 0.516.

Table 3 Experimental results of lag time characteristics

	D1				D2				D3				D4			
Lag	MAE	MSE	RMSE	R2	MAE	MSE	RMSE	R2	MAE	MSE	RMSE	R2	MAE	MSE	RMSE	R2
Scheme A	3.34	23.4	4.84	0.738	3.34	21.0	4.58	0.792	34.8	2332	48.3	0.577	3.02	23.1	4.81	0.492
Scheme B	3.31	23.7	4.86	0.735	3.36	21.2	4.61	0.79	23.3	964	31.1	0.825	3.02	22.1	4.7	0.514
Scheme C	3.46	25.9	5.09	0.71	3.27	20.3	4.51	0.799	31.3	1869	43.2	0.661	2.98	22.0	4.69	0.516
Scheme D	3.52	25.4	5.04	0.715	3.27	20.4	4.52	0.797	34.8	2454	49.5	0.554	3.03	22.8	4.77	0.499

Table 4 Influence of different dimensions of the hidden factor

	D1				D2				D3				D4			
Dim	MAE	MSE	RMSE	R2	MAE	MSE	RMSE	R2	MAE	MSE	RMSE	R2	MAE	MSE	RMSE	R2
10	3.74	32.4	5.70	0.637	3.33	20.8	4.56	0.793	28.5	1528	39.1	0.723	2.89	21.3	4.61	0.532
20	3.30	26.4	5.14	0.704	3.27	20.3	4.51	0.798	27.1	1381	37.2	0.749	2.93	21.3	4.62	0.531
30	3.29	24.4	4.94	0.727	3.31	20.4	4.52	0.797	24.0	1085	32.9	0.803	2.95	23.0	4.8	0.493
40	3.47	25.9	5.09	0.71	3.30	20.3	4.51	0.798	29.2	1636	40.5	0.703	2.98	24	4.90	0.472
50	3.38	24.4	4.94	0.727	3.24	20.0	4.47	0.802	21.3	823	28.7	0.851				

Hidden factor dimension experiment results

To explore the influence of the dimensions of hidden factors in matrix decomposition on the results, the hidden factors are set as the only variable in this work. The relevant experimental results are shown in the following Table 4.

In the experimental results of D1, the prediction results increase slightly with the increase of dimensions. When the dimensions are 20 and 40, the prediction results are the best, the value of R2 reaches 0.727, and when the dimensions are 10, the effect is the worst, and the value of R2 is 0.637.

The prediction results in D2 are good, with R2 reaching more than 0.79. With the change of hidden factor dimensions, the prediction results do not fluctuate significantly, and the fluctuations in the four datasets are the smallest.

The relevant experimental results of D3 are the best of the four datasets. Among them, when the dimension of the hidden factor is 50, the prediction result is the best, and the value of R2 is 0.85.

For D4, since the number of nodes is the least, the dimensions of hidden factors are not suitable for taking higher values. When the hidden factor dimension is 10, the effect is the best, and the value of R2 is 0.532. Later, when the dimensions of the hidden factor are 20, 30, and 40, the prediction effect decreases step by step.

Analysis and discuss

The overall analysis of the experimental results that the overall prediction effect of D1 is good. Under the influence of multiple parameters, the actual prediction results will improve and converge stably when the parameter settings conform to the characteristics of the data itself. The overall result of D2 is the best. The R2 value remains above 0.8 under a variety of parameters. With the change of parameters, the dataset maintains a high prediction accuracy and has a small fluctuation. The overall prediction effect of D3 fluctuates the most, but the relevant indicators of prediction are also the highest. It shows that the dataset has many characteristics and requires high parameters of the algorithm model. When the model parameters change, the prediction results will change significantly. For D4, the overall prediction effect is average. No matter how the relevant parameters change, the value of R2 is always around 0.5.

Therefore, analyze the causes of these results. First of all, traffic data covers a wider range and fluctuates more widely than speed data, covering the range from zero to hundreds (speed data is generally in the range of 0-100). Therefore, the variance of the traffic data itself is large, which also results in the situation that most of the traffic data-related results are worse than the speed data. Since D3 does not cover the early morning hours and the selected area is downtown, the average and median values of D3 are significantly higher than those of D4. On the other hand, the values of different nodes in D3 at the same time differ greatly, that is, the data variance is greater. Therefore, it can be judged that the greater the variance of the dataset, the stricter the requirements for the model parameters.

The node location selected in D4 is not in the core area of the city, and it contains all-day data. During the period from 23:00 to 6:00 of the next day, the traffic data is generally between 0-10. Compared with D2, D4 has fewer nodes and lower values, and the prediction effect is always different. Therefore, we analyze that the neural network algorithm based on graph structure can often show a good prediction effect when facing data with many nodes and large values. However, for the data with a small number of nodes and small numerical values, the neural network algorithm can not show a good prediction effect.

At the same time, when there are missing values in the data value, the method of matrix decomposition can effectively predict and fill the data. When the variance of the data itself is small, the increase in the proportion of missing values will, to a certain extent, help the algorithm grasp the commonalities in the data, thus enhancing the prediction effect.

The hidden factor work confirmed that the size of the hidden factor dimension and the size of the matrix have a certain degree of regularity. When the prediction effect is good under the four datasets, the dimensions of the hidden factor are within the 20% - 50% range of the matrix dimension. When it is less than 20% or more than 50%, the prediction effect will decline to varying degrees. For example, D1 has the largest number of nodes, up to 323. When the hidden factor dimension is 10, the matrix dimension after decomposition is too small and the matrix is too simple. Although the sparsity of the matrix can be guaranteed, the relevant features of the data cannot be accurately extracted, which leads to the decline of the prediction effect. The hidden factor dimension is lower than the number of nodes to ensure data sparsity, grasp the data commonality, and make the matrix still have a good fitting effect when it contains missing values. Therefore, the matrix decomposition method is more suitable for the data with a high proportion of missing values.

In addition, we checked the data of D4 and found that the sensor equipment in a single direction at one intersection failed, resulting in incomplete data statistics of the node, making the data value of the node low and affecting the overall prediction effect.

Conclusion and future work

In this work, several different types of traffic data are first collected, including floating car data, road monitoring data, and others. Then according to the characteristics of the data itself and the relevant physical location, different data processing methods are proposed, including using longitude and latitude to divide regions and using lanes to determine the driving direction of vehicles. Through these methods, different types of datasets are processed, and the adjacency matrix and feature matrix of the data are successfully obtained, which is ready for the prediction of the algorithm. After that, all datasets are used for prediction from the perspective of four parameters. The prediction results are promising, and the influence degree and change trend of these parameters on the prediction results are judged.

This work has two main contributions. On one hand, the experimental results prove the feasibility of the data preprocessing method, which can be combined with relevant machine learning algorithms to give effective predictions. On the other hand, the experimental results shed light on the variation of the data under several different parameters, the impact of the parameter variation on the prediction results, and the reason. It is convenient for subsequent researchers to clarify various modeling parameters of different types of datasets, and finally achieve the purpose of improving the prediction effect and speeding up the prediction speed. In the future, we will continue to focus on the analysis and prediction of various spatiotemporal data. Further research will be carried out from the aspects of data acquisition, feature engineering, and algorithm modeling.

Acknowledgements
We would like to thank the reviewers for their detailed suggestions they have indeed made a great contribution to this work.

Authors' contributions
TS guided the overall procedure. LZ completed the experiment. RG drafted the paper. SL assisted in writing and experimentation. BS revised the content and guided detailed directions. All authors read and approved the final manuscript.

Funding
This work is supported by National Key R&D Program of China 2018AAA0101703, Shandong Key Technology R&D Program 2019JZZY021005 and Natural Science Foundation of Shandong ZR2020MF067. Jinan City-School Integration Development Strategy Project (No. JNSX2023015).

Availability of data and materials
Data and materials are available on reasonable requests.

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare that there are no competing interests.

Received: 13 December 2022 Accepted: 5 November 2023
Published online: 08 April 2024

References

1. Yan S, Shao H, Min Z, Peng J, Cai B, Liu B (2023) Fgdae: a new machinery anomaly detection method towards complex operating conditions. Reliab Eng Syst Saf 236:109319. https://doi.org/10.1016/j.ress.2023.109319, https://www.sciencedirect.com/science/article/pii/S0951832023002338
2. Chen M, Shao H, Dou H, Li W, Liu B (2023) Data augmentation and intelligent fault diagnosis of planetary gearbox using ILoFGAN under extremely limited samples. IEEE Trans Reliab 72(3):1029–1037. https://doi.org/10.1109/TR.2022.3215243
3. Chen X, Shao H, Xiao Y, Yan S, Cai B, Liu B (2023) Collaborative fault diagnosis of rotating machinery via dual adversarial guided unsupervised multi-domain adaptation network. Mech Syst Signal Process 198:110427. https://doi.org/10.1016/j.ymssp.2023.110427, https://www.sciencedirect.com/science/article/pii/S0888327023003357
4. Vermesan O, Friess P (2013) Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems. River Publishers
5. Huang J, Lv B, Wu Y, Chen Y, Shen X (2021) Dynamic admission control and resource allocation for mobile edge computing enabled small cell network. IEEE Trans Veh Technol 71(2):1964–1973
6. Qureshi KN, Abdullah AH (2013) A survey on intelligent transportation systems. Middle-East J Sci Res 15(5):629–642
7. Shao H, Li W, Cai B, Wan J, Xiao Y, Yan S (2023) Dual-threshold attention-guided GAN and limited infrared thermal images for rotating machinery fault diagnosis under speed fluctuation. IEEE Trans Ind Inform 19(9):9933–9942. https://doi.org/10.1109/TII.2022.3232766
8. Huang J, Gao H, Wan S, Chen Y (2023) AoI-aware energy control and computation offloading for industrial IoT. Future Gener Comput Syst 139:29–37
9. Jamson AH, Merat N, Carsten OM, Lai FC (2013) Behavioural changes in drivers experiencing highly-automated vehicle control in varying traffic conditions. Transp Res Part C Emerg Technol 30:116–125
10. Darwish TS, Bakar KA (2018) Fog based intelligent transportation big data analytics in the internet of vehicles environment: Motivations, architecture, challenges, and critical issues. IEEE Access 6:15679–15701
11. Mahdavinejad MS, Rezvan M, Barekatain M, Adibi P, Barnaghi P, Sheth AP (2018) Machine learning for Internet of Things data analysis: a survey. Digit Commun Netw 4(3):161–175
12. Chen Y, Zhao J, Zhou X et al (2023) A distributed game theoretical approach for credibility-guaranteed multimedia data offloading in MEC. Inf Sci. https://doi.org/10.1016/j.ins.2023.119306
13. Li D, Deng L, Cai Z (2020) Intelligent vehicle network system and smart city management based on genetic algorithms and image perception. Mech Syst Signal Process 141:106623
14. Yang F, Wang S, Li J, Liu Z, Sun Q (2014) An overview of internet of vehicles. China Commun 11(10):1–15
15. Chen C, Li H, Li H, Fu R, Liu Y, Wan S (2022) Efficiency and fairness oriented dynamic task offloading in internet of vehicles. IEEE Trans Green Commun 6(3):1481-93. https://doi.org/10.1109/TGCN.2022.3167643
16. Huang J, Wan J, Lv B, Ye Q et al (2023) Joint computation offloading and resource allocation for edge-cloud collaboration in internet of vehicles via deep reinforcement learning. IEEE Syst J 17(2):2500–2511
17. Chen C, Zeng Y, Li H, Liu Y, Wan S (2023) A multihop task offloading decision model in MEC-enabled internet of vehicles. IEEE Internet Things J 10(4):3215-30. https://doi.org/10.1109/JIOT.2022.3143529
18. Papageorgiou M, Diakaki C, Dinopoulou V, Kotsialos A, Wang Y (2003) Review of road traffic control strategies. Proceedings of the IEEE 91(12):2043-67. https://doi.org/10.1109/JPROC.2003.819610.
19. Chen Y, Zhao J, Hu J et al (2023) Distributed task offloading and resource purchasing in NOMA-enabled mobile edge computing: Hierarchical game theoretical approaches. ACM Trans Embed Comput Syst. https://doi.org/10.1145/3597023
20. Srinivasan D, Choy MC, Cheu RL (2006) Neural networks for real-time traffic signal control. IEEE Trans Intell Transp Syst 7(3):261–272
21. Wen L, Kenworthy J, Guo X, Marinova D (2019) Solving traffic congestion through street renaissance: a perspective from dense Asian cities. Urban Sci 3(1):18

22. Nicopolitidis P, Obaidat MS, Papadimitriou GI, Pomportsis AS (2003) Wireless Networks. John Wiley & Sons
23. Chen Y, Zhao J, Wu Y et al (2022) Qoe-aware decentralized task offloading and resource allocation for end-edge-cloud systems: a game-theoretical approach. IEEE Trans Mob Comput. https://doi.org/10.1109/TMC.2022.3223119
24. Schlomer GL, Bauman S, Card NA (2010) Best practices for missing data management in counseling psychology. J Couns Psychol 57(1):1
25. Ozbayoglu M, Kucukayan G, Dogdu E (2016) A real-time autonomous highway accident detection model based on big data processing and computational intelligence. In: 2016 IEEE International Conference on Big Data (Big Data), IEEE, pp 1807–1813
26. Wang L, etc XD, (2023) A review of urban air mobility-enabled intelligent transportation systems: Mechanisms, applications and challenges. J Syst Archit 141:102902. https://doi.org/10.1016/j.sysarc.2023.102902, www.sciencedirect.com/science/article/pii/S1383762123000814
27. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2008) The graph neural network model. IEEE Trans Neural Netw 20(1):61–80
28. Chen Y, Hu J, Zhao J, Min G (2023) Qos-aware computation offloading in leo satellite edge computing for IoT: a game-theoretical approach. Chin J Electron
29. Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY (2020) A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst 32(1):4–24
30. Zhang S, Tong H, Xu J, Maciejewski R (2019) Graph convolutional networks: a comprehensive review. Comput Soc Netw 6(1):1–23
31. Velickovic P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y (2017) Graph attention networks. Stat 1050:20
32. Hamilton WL, Ying R, Leskovec J (2017) Inductive Representation Learning on Large Graphs. arXiv:1706.02216
33. Atluri G, Karpatne A, Kumar V (2018) Spatio-temporal data mining: a survey of problems and methods. ACM Comput Surv CSUR 51(4):1–41
34. Zaremba W, Sutskever I, Vinyals O (2014) Recurrent neural network regularization. ArXiv Prepr arXiv:1409.2329
35. Liu F, Huang J, Wang X (2023) Joint task offloading and resource allocation for device-edge-cloud collaboration with subtask dependencies. IEEE Trans Cloud Comput 11(3):3027–3039. https://doi.org/10.1109/TCC.2023.3251561
36. Zhao L, Zhang E, Wan S, Hawbani A, Al-Dubai AY, Min G, Zomaya AY Meson: a mobility-aware dependent task offloading scheme for urban vehicular edge computing. https://doi.org/10.1109/tmc.2023.3289611
37. Sak H, Senior AW, Beaufays F (2014) Long short-term memory recurrent neural network architectures for large scale acoustic modeling. Interspeech. p. 338-342
38. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. ArXiv Prepr arXiv:1406.1078
39. Saad W, Han Z, Debbah M, Hjorungnes A, Basar T (2009) Coalitional game theory for communication networks. IEEE Signal Process Mag 26(5):77–97
40. Li Y, Tarlow D, Brockschmidt M, Zemel R (2015) Gated graph sequence neural networks. ArXiv Prepr arXiv:1511.05493
41. Ang LM, Seng KP (2016) Big sensor data applications in urban environments. Big Data Res 4:1–12
42. Chen PH, Lin CJ, Schölkopf B (2005) A tutorial on ν-support vector machines. Appl Stoch Models Bus Ind 21(2):111–136
43. Miettinen P (2009) Matrix decomposition methods for data mining: computational complexity and algorithms. University of Helsinki, Helsinki. 170 p. (Series of publications / Department of Computer Science, University of HelsinkiA; 2009-4)
44. Yu H, Rao NS, Dhillon IS (2016) Temporal regularized matrix factorization for high-dimensional time series prediction. Neural Information Processing Systems
45. Merris R (1994) Laplacian matrices of graphs: a survey. Linear Algebra Appl 197:143–176
46. Yang JM, Peng ZR, Lin L (2021) Real-time spatiotemporal prediction and imputation of traffic status based on LSTM and Graph Laplacian regularized matrix factorization. Transp Res Part C Emerg Technol 129:103228

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.