東京大学大学院新領域創成科学研究科
社会文化環境学専攻

2021 年度
修　士　論　文

Predicting Human Mobility under Disaster with Twitter Data
ツイッターデータによる災害時の人間移動性の予測

2022 年 1 月 17 日提出
指導教員　柴崎　亮介　教授
副指導教員　瀬崎　薫　教授

汪　昊宇
Wang, Haoyu

Abstract

Human mobility flow in range of prefectures can provide sufficient information for urban planning and transportation management, which can help improving traffic efficiency and managing public transportation. However, human mobility flow under disaster situations is difficult to predict since people will have different behavior. By giving a historical human mobility flow information and Twitter data and other conditions, this study aims to train the neural network with this historical information and predict the current human mobility flow based on the given conditions. This issue is very important in the construction of cities, and especially in response to disasters. it is also very challenging because the human mobility flow during a disaster is very unpredictable. To achieve this task, I used cGAN neural network. It is able to estimate human mobility based on specific conditions in a continuous time series. Spatial auto-correlation is handled by using dynamic convolution, and temporal auto-correlation is handled by using self-attention mechanism. Finally, training and testing on real-world COVID-19 datasets show that using Twitter data can increase the model's accuracy in predicting people flow when disasters occur.

Key words: Human mobility flow, Twitter data, disaster, COVID-19, Conditional Generative Adversarial Nets

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Ryousuke Shibasaki, who supports us to do my research in human mobility. In every group meeting, he listened carefully to my report and he gave me lots of advice every time, which made me realize where I am lacking and how to improve myself. I am also grateful to my co-advisor Prof. Kaoru Sezaki, who also gave me practical advice on the research and thesis and it help me solve many problems.

Then, also thanks for my associate Professor Xuan Song, who gave me lots of advice on my research directions and often talk with me about my future life. Thanks to Researcher Renhe Jiang, who has been my tutor for two years and he taught me so much from research direction to programing skills. I often have meetings with him, he taught me how to do research on human mobility data and teach me about programming skills. He also cared about my daily life. I also thanks to research Zepei Fan, who also gave me lots of useful proposals and research directions.

In addition, I want to thank all my friends in Shibasaki Lab: Thanks to

Liqiang Xu, Hang Yin, Lifeng Lin, Zhiwen Zhang, Yinhao Liu, Chuang Yang, Zekun Cai, Wenjin Li, Yanxiu Jin, whom make me enjoy life and study in the University of Tokyo. Thanks to senior student Jinyu Chen, Yuhao Yao, Zhaonan Wang, Dou Huang, Tianqi Xia who also help me in daily life and studies.

Thanks to my parents abroad who gave birth to me and always care about me when I'm abroad. I am also grateful to all my other relatives for their concern.

Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Background

The rapid urbanization in recent years, due to the growth of the urban population, has a huge impact on urban traffic, which may increase travel demand, and the difficulty of traffic management and crowd forecasting and the risk of deteriorating traffic conditions when disasters such as typhoons occur. Therefore, urban traffic forecasting plays an important role in the process of urban development. It can provide insights for urban planning, traffic management and resource allocation, and help improve urban traffic efficiency and human settlements [1]. However, the occurrence of natural disasters such as typhoons has significantly reduced the accuracy of traffic prediction, because when faced with such a different situation, people's movements will be very different from those in the past.

Until now, many researches work in these kinds of urban flow prediction by using deep learning method to capture spatio-temporal characteristics. [2] tried to predict the traffic flow of the roads. [3] and [4] used convolution RNNs to predict the accidents and population density. What's more, [5] and [6] aimed to use CNN and RNN together to predict the speed and mobility flows. This research can capture the spatio-temporal

characteristics very well, but they did not use conditions, such as disaster-related condition to train models. In this case they cannot be used in such kind of disaster-related situation. Recently, [7] built up a GAN model to deal with the flow prediction task, but this GAN model did not consider the temporal characteristic and cannot be used in continuous time series prediction.

Actually, when a disaster or event happens, people will tend to discuss this happening on social media. As a result, the other users will be affected by these discussions and change their activities. Such kind of interaction between human mobility flow and social media can provide us with more information about the motivation and make the mobility flow predictions more accurate. Many previous studies have shown that social media contains important information about disasters and events. [8] and [9] both show that social data can help us to obtain useful information to estimate or forecast the disaster. What's more, we can get social media information in real-time level, while the other information, such as city-level's GPS data, cannot be obtained easily. In conclusion, social media cannot be a very useful data for us to predict human mobility, especially under disaster.

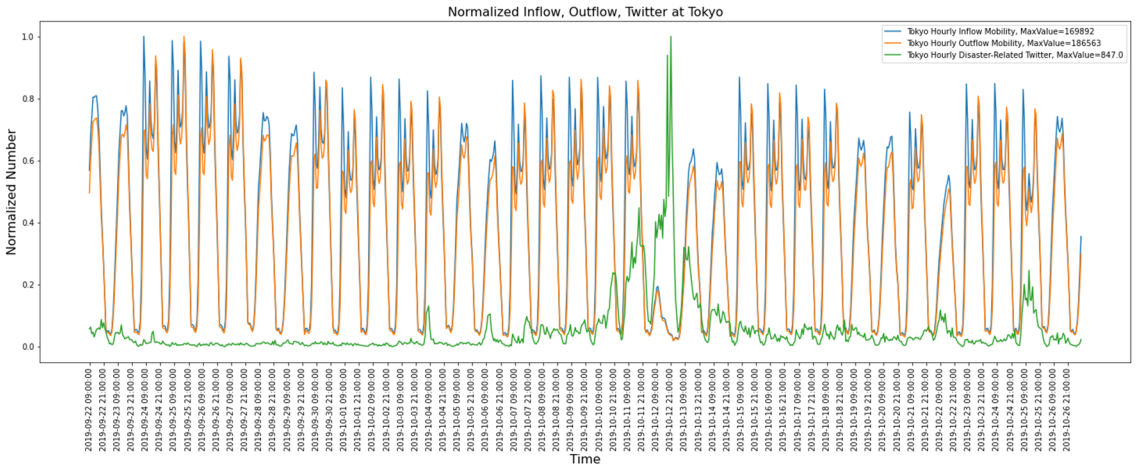

Figure 1: Tokyo's mobility flow in the period of Typhoon Hagibis

Motivated by the above, I aim to improve the accuracy of human mobility flow prediction when natural disasters occur by collecting the number of disaster-related tweets in Twitter. As shown in Figure 1, it is the hourly flow of people and the number of disaster tweets in Tokyo during the occurrence of Typhoon Hagibis from September 22nd, 2019 to October 27th, 2019. It can be seen that before the disaster occurs, Tweets will be significantly increased.

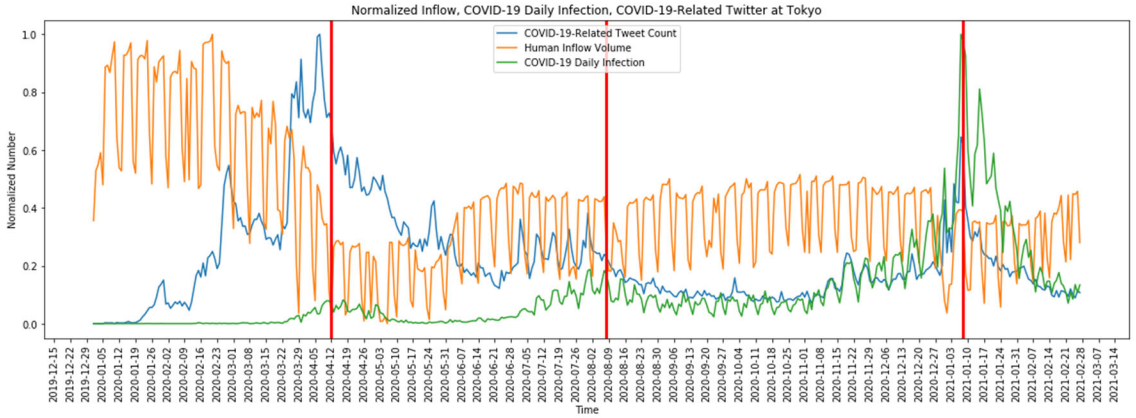

Figure 2: The comparison of Tokyo's inflow, tweets and inflection

Moreover, it is not just the situation during the typhoon. The mobility of people under COVID-19, which has affected the world in the past two years, has also been largely affected by Twitter data. As shown in Figure 2, this figure shows daily flow of people, daily increase in the number of infections and the number of COVID-19 tweets in Tokyo during the occurrence of COVID-19 from January 1st, 2020 to February 27th, 2021. We can see that the green line is the daily increase in the number of infections. During this period, there were approximately three peaks of infection which shown in red lines. These are April 12th, 2020, August 8th, and January 8th, 2021. While the increase in the number of infections causes a decline in human mobility, it is often accompanied by an increase in discussions about COVID-19 on Twitter. Therefore, in this example of COVID-19, the number of related tweets can still be used as an important condition for predicting human mobility.

By giving a historical human mobility flow information and Twitter data and other conditions, this study aims to use disaster-related tweets as a key condition and refer to Curb-GAN's network structure to train neural networks. Using historical mobility flow data to generate future mobility flow graphs to improve human mobility prediction accuracy when disasters

such as typhoons occur.

1.2 Outline

The research is structured as follows:

Chapter 1 introduces background and the outline about this study.

Chapter 2 introduces recent year's researches which to this study. In the last section, it introduces the methods which my study uses.

Chapter 3 describes problem of human mobility flow prediction under twitter data and defines some concepts.

Chapter 4 introduces the Conditional Generative Adversarial Nets, model structures and how to deal with spatial and temporal auto-correlations. The training process and loss function used in the study are also described.

Chapter 5 is about the experiment. It introduces the datasets, some pretreatment on the datasets and the adjacency matrix used in the capture spatial auto-correlations. At last, it introduces how to train the model. The measure criterion is also introduced. Different comparative experiments

and analysis are also made for evaluation.

Chapter 6 concludes this research and analyze how to improve the results in the future.

Chapter 2. Related Works

This chapter shows the previous work related to this study. First part is the study which about urban traffic flow prediction. Second part is the study which about deep learning methods used in such kinds of spatio-temporal flow prediction task. At last, it introduces the method this study uses.

2.1 Urban Flow Prediction

Previous researches which are related with such kinds of human mobility traffic flow prediction focus on many different ways. Some of the previous studies predict a person's actions based on the locations they have been to, such as [10] and [11]. However, these studies are used to forecast the trajectories of many individuals, which cannot be used to predict the mobility flow between regions. What's more, some studies try to predict the flow speed and flow volume between regions. For example, [12] built a combined framework, which combined the most advanced deep learning technology and perfect traffic flow theory to estimate the traffic flow volume of the whole city. In [13] and [14], the author developed a model to predict the flow of roads and the flow of passengers in metro stations. These researches are made based on the assumption that conditions of the flow in urban do not change and the mobility flow can be predicted only

with time and locations. When it comes to more traditional methods, agent-based simulation models [15] or physical models [16] were used to estimate traffic flow volume and mobility after constructions. However, without a good model choice and initial parameter settings, these kinds of models cannot have reasonable results across prefecture regions since they are difficult to transfer. In my research, I study the spatio-temporal auto-correlations of human flow and predict it base on different conditions and tweets number. The experiment results show that my model cannot only be used in normal situation, but also be used in disaster situation.

2.2 Deep Learning Methods in Flow Prediction

Deep learning methods are also used in many kinds of spatio-temporal flow prediction task. For example, CNN methods are often used in grid model prediction task such as citywide flow prediction [17] and taxi demand prediction [18], because CNN can capture the spatial auto-correlations of the data and have relatively good estimation results after training. What's more, RNN [19] are also widely used in such kinds of spatio-temporal prediction tasks because they are succeed in sequence prediction task. For example, [20] and [21] used RNN model to deal with video prediction and travel time prediction tasks. In other situations, [4] used ConvLSTM to

predict population density and try to capture temporal and spatial dependencies at the same time and [6] used the combination of CNN and LSTM to predict the traffic speed on the road. [3] is built to use a different kind of the ConvLSTM model to predict traffic accidents. [22] used a deep attention model to predict crimes. [23] proposed a reinforcement learning method to reposition shared bikes dynamically. However, the studies mentioned above can only be used to capture the spatial and temporal auto-correlations of mobility flow, they did not consider the influence of different conditions, such as disasters happen. In my research, I try to predict human mobility flow under different disaster situations with the spatial and temporal auto-correlations.

2.3 Mobility Flow Prediction under Disaster

Actually, there are previous studies which predicting the human mobility under disaster situation. Many previous works are made under normal situation, so they perform well. However, when disasters coming, human mobility will be very different from their daily patterns and the standard models cannot perform well. To deal with these kinds of cases, Some studies use deep learning methods [24]-[26], [27] use more traditional methods to handle the cases that under abnormal situation. What's more,

[28] and [29] try to use other technologies to solve the problem. More recently, [30] is made to predict long term mobility flow under abnormal situation with deep learning methods. [31] also studied how to predict abnormal mobility flow based on the historical mobility flow. However, all of these methods can only predict the disasters happening but cannot generate the mobility flows. To solve this problem, [32] try to get the information from disasters and use them to simulate the human mobility flow. What's more, [33] use the convolutional LSTM to predict the short-term mobility density under events or disasters. [34] use the recent COVID-19-related data are used to train a cGAN network to predict the mobility flows under disaster.

2.4 Twitter-based Human Mobility Flow Prediction

In my study, I use disaster-related tweets as an important condition to predict the human mobility, because we can get the public awareness and attention about the disaster and how they influence the human mobility to make their activity different from usual ones.

Besides Twitter condition, the normal GPS condition is also used. With normal condition, we can generate the human mobility under the situations

without disaster, no matter it is weekend, weekday, morning or evening. Combining Twitter with the normal condition, we can make the model be more sensitive with the emergency events and forecast the disaster before it happens.

Towards these ideas, a condition based Generative Adversarial Network (cGAN) is used in my study. With the combination of normal and Twitter conditions, it can learn the information from past human mobility flow data and predict the future human mobility flow under a disaster. After a series of experiments made over 7 prefectures in Japan under the COVID-19 dataset in2020. Compared with other methods, my method can achieve a best RMSE in the dataset.

Chapter 3. Preliminary

In this chapter, I first introduce the notations that used in my research. Then I introduced the problem definition of my research.

3.1 Notations

There are the definitions that will be used in the study.

Definition 1 (Regions). The whole Japan area is split into 47 prefectures, denoted as $S = \{s_i\}$, where $1 \leq i \leq 47$.

Definition 2 (Target region). The target region R is the prefectures which currently have corresponding disaster-related tweets. Currently 7 prefectures' disaster-related tweets are collected, so $R = \{s_i\}$, where s_i currently is 8,9,10,11,12,13,14.

Definition 3 (Disaster-related Tweets). The number of disaster-related tweets of a prefecture captures how people react with the disaster in a period of time. Thus, I denote number of disaster-related tweets of a region s in time slot t as $d_t^s \in \mathbb{N}$. Moreover, the number of disaster- related tweets of a target region R within a day is denoted as a sequence $D^R = \{d_1^R, \dots, d_{N_t}^R\} \in \mathbb{N}^{N_t}$, where d_t^R is the sum of number of disaster-related tweets in all prefectures within R in time slot t.

Definition 4 (Flow status and flow distribution). Flow status indicates the quality of flow, which can be measured by traffic speed, traffic inflow/outflow, traffic volume, etc. I denote m_t^s as the average flow status of prefecture s in time slot t. The flow distributions of a target region R within a day is denoted as a tensor $M^R = \{M_1^R, \dots, M_{N_t}^R\} \in \mathbb{R}^{N_t \times l}$, where we denote the l is the number of all prefectures in the region R. each entry of M_t^R is m_t^s, where $s \in R$.

Definition 5 (Correlation matrix). The correlation matrices of a target region R within a day is denoted as a tensor $A^R = \{A_1^R, \dots, A_{N_t}^R\} \in \mathbb{R}^{N_t \times l \times l}$, where A_t^R is a mobility flow correlation matrix of size $l \times l$ in region R in time slot t, A_t^R is non-negative and row-normalized.

The mobility flow correlations capture the inbuilt traffic dependencies between prefectures. I use adjacency matrix of each pair of prefectures to quantify their corresponding spatial correlations. Note here I assume that flow status at different locations is either positively correlated or independent.

3.2 Problem Definition

When applying this problem to my dataset, the input can be divided into

four parts: input ground-truth flow data X^R, noise Z, input condition C^R and adjacent matrix A^R. Among them $X^R = \{X_1^R, \dots, X_B^R\} \in \mathbb{R}^{B\times l\times p}$, $Z = \{Z_1, \dots, Z_B\} \in \mathbb{R}^{B\times l\times p\times p}$, $C^R = \{C_1^R, \dots, C_B^R\} \in \mathbb{R}^{B\times l\times p\times 38}$ and $A^R = \{A_1^R, \dots, A_B^R\} \in \mathbb{R}^{B\times l\times p\times p}$, where B is the input batch size, l is the input length of the time series, p is the number of input prefectures, the number 38 in C shows the total number of input condition.

The reason why I combine normal GPS conditions and twitter condition is that the GPS condition can be used to predict the flow of people in general situations (that is, when the disaster does not occur), and the twitter condition can be used to predict the flow of people that is different from the normal one when the disaster occurs. Combining the two together can effectively improve the prediction accuracy of the model.

The output is Y^R, and $Y^R = \{Y_1^R, \dots, Y_B^R\} \in \mathbb{R}^{B\times l}$. Y^R is the result distinguished by the trained discriminator, shows whether the time series generated by the trained generated is true or not. Then the results will feedback to the generator to make it generate a more “real” time series.

Therefore, the problem can be described as that, the selected Japan

prefectures are partitioned into regions R, given $B \times \| R \|$ samples of C^R, for one specific target region R, estimate the traffic distributions $\widehat{M}^R = \{\widehat{M}_1^R, \dots, \widehat{M}_B^R\}$ in continual time slots based on a given number of disaster-related tweets sequence $\hat{C}^R = \{\hat{C}_1^R, \dots, \hat{C}_B^R\}$. The overview of the problem can be seen as Figure 3.

After training, the model expects that the test results will maintain high accuracy in the absence of disasters, and at the same time, the test results should have relatively good accuracy when disasters occur. Therefore, it is expected to have an average RMSE below 0.13 on our COVID-19 dataset. The speed of the testing is expected to be able to achieve real-time generation.

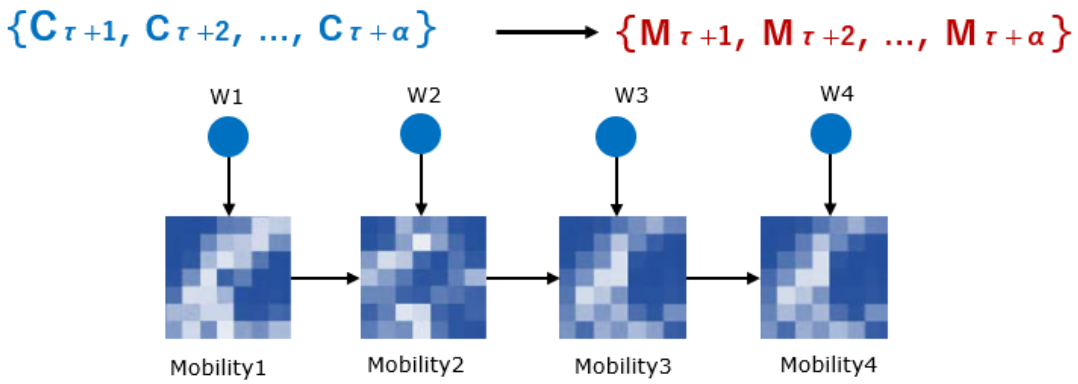

Figure 3: Overview of the problem

Chapter 4. Methodology

Nowadays, deep neural network (DNN) is a very good way to solve the problem of human mobility prediction. Among these, the use of GAN to solve is a great innovation. Since I want to solve the problem under certain condition, the conditional GAN is good choice. Human mobility flow generation task can be seen as image sequences task, the disaster-related tweets can be treated as conditions, the mobility flow between prefectures in one timestamp can be seen as an 'image' and the flow of each prefecture can be seen as a pixel of the image. In this way, the conditional GAN (cGAN) structure can be used in my study. However, directly using cGAN in my research cannot solve the problem with good results since simple cGAN model cannot capture the spatio-temporal auto-correlations very well. Therefore, in order to solve these problems. I try to train the model based on the Curb-GAN structures, which uses dynamic convolution [35] and self-attention mechanisms [36] to solve the spatio-temporal auto-correlations under certain disaster-related tweets. Finally, after using the modified model, it can generate the future human mobility flows in continual timestamp.

In this chapter, I first introduce the dynamic convolution and self-attention

mechanisms used in the study. Then I introduce the structure of the model with the generator G and the discriminator D.

4.1 Conditional Generative Adversarial Network (cGAN)

GAN (Generative adversarial network) [37] is a network that is very widely used in image generation and other fields. GAN mainly includes two networks: one is a generator and a discriminator. The purpose of the generator is to map random input Gaussian noise into a fake image, and the discriminator is to determine whether the input image comes from the generator. That is, the probability of judging whether the input image is a fake image.

The purpose of GAN is to make the so-called "fake picture" generated by the generator deceive the discriminator, then the optimal state is that the so-called "fake picture" generated by the generator has a discrimination result of 0.5 in the discriminator, which cannot distinguish it is a real picture or a fake picture.

However, the original GAN generator can only generate images based on random noise. As for what the image is (that is, we don't know what the

label is), the discriminator can only receive image input to determine whether the image is made by the generator. Therefore, the main contribution of the conditional GAN (cGAN) [38] is to add additional information to the input of the original GAN generator and discriminator. The additional information can be any information, such as a label. Therefore, cGAN allows GAN to use images and corresponding labels for training, and use the given labels to generate specific images in the testing part.

Figure 4: The network architecture diagram of cGAN

The cGAN uses the MLP (Fully Connected Network) in the network architecture. In the generator in cGAN, we give an input noise $p_z(z)$ and additional information, and then connect the two together through a fully

connected layer as the hidden layer input. Similarly, the input image and additional information in the discriminator will also be connected together as the hidden layer input. The network architecture diagram of cGAN is as shown in Figure 4. The objective function of cGAN can be expressed as follow.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

4.2 Graph Convolution Network (GCN)

In human mobility prediction task, the strength of spatial auto-correlations is often heterogeneous, which mostly relies on the locations and the complex structures. Based on the First Law of Geography [39], nearby locations often have stronger traffic spatial auto-correlations. Hence, graph convolution networks are used in both G and D, which can better capture the diverse footprints of spatial auto-correlations.

The goal of dynamic convolution is to seek a balance between network performance and computation cost. Conventional methods of improving network performance (wider and deeper) often lead to higher computational consumption, and therefore are not friendly to efficient networks. The graph convolution will not increase the depth and width of

the network. On the contrary, the fusion of multiple convolution kernels will improve the expression ability of the model. The obtained convolution kernel is related to the input, that is, different data have different convolutions, which is the origin of graph convolution.

The traditional perceptron is defined as $y = g(\widetilde{W}^T x + \tilde{b})$, where W, b, g respectively represents the weight, bias, and activation function; then, the author defines the dynamic perceptron as follows:

$$y = g(\widetilde{W}^T x + \tilde{b})$$

$$\widetilde{W} = \sum_{k=1}^{K} \pi_k(x) \widetilde{W}_k$$

$$\tilde{b} = \sum_{k=1}^{K} \pi_k(x) \tilde{b}_k$$

$$s.t. 0 \leq \pi_k(x) \leq 1, \sum_{k=1}^{K} \pi_k(x) = 1$$

Among them π_k represents the attention weight. The attention weight is not fixed, but changes with the input. Therefore, compared with static convolution, graph convolution has stronger feature expression ability.

4.3 Self-Attention Mechanism (SA)

The graph convolution layer is applied to capture the spatial autocorrelation of urban traffic. In order to capture the time dependence. Self-attention mechanism has achieved very good performance in dealing with language modeling and machine translation problems mainly in seq2seq model. Self-attention mechanism processes sequential data including text, audio and video, and learns time dependence. Compared with LSTM [40] and GRU [41], the self-attention mechanism is parallel computing, so the training time is less and the training quality is higher.

First, self-attention will calculate 3 new vectors. The 3 vectors are called Query, Key, and Value. These 3 vectors are the result of multiplying the embedding vector and a matrix. The second dimension needs to be combined with embedding. The dimension's value will always be updated during the BP process, and the dimensions of the 3 vectors obtained are lower than the embedding dimension. The calculation process of the self-attention mechanism can be shown as below:

Step.1 Convert the input word into an embedding vector.

Step.2 Obtain the three vectors q, k, v according to the embedding vector.

Step.3 Calculate a score for each vector: score = q. k.

Step.4 In order to stabilize the gradient, Transformer uses score for

normalization, which is divided by $\sqrt{dk}$.

Step.5 Apply softmax activation function to the score;

Step.6 The value v is multiplied by the softmax function to obtain the weighted score v of each input vector;

Step.7 After the addition, the final output result z is $z = \sum v$.

The calculation formula is as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

where d_k is the dimension of K.

4.4 Model Structure

In order to provide the estimation of daily human mobility based on date and twitter data, I use Curb-GAN [42] structure to control the number of human mobility under different conditions. Figure 5 shows the overall structure of Curb-GAN. The Curb-GAN contains a generator G and a discriminator D. The purpose of generator G is to generate a human mobility distribution sequence similar to the real sequence in continuous time slots, so that discriminator D cannot well distinguish the generated sequence from the real sequence.

Figure 5: The structure of Curb-GAN

4.4.1 Generator

The generator G are used to generate human mobility distributions with respect to the daily tweets count sequence D^R in a specific region. The input of the generator G includes three parts,

1.A noise tensor $Z = \{z_1, \dots, z_{N_t}\} \in \mathbb{R}^{N_t \times N_s \times N_s}$, randomly sampled from Gaussian distribution.

2.A condition tensor $C^R = \{C_1^R, \dots, C_{N_t}^R\} \in \mathbb{R}^{N_t \times N_s \times c}$, where C_t^R is a matrix of size $N_s \times c$ defining the current date, time and the level of twitter data.

3. A correlation matrix tensor A^R. In generator, C^R is concatenated into Z and it builds the mapping from distribution $p_Z(Z)$ to a human mobility $G(C^R, A^R, Z)$.

4.4.2 Discriminator

The discriminator D tries to rise the output score if the input is real human mobility sequence, and lower down the score if the input is generated human mobility sequence. D takes three inputs,

1. 12hours human mobility tensor M^R.
2. A condition tensor C^R.
3. A correlation matrix tensor A^R. D outputs a number indicating whether the input human mobility tensor M^R is real and whether the input M^R and C^R are matched.

4.5 Activation Function

Activation functions are very important for artificial neural network models to learn and understand complex and non-linear functions. They introduce non-linear characteristics into network. In the neuron, the input inputs are weighted and summed, and then a function is applied. This function is the activation function. The activation function is introduced to increase the nonlinearity of the neural network model. Each layer without activation function is equivalent to matrix multiplication.

Sigmoid function

A sigmoid function is a bounded, differentiable, real function that is defined for all real input values and has a non-negative derivative at each point and exactly one inflection point. A sigmoid "function" and a sigmoid "curve" refer to the same object.

$$S(x) = \frac{1}{1 + e^{-x}}$$

Figure 6: The shape of sigmoid function

Tanh function

It is very similar to the sigmoid activation function. The function takes any real value as input and outputs values in the range -1 to 1. The larger the input (more positive), the closer the output value will be to 1.0, whereas the smaller the input (more negative), the closer the output will be to -1.0.

$$tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Figure 7: The shape of Tanh function

ReLU function

It is an activation function commonly used in artificial neural networks, usually referring to a non-linear function represented by a ramp function and its variants.

$$ReLU(x) = \max(0, x)$$

Figure 8: The shape of ReLU function

Softmax function

The Softmax function is an extension of the logic function. It can compress a K-dimensional vector x containing any real number into another K-dimensional real vector $\sigma(x)$ so that the range of each element is between (0,1), and the sum of all elements is 1.

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \; for \; j = 1, \ldots, K$$

4.6 Loss Function

In order to obtain good training results, the loss functions of the generator and discriminator use the following two functions respectively

$$\tilde{V}_D = \frac{1}{m}\sum_{i=1}^{m}(\log\left(1 - D\left(C^i, A^i, \widetilde{M}^i\right)\right) + \log D\left(C^i, A^i, M^i\right)$$

$$+ \log\left(1 - D\left(C^i, A^i, \widehat{M}^i\right)\right))$$

$$\tilde{V}_G = \frac{1}{m}\sum_{i=1}^{m}\log D(G(C^i, A^i, Z^i))$$

Among them, the loss function of the discriminator is $\tilde{V}_D$, the goal of the discriminator is to correctly identify all human mobility tensors matched with the corresponding conditions, and all generated fake tensors and mismatched human mobility tensors.

The loss function of the generator is $\tilde{V}_G$, the goal of the generator is to generate the corresponding human mobility tensors according to the given conditions, which is real enough for the discriminator to judge that these are real tensors.

After training, I use the well-trained generator to generate the future human mobility flow in continued timestamps between prefectures with certain disaster-related tweets.

4.7 Training Process

In the training process, I use backpropagation through time. The whole training process can be divided into the following processes.

Input: Training iteration k, a training set P, initialized G and D.

Output: Well trained G and D.

1. In every training iteration $iter$:
2. **Repeat**
3. Sample P_0 from training set P.
4. Sample B from Gaussian distribution.
5. Generate $\tilde{O}$ with G.
6. Sample $\hat{O}$ from training set Z.
7. Update D.
8. Update G.
9. **Until** $iter > k$.

During the training process, the discriminator D and the generator G are respectively updated in line 3 – 7 and line 8. We let the training set which contains n samples as $P = \{(C^1, A^1, M^1), \dots, (C^n, A^n, M^n)\}$ let $P_0 = \{(C^1, A^1, M^1), \dots, (C^m, A^m, M^m)\}$ as a subset of P which has m samples. Let $B = \{Z^1, Z^2, \dots, Z^m\}$ as a set of m noise tensors sampled from Gaussian distribution, $\tilde{O} = \{\widetilde{M}^1, \dots, \widetilde{M}^m\}$ as a set of m human mobility tensors generated with G, where $\widetilde{M}^i = G(C^i, A^i, Z^i)$. Let $\hat{O} = \{\widehat{M}^1, \dots, \widehat{M}^m\}$ as a set of m human mobility tensors sampled from the training set P, each $\widehat{M}^i$ is mismatched with (C^i, A^i).

Chapter 5. Experiment

In this chapter, I first describe the real-world dataset and the conditions I used and then introduce baselines and the evaluation metrics. At last, I make experiments on the dataset.

5.1 Dataset

In this research, I mainly use two datasets. Japan prefecture-level mobility flow data and COVID-19-related tweets.

5.1.1 Data Description

• Japan prefecture-level mobility flow data.

The inflow/outflow and ODflow datasets between prefectures are collected from Jan 1st, 2018 to Feb 28th, 2021. The datasets are hourly collected, which means over 25000 time slots are included. It contains information on human mobility in 47 prefectures in Japan. In this experiment, 7 prefectures in the Kanto region are selected as the targets of the experiment.

• COVID-19-related tweets data

We also collect the number of tweets every hour of 47 prefectures from Jan 1st, 2020 to Feb 28th, 2021. When someone mentions keywords such as the

new crown virus, infection, emergency, and self-defense in a tweet, we call it COVID-19-related tweets. We count the number of COVID-19-related tweets per hour and record the dataset by prefecture.

The detailed information of the dataset is shown in Table 1.

Dataset	**Human mobility flow**	**COVID-19-related tweets**
Timespan	01/01/2018-02/28/2021	01/01/2020-02/28/2021
Time slot	1 hour	1 hour
Total prefectures	47	47
Selected prefectures	7	7

Table 1: Detailed information of the dataset

5.1.2 Data Pretreatment

Since the tweets data contains data from 47 prefectures in Japan, the amount of data is too much for neural network processing, and many prefectures with a small population have less influence on the training results. Therefore, among the tweets of 47 prefectures, the 7 Kanto prefectures, which are Tokyo, Kanagawa, Chiba, Saitama, Ibaraki, Tochigi, and Gunma, are selected as the final network training data.

In order to obtain the inflow and outflow data of the 7 prefectures in Kanto,

it is necessary to intercept the ODflow data, count the number of people who flow in or out of the corresponding seven prefectures, and add them to obtain the inflow and outflow data of the seven prefectures in Kanto.

5.1.3 Adjacency Matrix

	Ibaraki	Tochigi	Gunma	Saitama	Chiba	Tokyo	Kanagawa
Ibaraki	1	1	0	1	1	0	0
Tochigi	1	1	1	1	0	0	0
Gunma	0	1	1	1	0	0	0
Saitama	1	1	1	1	1	1	0
Chiba	1	0	0	1	1	1	0
Tokyo	0	0	0	1	1	1	1
Kanagawa	0	0	0	0	0	1	1

Table 2: The adjacency matrix of Kanto prefectures

In order to better learn the characteristics of the spatial dimension of the training data, dynamic convolution is used in the neural network. As input, dynamic convolution needs to obtain the spatial features of the seven Kanto prefectures, so I used the adjacency matrix. As shown in Table 2, the shape of the adjacency matrix is 7x7, and the rows and columns represent Ibaraki, Tochigi, Gunma, Saitama, Chiba, Tokyo, Kanagawa respectively. If the prefectures are adjacent in the physical sense, the corresponding coordinate

of the point is set to 1, otherwise, it is set to 0. Using the adjacency matrix, the model can better learn the mobility flow information between adjacent prefectures.

5.2 Input Condition

As parts of input of cGAN, conditions are necessarily for training. I use time-level conditions to capture the feature without disaster happens. Then disaster-related tweets are concatenated to the standard conditions to generate the human mobility flow under disaster situation.

5.2.1 Standard Condition

Because of the introduction of cGAN, the goal of the model is to generate the corresponding time series under the corresponding conditions. As the standard input conditions, there are 32-dimensional one-hot data. The meaning of each dimension is shown in Table 3.

Monday	Tuesday	…	Saturday	Sunday	0:00	1:00	…	22:00	23:00	Holiday

Table 3: The content of 32-dimensional conditions

Each dimension indicates whether this timestamp is the corresponding statement, if it is, it is 1, otherwise it is 0.

5.2.2 Tweets One-hot Condition

How to add Twitter data as a condition to cGAN is also a topic, because the previous conditions are all one-hot data, and Twitter data is a continuous data, so it is necessary to convert the type of Twitter data into one-hot data to add to the condition. I tried two methods to add Twitter data to the conditions. The transformation process can be shown in Figure ?.

1. Set a threshold. If the number of tweets of the timestamp is greater than the threshold, set it to 1, and set it to 0 if it is less than the threshold.
2. Divide the value of the Twitter data into 6 intervals from small to large, and set six dimensions. If the Twitter data corresponding to the timestamp is in which interval, the dimension where the Twitter data is located is set to 1, and the other 5 dimensions is set to 0. The Twitter data after being divided into six intervals is shown in the Figure 9 below, which can be seen that this method can better preserve the information held by the Twitter data and turn it into one-hot data corresponding to the format.

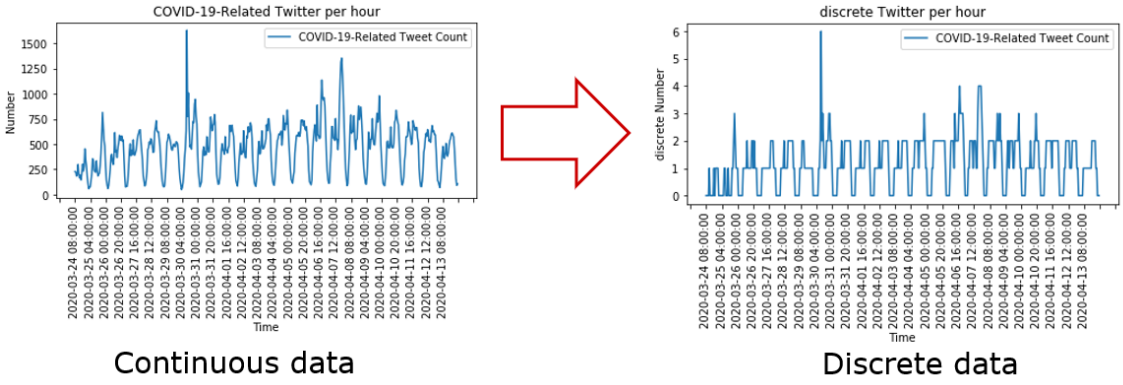

Figure 9: Transformation process of tweets

5.3 Baseline

After studying the data, it is found that the characteristics of the spatial dimension have relatively little influence on the training results. Whether the network can learn the changes in the inflow and outflow of human mobility in the same prefecture in the time dimension has a great influence on the effect of the final model.

• Long term data

In the training phase, the use of long-term data may have a great impact on the results. Because cGAN is a relatively difficult network to train, the use of long-term data will affect whether the model can effectively learn the characteristics of time series.

• Self-attention head

Since I use multi-head self-attention mechanism, where the queries, keys

and values are linearly transformed h times to get h different attentions, which are then concatenated together and go through a linear transformation to get the final values, the multi-head self-attention is calculated with

$$MultiHead(Q,K,V) = Concat(head1,\dots,head_h)W^O$$

$$\text{where } head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$$

where $W_i^Q \in \mathbb{R}^{d_{model}\times d_k}$, $W_i^K \in \mathbb{R}^{d_{model}\times d_k}$, $W_i^V \in \mathbb{R}^{d_{model}\times d_v}$ and $W^O \in \mathbb{R}^{d_{model}\times hd_v}$ are parameter matrices, d_v is the dimension of V and d_{model} is the dimension of the outputs.

Therefore, by changing the number and structure of the heads in the multi-head attention mechanism, to find out how many heads can bring us the best model effect. In general, I tried three different methods of different number of heads in the experiment. They are the single-head attention mechanism, the 7-head attention corresponding to the model dimension, and the 2-head attention mechanism as the baseline.

• Discriminator block

If the number of the discriminator blocks is too large, it may make the model too "strong" and distinguish all the generated time series as false sequences. On the contrary, if it is too weak, it may not be able to produce

enough real time series. Therefore, we need to train with different numbers to find the most appropriate value.

• Twitter data

In order to prove the improvement of the model effect after the introduction of twitter data, it is necessary to compare the training effect with and without twitter data.

5.4 Measure Criterion

•MAPE

The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD), is a measure of prediction accuracy of a forecasting method in statistics. It usually expresses the accuracy as a ratio. Mean absolute percentage error is commonly used as a loss function for regression problems and in model evaluation, because of its very intuitive interpretation in terms of relative error.

$$MAPE = \frac{1}{N_s N_t} \sum_{s=1}^{N_s} \sum_{t=1}^{N_t} \left| y_{s,t} - \hat{y}_{s,t} \right| / y_{s,t}$$

•RMSE

The root-mean-square error (RMSE) is a frequently used measure of the

differences between values (sample or population values) predicted by a model or an estimator and the values observed. The RMSR represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences.

$$RMSE = \sqrt{\frac{1}{N_s N_t}\sum_{s=1}^{N_s}\sum_{t=1}^{N_t}(y_{s,t} - \hat{y}_{s,t})^2}$$

In both RMSE and MAPE, $y_{s,t}$ is the ground-truth human mobility tensor observed in the s-th prefecture and t-th time slot, and $\hat{y}_{s,t}$ is the corresponding prediction.

5.5 Experiment Setups

My experiments are performed on a ubuntu server with 4x GeForce RTX 3090 24GB graphics card. 80% of the COVID-19-related tweets and human mobility data are used for training, and the rest are used for testing.

To compare the results of different experiments, I basically set the same hyperparameters for different experiments. The length of input timestamp is 12, the batch size is 64, the initial learning rate is set to 0.0002, and the

Adam dynamic learning rate algorithm is used. When the epoch is 10, the learning rate of the dynamic convolution layer is changed to one-tenth of the original. Since I only used data from the seven Kanto prefectures for training, the shape of the adjacency matrix is 7x7. The entire training and testing data period is from January 1st, 2020 to February 28th, 2021.

Because I used cGAN as the training method, GAN is a very difficult neural network to train, and the effect was not very good at the beginning of training, so I think the reason is that the effect of discriminator D is too strong, so that all generated time series are judged to be false. Therefore, it is adopted to reduce the number of building blocks of the discriminator D. As a comparative experiment, two schemes of D=1 and D=2 were adopted.

In terms of Twitter data, two schemes are adopted: adding one-dimensional one-hot, and dividing the Twitter data into six parts and adding six-dimensional one-hot.

5.6 Results

Experiments are carried out by changing different hyperparameters to find out when the results are the best.

5.6.1 Performances (long-term data)

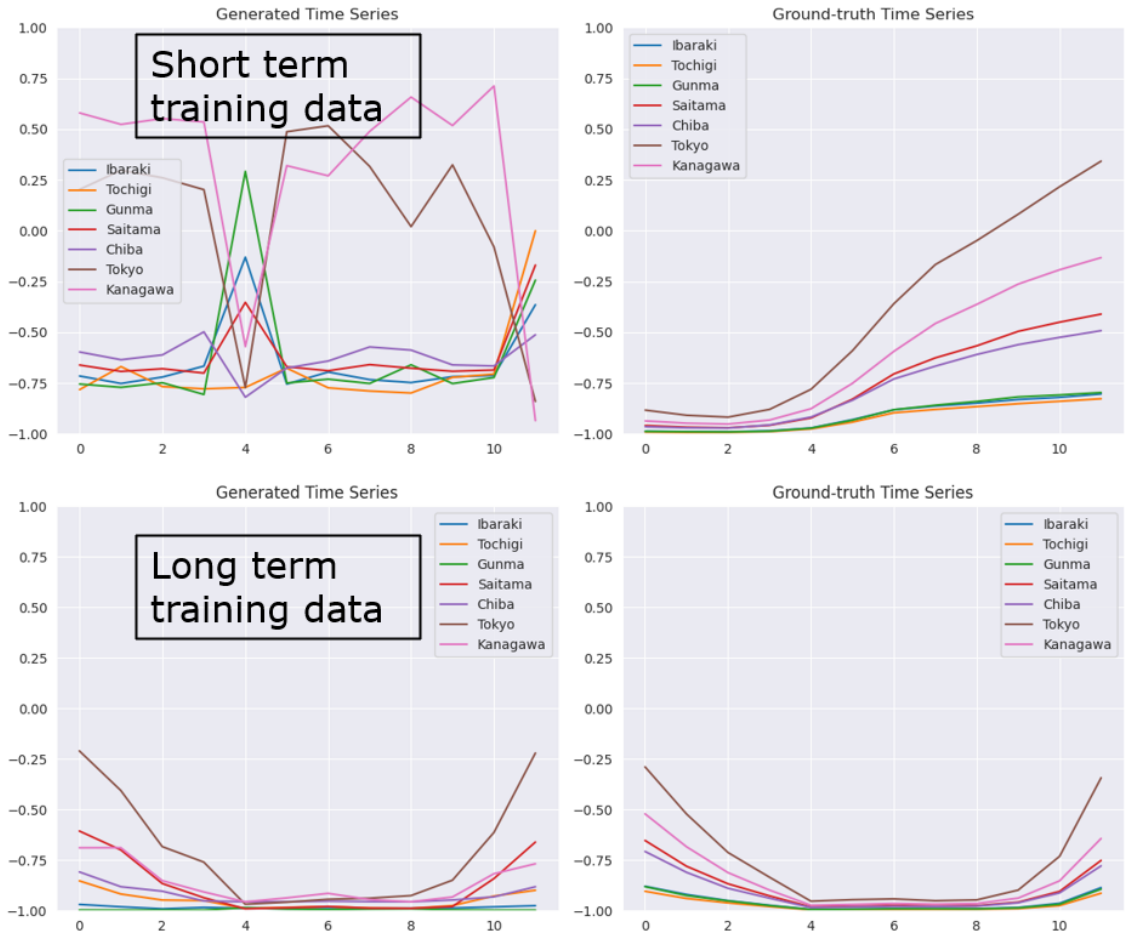

Figure 10: Results of different training data length

At the beginning, I only used the data of about 3 months to train, and the results were not ideal. After changing into one-year long-term data, the model successfully learns the characteristics of time series and can generate normal time series. Therefore, too short training data will make the model unable to learn the characteristics of the time series of the data and unable to fit. The Figure 10 shows the results of using long term data for training or not. We can see that if we only use short-term data for training, the

model cannot even fit. When using long-term data training, the model performs well in prediction, which further proves the necessity of long term training data.

5.6.2 Performances (attention heads)

I also tried the effect of the number of heads of self-attention mechanisms on the results. I tried the results of 1 head, 2 heads and 7 heads respectively. The results after training for 1000 epochs can be shown in Figure 11.

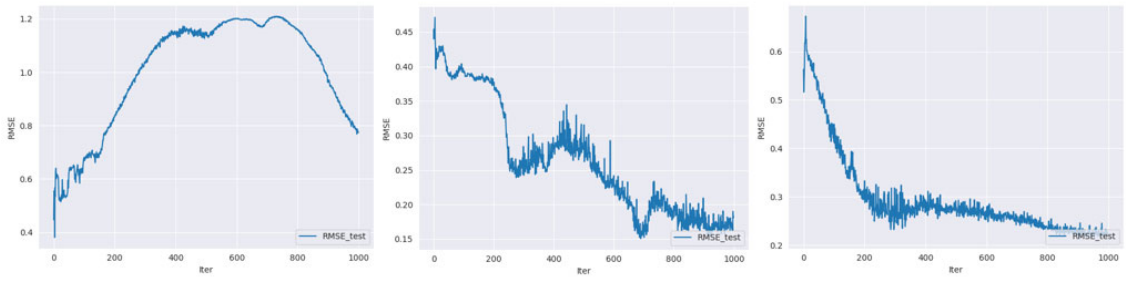

Figure 11: RMSE-test by using different numbers of heads

We can see that the model fails to converge when using one head. Comparing 2 heads with 7 heads, 7 heads can get lower RMSE. This means that when the number of heads of the self-attention mechanism is the same as the number of prefectures trained, the model can better learn the changes in the human mobility in the time dimension.

5.6.3 Performances (discriminator blocks)

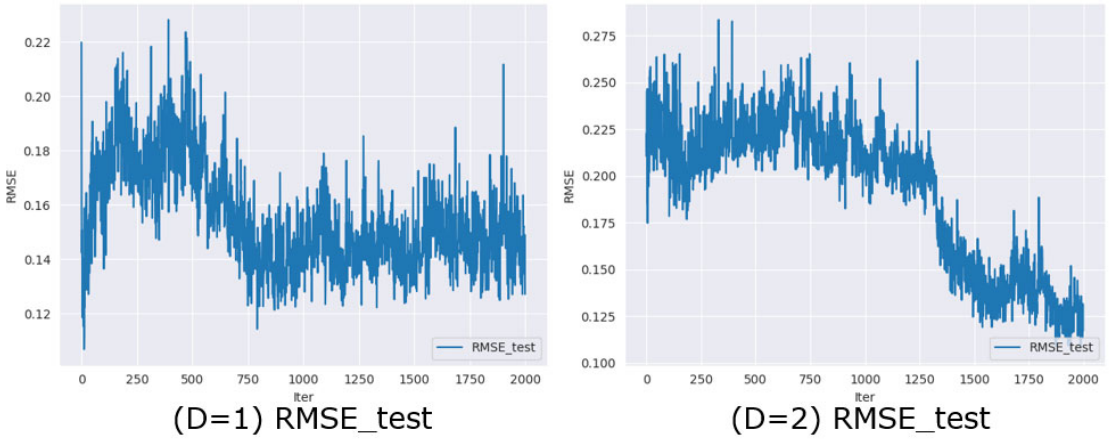

Figure 12: RMSE-test by using different numbers of D blocks

In terms of discriminator blocks, two cases where the number of building blocks of the discriminator is 1 and 2 are tried respectively. Finally, it is found that when the number of building blocks is 2, the model can obtain better RMSE. As shown in Figure 12, when D=1, the RMSE of the model remains around 0.145 and is not decreasing. When D=2, after training for 2000 epochs, the RMSE of the model dropped to 0.138 and still maintained a downward trend. This shows that a discriminator that is too strong cannot train a model, but if it is too weak it cannot generate a sufficiently realistic human mobility series.

5.6.4 Performances (using twitter data)

Because of the difficulty of GAN training, directly adding 6-dimensional Twitter data to the 32-dimensional condition cannot train the model. Adding 6-dimensional tweets directly makes the model unable to fit.

Therefore, I took the data of adding 6 dimensions with all 0 dimensions first, and after training for 2000 epochs, until the RMSE of the model to be low enough, then took the method of transfer learning and used the real Twitter data with 6 dimensions added to finetune the model. After using this kind of finetuning method, the model becomes easier to train. Such as Figure 13 shows, the left is the curves of the loss function without finetuning method, and the right is the curves of the loss function with finetuning method. It can be seen that the curve on the left is high except for D_loss_train, which means that the model has hardly learned the characteristics of the data. On the right side, after a certain epoch of training, D_loss increases, G_loss decreases, and finally reaches a balance, which is the ideal state of the loss function trained by GAN.

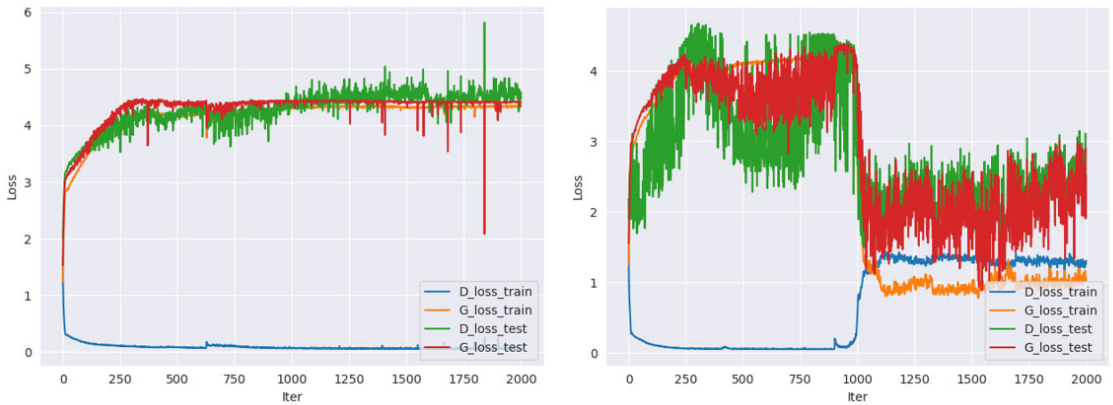

Figure 13: Loss function curves of using finetuning or not

Finally, after the introduction of Twitter data, the model also has very good prediction results for the mobility flow decline during the COVID-19

epidemic. Figure 14 shows 2021 New Year's holidays hourly inflow prediction, model successfully capture the feature that when holiday is coming, mobility flow will less than workday. What's more, as shown in Figure 15, compared with the model without Twitter data, the model with Twitter data was able to predict the decline in mobility flow due to the increase in the number of infected people during this period.

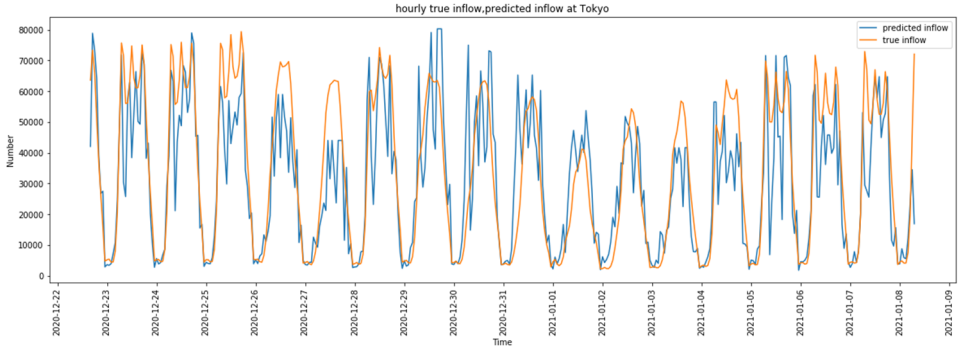

Figure 14: Generated and true inflow during New Year's holidays

Figure 15: The results of models trained with and without tweets

5.7 Analysis

Finally, the model achieves the best RMSE with 7 self-attention heads, 2 discriminator building blocks, and 6 one-hot data tweets. As shown in Table 4, it is the comparison between the best model and other models.

By adding 32-dimensional condition related to the present timestamp, the model can generate a time series that is very similar to the human flow in the real world. The flow of people increases during the morning and evening rush hours, and decreases on weekends and holidays relative to working days. These basic features of human mobility can be well fitted.

After adding Twitter data as an input condition, the feature of the decrease in human mobility on the general trend caused by the increase in the number of infected people is also fitted to a certain extent. Besides, because the addition of 6 dimensions greatly increases the training difficulty of the model, so set the 6 dimensions to be all 0 in advance, and then using the transfer learning method for finetuning can greatly reduce the difficulty of training.

In the case of using two discriminator building blocks instead of one, the model retains better generative performance. For example, if only one discriminator's building block is used, the model will not be able to generate a time series with a decrease in the number of people based on the condition of infections. After adding one discriminator, the model can increase the recognition of this condition. The training difficulty when using three blocks is also reduced.

Number of heads	Number of D blocks	Include twitter or not	Testing RMSE
2	1	no	0.2436
2	2	no	0.3501
7	1	no	0.1814
7	2	no	0.2678
7	1	yes	0.1455
7	**2**	**yes**	**0.1363**

Table 4: Performance with different model settings

Chapter 6. Conclusion and Future Work

In this research, I estimate the human mobility flow by providing conditions such as the corresponding time and the number of associated tweets. Solving this problem can be very important in early warning of disasters. And by using dynamic convolution and self-attention mechanism, the model can better obtain the auto-correlation of data in time and space dimensions. Finally, by using the data of human mobility in the real world, and training the model on this dataset, finally the model can obtain the time series that has not been observed under the given conditions.

In the study using Twitter data as an input to predict human mobility during disasters, I also found that the decline in human mobility varies in severity. That is, when a disaster occurs, the number of related tweets will rise immediately, but the degree of decline in the flow of people may vary depending on the type of disaster. If it is a disaster such as a small typhoon, people may not care at all. If it is a large typhoon or a large infectious virus like COVID-19, the flow of people will drop significantly. The impact of different types of disasters on the results is also a topic worthy of study.

However, there is still something can be improved in the current model. For

example, sometimes Twitter data does not fully correlate with human mobility. What's more, in the 2019 typhoon season, although typhoon-related tweets increased, human mobility flow did not decrease accordingly. Therefore, in order to better predict the human mobility flow based on the corresponding conditions, it may be necessary to add other inputs, such as increasing the number of keywords of tweets or adding more information. I will consider the methods and make the prediction become better in the future.

Reference

[1] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: concepts, methodologies, and applications. TIST (2014).

[2] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, Fei-Yue Wang, et al. 2015. Traffic flow prediction with big data: A deep learning approach. IEEE Transactions on Intelligent Transportation Systems (2015).

[3] Zhuoning Yuan, Xun Zhou, and Tianbao Yang. 2018. Hetero-ConvLSTM: A Deep Learning Approach to Traffic Accident Prediction on Heterogeneous Spatio-Temporal Data.

[4] Ali Zonoozi, Jung jae Kim, Xiao-Li Li, and Gao Cong. 2018. Convolutional Recurrent Model for Crowd Density Prediction with Recurring Periodic Patterns. In IJCAI.

[5] Zhiyong Cui, Ruimin Ke, and YinhaiWang. 2017. Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction. In 6th International Workshop on Urban Computing.

[6] Haiyang Yu, Zhihai Wu, Shuqin Wang, Yunpeng Wang, and Xiaolei Ma. 2017. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. Sensors (2017).

[7] Yingxue Zhang, Yanhua Li, Xun Zhou, Xiangnan Kong, and Jun Luo. 2019. TrafficGAN: Off-Deployment Traffic Estimation with Traffic Generative Adversarial Networks. In ICDM.

[8] T. Sakaki, M. Okazaki, and Y. Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In WWW.

[9] Yabe, T., Tsubouchi, K., Shimizu, T., Sekimoto, Y., & Ukkusuri, S. V. 2019. Predicting Evacuation Decisions using Representations of Individuals' Pre-Disaster Web Search Behavior. In SIGKDD.

[10] Zipei Fan, Xuan Song, Ryosuke Shibasaki, and Ryutaro Adachi. 2015. CityMomentum: An Online Approach for Crowd Behavior Prediction at a Citywide Level. In ACM UbiComp.

[11] Xuan Song, Quanshi Zhang, Yoshihide Sekimoto, and Ryosuke Shibasaki. 2014. Prediction of Human Emergency Behavior and Their Mobility Following Large-Scale Disaster. In SIGKDD.

[12] Xianyuan Zhan, Yu Zheng, Xiuwen Yi, and Satish Ukkusuri. 2017. Citywide Traffic Volume Estimation Using Trajectory Data. TKDE (2017).

[13] Xinyue Liu, Xiangnan Kong, and Yanhua Li. 2016. Collective traffic prediction with partially observed traffic history using location-base social media. In CIKM.

[14] Ermal Toto, Elke A. Rundensteiner, Yanhua Li, Richard Jordan, Mariya Ishutkina, Kajal Claypool, Jun Luo, and Fan Zhang. 2016. PULSE: A Real Time System for Crowd Flow Prediction at Metropolitan Subway Stations. In ECMLPKDD.

[15] Benhamza Karima, Salah Ellagoune, Hamid Seridi, and Herman Akdag. 2019. Agent-based modeling for traffic simulation. (June 2019).

[16] John Taplin. 2008. Simulation models of traffic flow. (2008).

[17] Junbo Zhang, Yu Zheng, and Dekang Qi. 2016. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. CoRR (2016).

[18] Huaxiu Yao, Fei Wu, Jintao ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, and Jieping Ye. 2018. Deep Multi-View Spatial Temporal Network for Taxi Demand Prediction. (2018).

[19] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. CoRR (2014).

[20] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S. Yu. 2017. PredRNN: Recurrent Neural Networks for Predictive Learning using Spatiotemporal LSTMs. In NIPS.

[21] Dong Wang, Junbo Zhang, Wei Cao, Jian Li, and Yu Zheng. 2018.

When Will You Arrive? Estimating Travel Time Based on Deep Neural Networks. In AAAI.

[22] Chao Huang, Junbo Zhang, Yu Zheng, and Nitesh V Chawla. 2018. DeepCrime: Attentive Hierarchical Recurrent Networks for Crime Prediction. In CIKM.

[23] Yexin Li, Yu Zheng, and Qiang Yang. 2018. Dynamic Bike Reposition: A Spatio-Temporal Reinforcement Learning Approach. In KDD.

[24] R. Liu, S. Zhao, B. Cheng, H. Yang, H. Tang, and T. Li. 2020. Deep spatio-temporal multiple domain fusion network for urban anomalies detection. In CIKM.

[25] Anno, Soto, Kota Tsubouchi, and Masamichi Shimosaka. 2020. Supervised-CityProphet: Towards Accurate Anomalous Crowd Prediction. In SIGSPATIAL.

[26] Huang, C., Zhang, C., Zhao, J., Wu, X., Yin, D., & Chawla, N. 2019. Mist: A multiview and multimodal spatial-temporal learning framework for citywide abnormal event forecasting. In WWW.

[27] Huang, Chao, Xian Wu, and Dong Wang. 2016. Crowdsourcing-based urban anomaly prediction system for smart cities. In CIKM.

[28] Zhang, Huichu, Yu Zheng, and Yong Yu. 2018. Detecting urban anomalies using multiple spatio-temporal data sources. (2018).

[29] Zhang, M., Li, T., Shi, H., Li, Y., & Hui, P.. 2019. A decomposition approach for urban anomaly detection across spatiotemporal data. In IJCAI.

[30] Huang, Huiqun, Xi Yang, and Suining He. 2021. Multi-Head Spatio-Temporal Attention Mechanism for Urban Anomaly Event Prediction. (2021)

[31] Wu, X., Dong, Y., Huang, C., Xu, J., Wang, D., & Chawla, N. V. 2017. Uapd: Predicting urban anomalies from spatial-temporal data. (2017)

[32] Song, X., Zhang, Q., Sekimoto, Y., Shibasaki, R., Yuan, N. J., & Xie, X. 2015. A simulator of human emergency mobility following disasters: Knowledge transfer from big disaster data. In AAAI.

[33] Jiang, R., Song, X., Huang, D., Song, X., Xia, T., Cai, Z., ... & Shibasaki, R. 2019. Deepurbanevent: A system for predicting citywide crowd dynamics at big events. In KDD.

[34] Bao, H., Zhou, X., Zhang, Y., Li, Y., & Xie, Y. 2020. Covid-gan: Estimating human mobility responses to covid-19 pandemic through spatio-temporal conditional generative adversarial networks. In SIGSPATIAL.

[35] Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., & Liu, Z. 2020. Dynamic convolution: Attention over convolution kernels. In CVPR.

[36] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. 2017. Attention is all you need. In NIPS.

[37] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. 2014. Generative adversarial nets. In NIPS.

[38] Mirza, M., & Osindero, S. 2014. Conditional generative adversarial nets.

[39] Waldo R Tobler. 1970. A computer movie simulating urban growth in the Detroit region. Economic geography (1970).

[40] Hochreiter, S., & Schmidhuber, J. 1997. Long short-term memory. Neural computation (1997).

[41] 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.

[42] Zhang, Y., Li, Y., Zhou, X., Kong, X., & Luo, J. 2020. Curb-GAN: Conditional Urban Traffic Estimation through Spatio-Temporal Generative Adversarial Networks. In KDD.