Traffic Network Flow Estimation Based On Social Network Influence Model

Final Report

11812804 董　正

11810419 王焕辰

11811305 崔俞崧

Supervisior: 宋轩

Department of Computer Science and Engineering

Jan. 2022

Contents

1 Preliminaries

1.1 Review

In this semester, we will try to build a traffic flow estimation system based on graph neural network and social network influence model. We have changed the system design a little. Currently, the structure of the whole system is

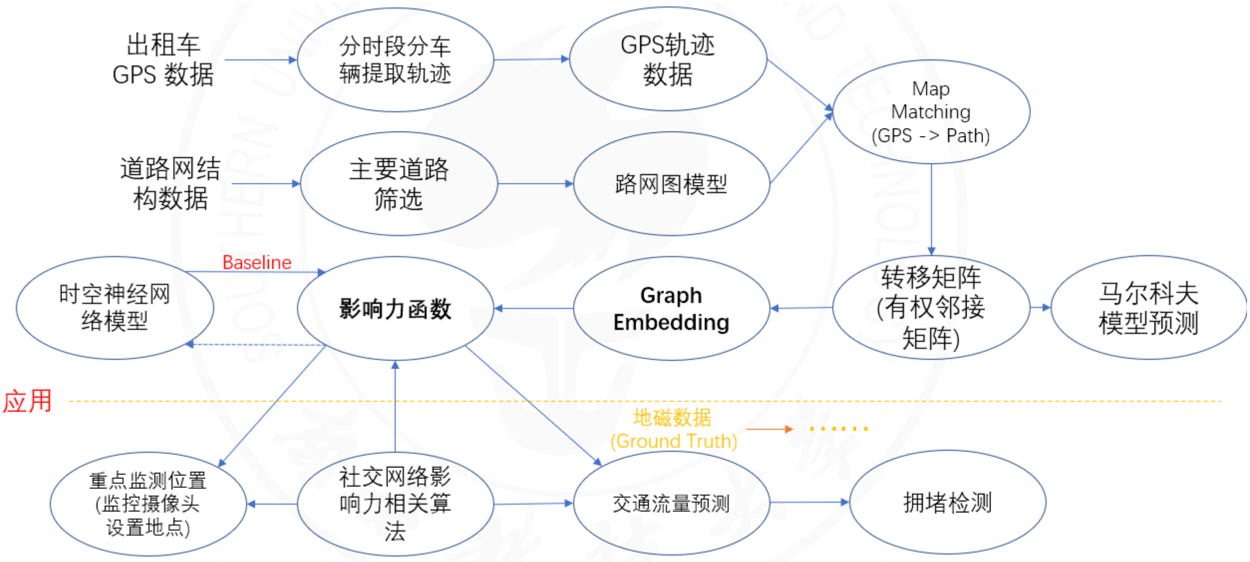

Figure 1: System Structure

1. Process taxi GPS data to get tracks
2. Process road network data to get a basic graph model
3. Match tracks to each road and get the adjacent matrix of the graph
4. Try a simple prediction based on Markov model
5. **Graph embedding**
6. **Influence function design**
7. Combine spatial-temporal models and use them as baseline
8. Combine social network influence algorithms to predict traffic network flow, use geomagnetic data as one of the ground truth
9. Applications: traffic surveillance camera position and traffic jam detection

1.2 Report Contents

Breifly, we will state our work in this report as

- AAAI21: Traffic Flow Prediction with Vehicle Trajectories 董正 & 崔俞崧

- LibCity Exploring 董正 & 崔俞崧
- Geomagnetic data Network Construction & Basic regression Prediction 王焕辰

2 AAAI21: Traffic Flow Prediction with Vehicle Trajectories

In this part, we will introduce a paper in AAAI [1] whose work is very similar to ours. What's more, we will make a comparsion on design ideas between thier and ours.

2.1 Trajectory Transition

- Model trajectory transition as a Markov process.
- Calculate transition matrix for each time interval, the design and calculation process is exactly same as ours.
- However, 1^{st} order transition matrix cannot capture high-dimensional transition information. Therefore, we decided to use graph embedding on trajectory.

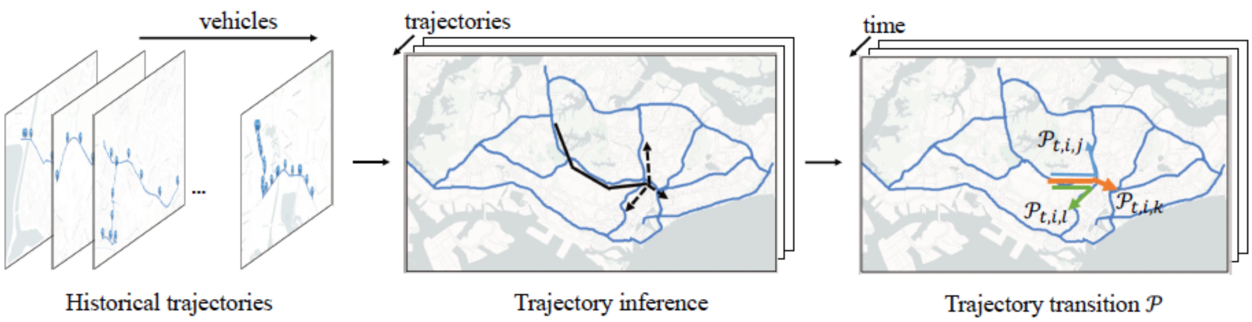

Figure 2: Trajectory Transition Model

2.2 Traffic Demand

For spatial modeling, they use graph propagation from Graph Convolutional Networks to simulate the transition of vehicles along the road network. Then perform graph propagation in d hops, resulting in a graph of traffic demand for each hop. For each input time interval t, the traffic demand is

$$D_t = GraphProp(X_t, \mathcal{P}_t^T; d) = [X_t || \mathcal{P}_t^T X_t || (\mathcal{P}_t^T)^2 X_t || \ldots || (\mathcal{P}_t^T)^d X_t]$$

As we can see, it is a Markov propagation.

For temporal modeling, they use traffic status for attention. Traffic status refers to the overall traffic volume in the neighboring of each road segment. If the traffic status is congested around a road segment (i.e., high volume of flows in the neighboring road segments), the propagation of flows along that road segment should be slow, and vice versa.

For each time interval, they applied bidirectional graph propagation method to get current traffic statusof each neighbor.

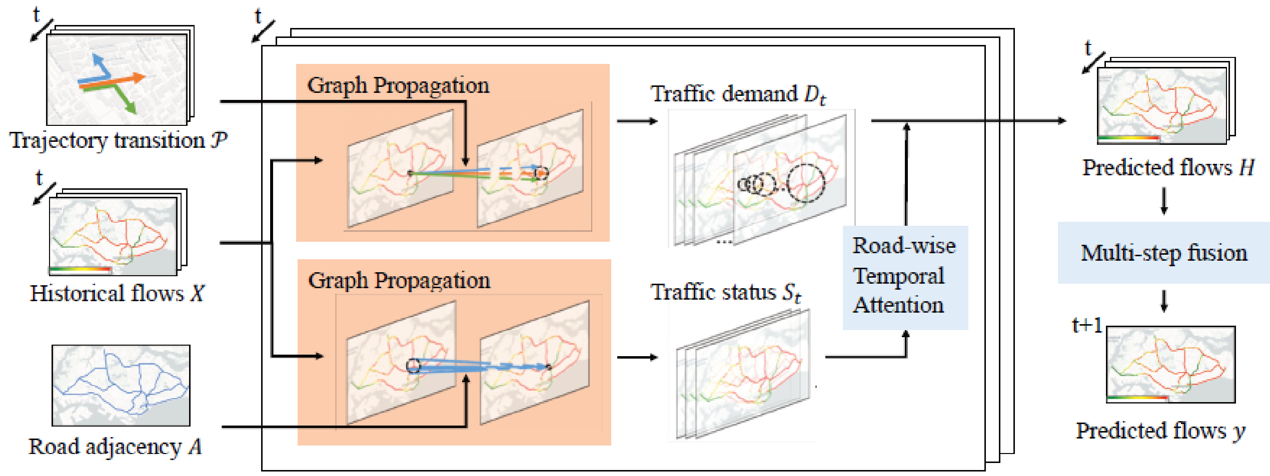

Figure 3: Model Structure

2.3 Performance

Method	Overall			Peak hours			Non-peak hours			MRT breakdown		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
HA	33.74	0.34	52.58	36.83	0.25	55.02	32.53	0.28	48.67	**40.07**	0.27	**59.34**
MA	31.55	0.35	47.69	36.14	0.26	53.18	28.18	0.27	39.41	44.85	0.30	71.43
VAR	29.27	0.33	43.22	34.23	0.24	49.71	28.10	0.26	39.28	40.68	**0.27**	64.41
RF	29.26	0.33	43.38	34.13	**0.24**	49.75	**27.53**	**0.26**	**38.53**	42.28	0.28	66.53
T-GCN	31.12	0.35	45.69	36.57	0.27	52.91	30.03	0.29	41.53	42.38	0.30	67.39
STGCN	29.88	0.33	44.51	34.86	0.24	50.86	27.94	0.27	39.05	42.19	0.28	66.40
DCRNN	**29.01**	**0.31**	**43.12**	**33.74**	0.25	**48.88**	27.75	0.27	38.74	40.39	0.28	64.28
TrGNN-	27.34	0.31	40.05	31.35	0.23	45.11	26.61	0.26	37.20	38.57	0.27	59.53
TrGNN	**26.43**	**0.30**	**38.65**	**29.81**	**0.23**	**42.62**	**25.65**	**0.25**	**35.68**	**34.56**	**0.25**	**54.31**
%diff	-9%	-5%	-10%	-12%	-6%	-13%	-7%	-4%	-7%	-14%	-8%	-8%

Numbers in bold denote the best baseline performance and the best performance.
%diff denotes the error reduction of TrGNN from the best baseline performance.

Figure 4: Performance

As we can see, this model achieves a much better result than SOTA GCNs. However, the baseline models, i.e. T-GCN, STGCN and DCRNN are designed for speed prediction. We doubt that why the author did not choose flow prediction models.

2.4 Preprocessing

For raw GPS data, the author applied these methods to convert it to flow data:

1. Map Matching: Hidden Markov Map Matching (HMMM)

   HMMM maps a whole trajectory to road network and needs road connectivity information.

2. Trajectory Split

   - GPS is off for over 10 minutes
   - Driver stay on same road for 2 minutes

  - No path between two consecutive GPS points

3. Trajectory Recovery

4. For each two consecutive GPS points, run Dijkstra algorithm to find shortest path.

5. Flow Aggregation: aggregate on 15 minutes time interval

The preprocessing methods are worthy to learn, thus, we applied many similar procedure when process our data.

3 LibCity Exploring

LibCity [2] is a unified, comprehensive, and extensible library, which provides researchers with a credible experimental tool and a convenient development framework in the traffic prediction field.

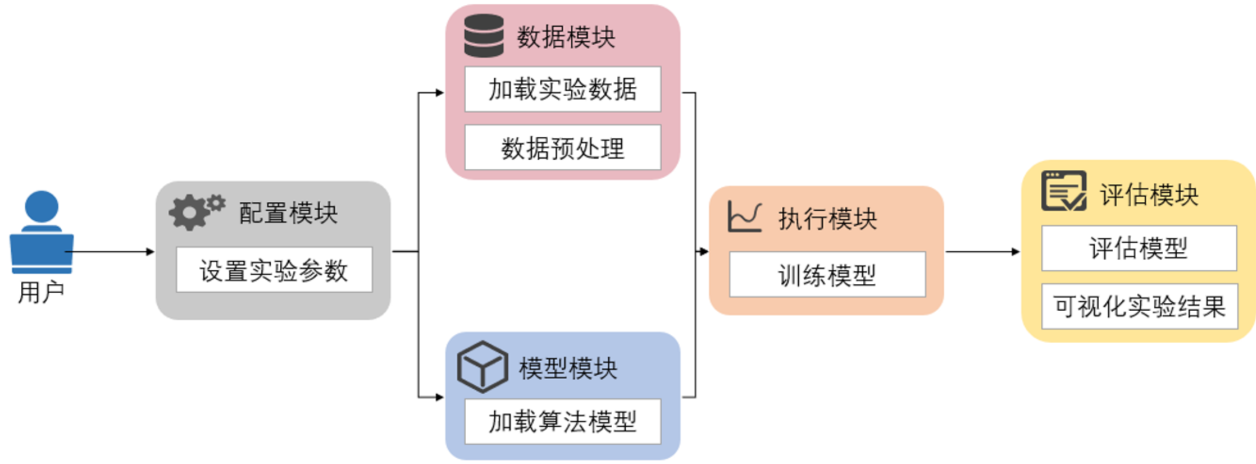

Figure 5: LibCity Structure

3.1 Atomic Files

Atomic files are `.csv` files defined by LibCity. Every raw data should be converted to these atomic files, which gurantees the uniformity of different raw data.

Filename	Content
`xxx.geo`	Store geographic entity attribute information.
`xxx.usr`	Store traffic user information.
`xxx.rel`	Store the relationship information between entities, such as road networks.
`xxx.dyna`	Store traffic condition information.
`xxx.ext`	Store external information, such as weather, temperature, etc.
`config.json`	Used to supplement the description of the above table information.

The core file is `.dyna` which stores traffic state data for traffic prediction, or trajectory data for map matching.

3.2 LibCity Map Matching

The matching model we chose is HMMM.

1. Convert raw GPS data to atomic files.

    - `.geo`: Road ID and geometry information.
    - `.rel`: Adjacency list.
    - `.usr`: Taxi ID.

- `.dyna`: Trajectories, where we applied trajectory split methods.

dyna_id	type	time	entity_id	traj_id	coordinates
0	trajectory	2019-12-02T00:10:30Z	15876	0	[114.06776, 22.550152]
1	trajectory	2019-12-02T00:11:32Z	15876	0	[114.06859, 22.54198]
2	trajectory	2019-12-02T00:12:12Z	15876	0	[114.07125, 22.542738]
3	trajectory	2019-12-02T00:51:17Z	15876	1	[114.04781, 22.539036]
4	trajectory	2019-12-02T00:51:47Z	15876	1	[114.04774, 22.538887]
...	...	...	...	...	...
95	trajectory	2019-12-02T15:43:25Z	15876	7	[114.05783, 22.531305]
96	trajectory	2019-12-02T15:51:55Z	15876	7	[114.05137, 22.53659]
97	trajectory	2019-12-02T15:55:40Z	15876	7	[114.05117, 22.53749]
98	trajectory	2019-12-02T15:57:38Z	15876	7	[114.051216, 22.539156]
99	trajectory	2019-12-02T15:59:26Z	15876	7	[114.05119, 22.545998]

However, we also found some shortcomings:

- Lack APIs for `OSM` and `NetworkX`.
- Different raw data need totally different convert scipts.
- .csv format leads to lots of duplicated information.
- Lack of documents.

2. Run HMMM model.

   - It is convenient. If the structure of atomic files are correct, we can run directly by a simple command.
   - LibCity provides a set of parameters.
   - LibCity outputs very detailed logs.

   Still, the shortcomings are:

   - Bugs in array index, eg. `while a[k] and k < len(a)-1`.
   - Does not check `null` or empty sets.
   - Wide usage of time-costing functions, eg. `DataFrame.iterrows()`.
   - Does not support multithreading.

3.3 Traffic Flow Prediction Baseline

1. Convert matched trajectory data to `.dyna` file.

   Here we used the data for calculating transition matrix and graph embedding in our last report.

   It contains 16153 roads, and the spatial range is whole Shenzhen, the time range is Mon. to Fri.

2. Flow Aggregation: the time interval is 15min, and 96 intervals per day.

dyna_id	type	time	entity_id	flow
0	state	2019-12-02T00:00:00Z	0	5
1	state	2019-12-02T00:00:00Z	1	2
2	state	2019-12-02T00:00:00Z	2	9
3	state	2019-12-02T00:00:00Z	3	2
4	state	2019-12-02T00:00:00Z	4	3
...	...	...	...	...
95	state	2019-12-02T00:00:00Z	95	0
96	state	2019-12-02T00:00:00Z	96	0
97	state	2019-12-02T00:00:00Z	97	0
98	state	2019-12-02T00:00:00Z	98	1
99	state	2019-12-02T00:00:00Z	99	0

3. Load model. LibCity auto splits train, vaild and test datasets.

4. Model training. LibCity uses `Ray Tune` to adjust parameters automatically.

5. Model evaluation. LibCity provides many kinds of metrics.

However, the result of our dataset is bad. We only run simple NNs successfully, and GCNs are out of memory when training because our road network is too large.

The evaluation for simple NNs are also not good:

Model	MAE	Masked MAPE	Masked RMSE
AutoEncoder	3.13	0.79	11.57
GRU	5.22	2.01	12.11
LSTM	3.24	0.87	11.45
FNN	2.52	0.75	11.98
Seq2seq	5.44	2.1	12.66

3.4 Future Plan

1. Re-select road network based on raw GPS data. We plan to select about 200 500 roads.

2. Try only one day's data.

3. Try to run GCN baseline successfully first, and then enlarge time duration.

4 Geomagnetic data Network Construction & Basic regression Prediction

4.1 Divide the Area and Build the Road Network

According to the administrative divisions of Shenzhen and the concentration of traffic flow recorded at each monitoring point, select two regions and divide them.

According to the actual geomagnetic detection points recorded in the data, construct the road connection network map of this region and generate an adjacency matrix.

However, the two areas only take each geomagnetic detection point as the graph node, without considering the inflow and outflow, at present. The adjacency matrix of the road network in Futian District is 34×34, with 59 edges in total.

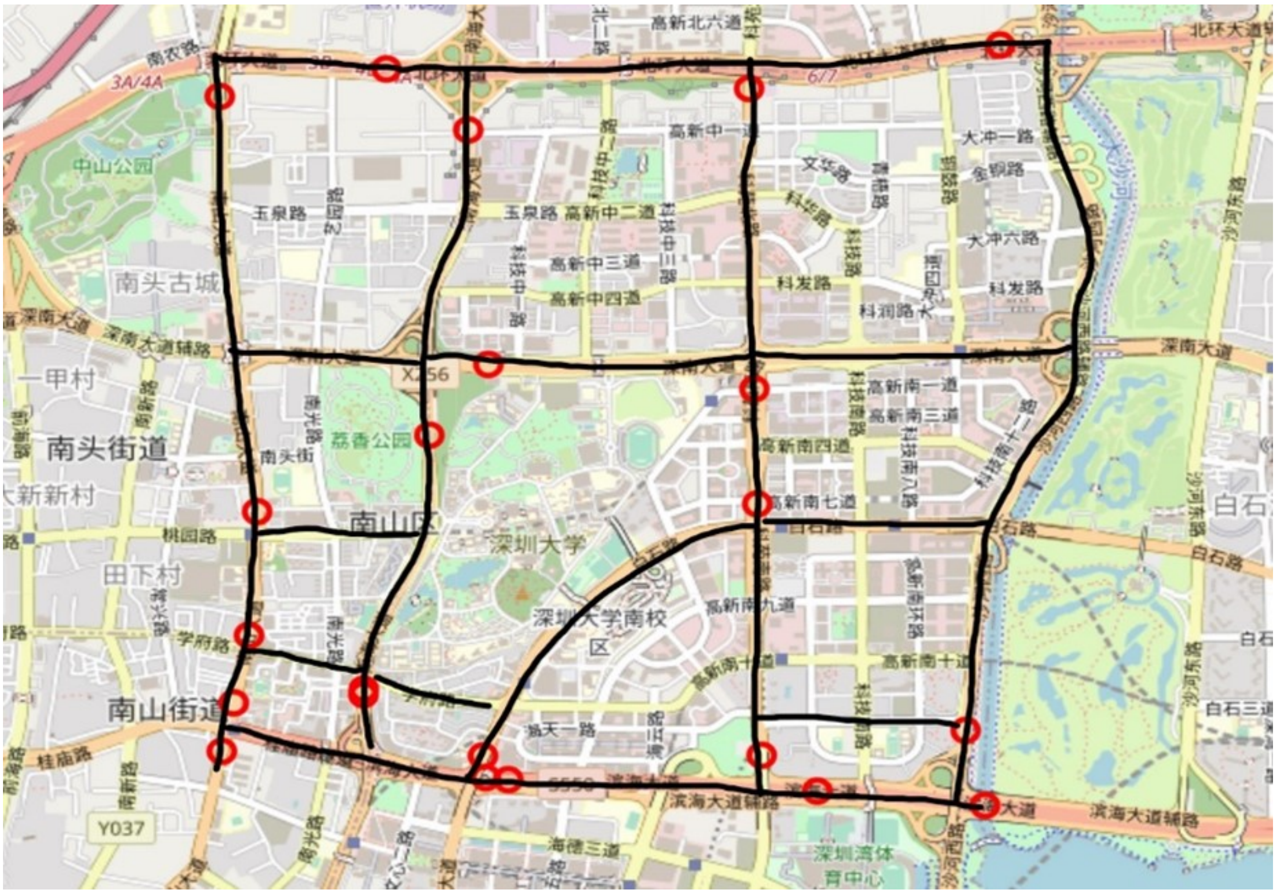

Figure 6: Nanshan district central road network

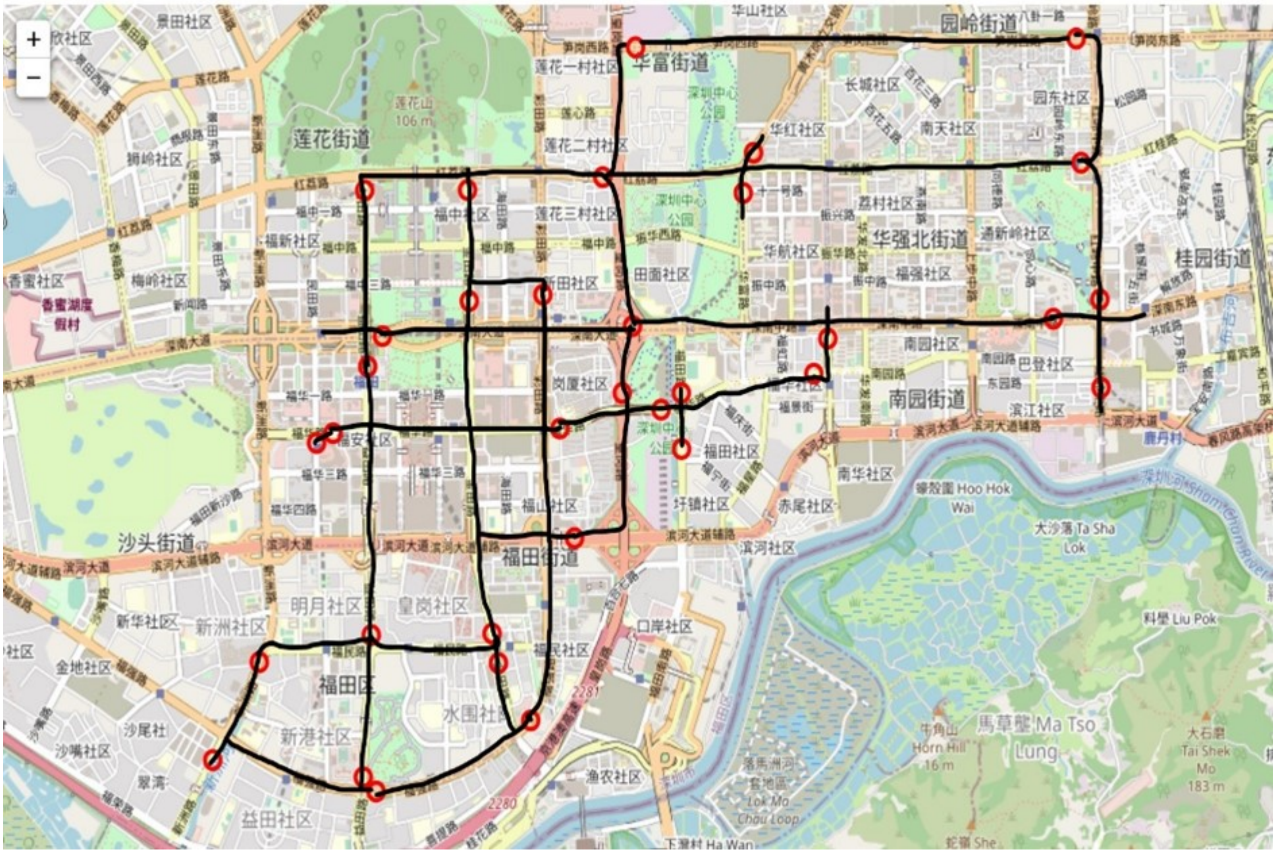

Figure 7: Futian district central road network

4.2 Existing Problems in Network Construction

At present, the matrix only contains connection relation and does not contain lane information. The network is only established by connecting the recorded detection points in the data, without considering lane information and traffic flow direction.

There are data records of the detection point is only 127, not in conformity with the point description provided in 318, In January 2019 to August 2020 data, it has been found after the preprocessing, which causes the original connection relations of intensive figure and become sparse, have to expand the area (such as Futian area had a smaller area should have 45 points). At present, we are communicating with Shenzhen Transportation Bureau to find the cause of missing data.

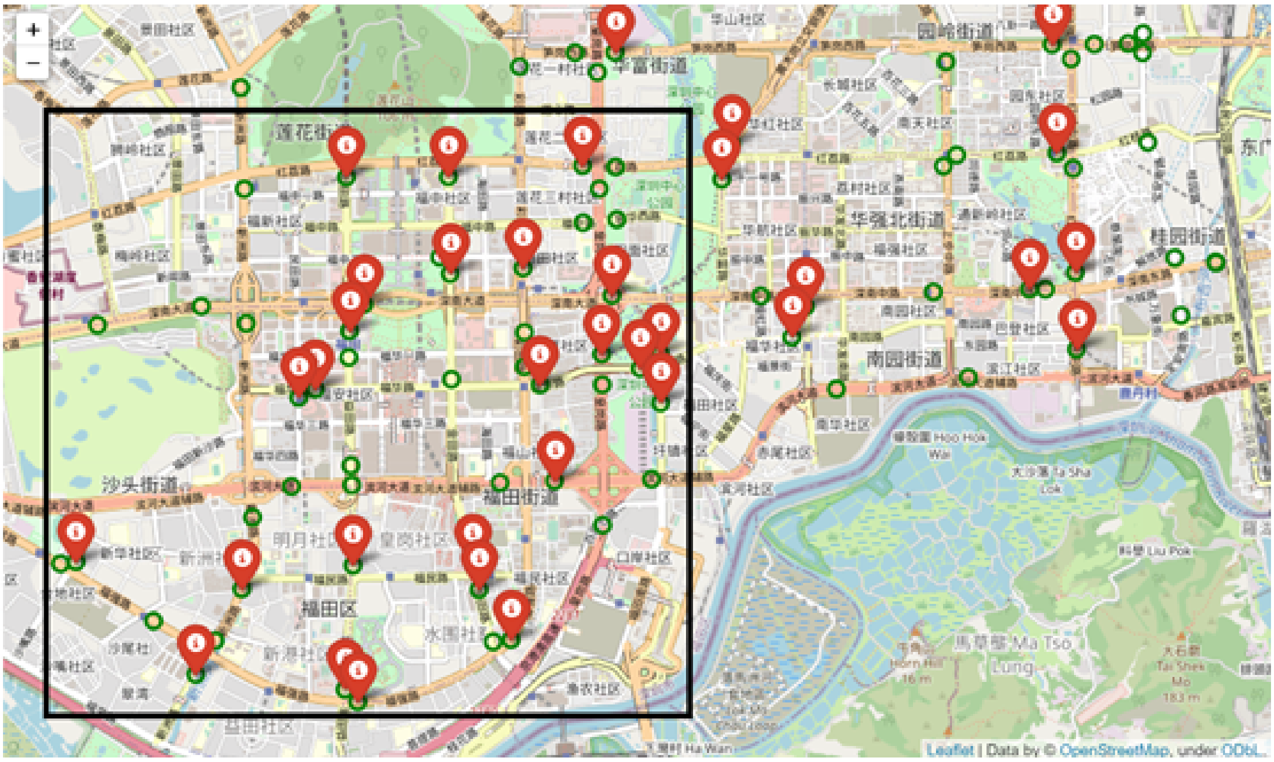

Figure 8: Actual points and missing points (green are the missing points)

4.3 Process Data in Groups by Flow Direction

After the network construction is completed, the traffic data of each node are grouped according to time and traffic flow for regression prediction to generate both inflow and outflow data. Among the incoming or outgoing data, the traffic of some data is not recorded in all time periods or is not recorded after a certain time period. Considering one-way street and traffic restriction factors after a fixed time period, these NAN values are set to 0.

	2019-05-21 00:00:00	2019-05-21 00:10:00	2019-05-21 00:20:00	2019-05-21 00:30:00	2019-05-21 00:40:00	2019-05-21 00:50:00	2019-05-21 01:00:00	2019-05-21 01:10:00	2019-05-21 01:20:00	2019-05-21 01:30:00	...	2019-05-21 22:20:00	2019-05-21 22:30:00	2019-05-21 22:40:00	2019-05-21 22:50:00	2019-05-21 23:00:00
19980237	248	210	193	170	180	170	206	137	35	0	...	486	420	502	380	398
19980260	93	111	149	73	121	131	68	110	76	52	...	253	311	199	208	250
19980198	142	125	144	149	129	126	100	89	105	91	...	284	187	253	223	209
19980262	217	263	185	200	187	178	151	176	152	161	...	443	425	394	391	364
19980195	222	301	165	0	0	0	0	0	0	0	...	879	837	703	514	684
19980235	104	106	84	80	99	100	83	60	87	75	...	154	163	141	163	143
19980126	46	53	28	36	36	25	23	32	20	30	...	109	84	93	94	80
19980206	204	157	155	148	146	145	122	103	116	114	...	502	451	413	388	343
19980197	115	144	136	109	93	114	118	96	102	86	...	203	194	231	217	205
19980204	543	573	499	480	430	438	367	389	352	327	...	1076	907	962	850	882
19980120	65	68	58	58	91	68	63	56	74	57	...	151	144	132	177	158
19980118	170	158	159	132	159	135	100	91	90	77	...	260	277	236	238	213
19980123	78	64	87	72	67	56	56	75	69	63	...	124	144	99	114	117
19980115	153	149	97	119	141	99	87	108	106	58	...	418	438	343	307	301
19980226	215	190	186	178	180	171	143	123	166	143	...	422	433	404	358	291
19980199	73	72	58	61	68	54	75	63	54	41	...	128	102	117	141	122
19980192	94	129	99	50	50	62	58	58	34	55	...	812	834	603	406	242
19980117	176	80	75	78	96	104	91	148	108	106	...	126	115	137	169	127
19980124	54	37	48	54	53	44	41	46	49	33	...	49	49	87	52	61
19980246	63	73	90	66	58	84	69	80	43	47	...	108	101	98	93	91
19980253	73	50	54	43	106	55	41	48	50	38	...	127	120	108	110	105
19980125	35	41	44	39	27	27	32	25	28	28	...	97	62	83	87	84
19980252	71	86	84	69	64	84	61	66	50	55	...	0	0	0	0	0
19980232	68	50	50	49	48	53	58	60	52	44	...	125	104	98	98	91
19980112	82	83	85	73	72	104	79	79	83	56	...	184	162	148	158	121
19980116	41	45	30	32	36	38	36	43	26	32	...	426	290	198	259	414
19980121	11	10	8	5	9	16	14	6	19	24	...	51	42	44	44	37

Figure 9: Inflow of each road in Futian regional road network

4.4 Basic Prediction

Basic prediction of flow data is made through linear regression and random forest, and the results are shown in the table below:

Model	MAE	MAPE	RMSE
Random Forest	37.45	0.24	58.82
Linear Regression	46.83	0.26	74.98

At present, only two basic models are used and only the inflow and outflow data are segmented for training, without more effective use of spatial data (adjacent matrix). Besides, the LSTM, GRU, and other neural network models will be added later to complete the baseline.

References

[1] M. Li, P. Tong, M. Li, Z. Jin, J. Huang, and X.-S. Hua, "Traffic flow prediction with vehicle trajectories.," in *National Conference on Artificial Intelligence*, 2021.

[2] J. Wang, J. Jiang, W. Jiang, C. Li, and W. X. Zhao, "Libcity: An open library for traffic prediction," in *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '21, (New York, NY, USA), pp. 145–148, Association for Computing Machinery, 2021.