HFSSG: A Hybrid Framework combining SIR pandemic model and Spatio-temporal Graph neural network forecasting the COVID-19 Impact in the United States

Liqin Zhang[1], Siyin Ma[2], Zhejian Jin[3]

Abstract Researches on infectious diseases has always been an important topic in interdisciplinary fields. The spread of infectious diseases not only depends on the attributes of diseases itself, but also depends on the network structure of the crowd. Since late 2019, COVID-19 has become a worldwide pandemic. In order to better understand and control the spread of COVID-19, many researchers collaborated and modeled this process to forecast its influence on the neighbouring areas. SIR model is a common compartmental model that well-defines disease transmission dynamics. However, it assumes a relatively closed system, and it does not take interactions among regions into consideration. Based on this, we utilize Graph Neural Network using data that describes inter-region interactions, which captures the features of messages among the regions. In this work, we propose a novel framework for COVID-19 case examination and prediction that uses Graph Neural Network. Different with existing time series models, this model learns from a spatio-temporal graph. We then evaluate this model on the California COVID-19 data set.

Key words Pandemic Prediction Model, Deep Learning, Graph Neural Network

1. Student ID: 517370910123, Email: graves_zhang@sjtu.edu.cn
2. Student ID: 517370910003, Email: msy841@sjtu.edu.cn
3. Student ID: 517370910167, Email: jinzhejian@outlook.com

Introduction

Problem Statement

With the rapid growth and spread of Covid-19, the ability to accurately forecast caseload is extremely important to help inform policymakers on how to provision limited healthcare resources, rapidly control outbreaks, and ensure the safety of the general public. In order to prepare, understand, and control the spread of the disease, we have come together in a collaborative effort to model and forecast COVID-19. In this project, we combine two powerful models (SIR and ST-GNN) into one integrate framework to perform better prediction capabilities in COVID-19 cases with lower cost. We propose a novel hybrid multi-network based framework named HFSSG that learns to select relevant edges and graph representations by pre-computing results from SIR model, along with RMSLE and Pearson values. On top of that, this model, with the help of ST-GNN, it learns the parameters for the pandemic model, and better fit the coronavirus model in real life. This main flowchart of our framework is shown in Figure 1.

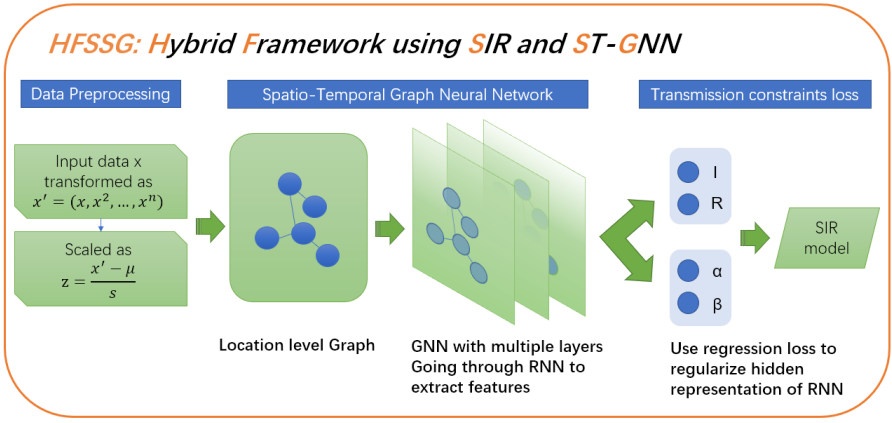

FIGURE 1. *Flowchart of our main framework*

Planned Approach

SIR SIR model is a simple method describing the epidemic spread that can be applied to any network structures. It divides the popularity into three categories, representing three stages of a person. The three functions with time t is used to simulate the three compartments.

1. Susceptible $S(t)$: people before getting infected, but can be infected by neighbors.
2. Infectious $I(t)$: people who have been infected, and can infect others.

3. Removed $R(t)$: people who endured a complete infection cycle and are immune to the disease.

The number of population is $N(t)$, where $N(t) = S(t) + I(t) + R(t)$. The SIR model describes the change between the three parts of people. We introduce two parameters β and γ.

- β Effective contact rate
- γ Recovery rate

To be more specific, β describes number of susceptible persons that a patient can infect per unit time is proportional to the total number of susceptible persons in the environment. γ implies that the number of people removed from the infected person per unit time is proportional to the number of patients. Three differential equations are used to sketch the model:

$$\begin{aligned}\frac{\mathrm{d}S}{\mathrm{d}t} &= -\frac{\beta SI}{N}\\ \frac{\mathrm{d}I}{\mathrm{d}t} &= \frac{\beta SI}{N} - \gamma I\\ \frac{\mathrm{d}R}{\mathrm{d}t} &= \gamma I\end{aligned}$$

We have to estimate the number of β and γ, where gradient descent or other methods would be applied to find the best parameters fitting the data of COVID-19.

GNN GNN is a suitable model to disseminate the geographical information as well as the human mobility across location to address the preliminary prediction with a certain amount of factors taken into account. In order to imitate infectious disease modeling, we plan to take create a graph with multiple nodes representing the states of a country and edges defining the spatial and temporal dependencies among states. We present our graph as a multi-layer stack, where each layer manifest the county connectivity graph for one day. Similar as the message-passing framework, we define the update at layer l as

$$\boldsymbol{m}_i^{(l+1)} = \sum_{j \in \mathcal{N}(i)} \mathcal{F}^{(l)}\left(\boldsymbol{h}_i^{(l)}, \boldsymbol{h}_j^{(l)}\right), \quad \boldsymbol{h}_i^{(l+1)} = \mathcal{G}^{(l)}\left(\boldsymbol{h}_i^{(l)}, \boldsymbol{m}_i^{(l+1)}\right)$$

where $\mathcal{F}^{(l)}$ and $\mathcal{G}^{(l)}$ are learned message functions and node update functions respectively, $\boldsymbol{m}^{(l)}$ are the messages passed between nodes, and $\boldsymbol{h}^{(l)}$ are the node representations. Notice that the bottom layer representing the date cases began appearing in the US and the top layer representing the newest date in our dataset.

Related Works

Various work on SIR model and its extensions has been applied on COVID-19 analysis. Hao et al.[4] reconstruct the outbreak in Wuhan (China) using an extended SAPHIRE model including seven compartments. A CovsirPhy Development Team[5] provided data analysis methods in python package based on SIR-derived models including SIR-F/SIR-FV/SEWIR-F. Yang et al.[6] used previous pandemic data to pretrain the LSTM, and then apply it to predict COVID-19 progression in China. However, different pandemics have different infect ability, so it may lead to inferior prediction results if the model transfer previous pandemic progression directly at the early stage of the pandemic. We will mainly focus on SIR fitting with least squares. We simplify the compartment process since our approach is to apply SIR result in featurization of graph neural network.

Apart from traditional epidemic infectious models like illness incidence forecasting and dynamic disease transmission model as SIR, SEIR[7]. Many novel approaches are based on deep neural networks. Deng et al.[8] proposed a graph message passing framework to combine learned feature embeddings and an attention matrix to model disease propagation over time. Google research team Kapoor et al.[9] apply a simple graph neural network for COVID-19 forecast prediction.

Moreover, some hybrid model has ingeniously combined disease transmission model and gnn model. Jie Zhou, Ganqu Cui etal.[10] combine SEIR and RNN and develop a spatio-temporal graph neural model, where the node feature is based on SEIR and the edge feature is based on the RNN model.

Datasets

We make full use of two datasets: the New York Times (NYT) COVID-19 dataset and the Google COVID-19 Aggregated Mobility Research Dataset. The NYT dataset collects all COVID-19 related data and is up to date til Nov.12 2020. It includes but not limits to the number of active cases, confirmed cases and deaths among different locations in the US. The Aggregated Mobility Research Dataset helps us understand the quantity of movement, while the other information combined make distinct node features for NYT. The following figure shows a overview of the infection rate in NY in the United States.

4. Xingjie Hao 2020.
5. Lisphilar 2020.
6. Yang et al. 2020.
7. Pei and Shaman 2020.
8. Deng et al. 2019.
9. Kapoor et al. 2020.
10. Zheng et al. 2020.

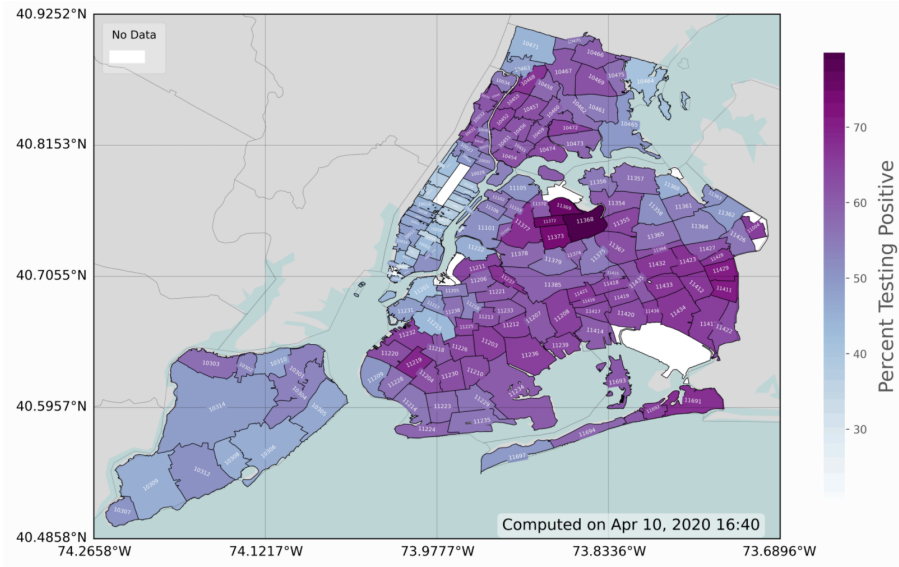

FIGURE 2. *The zip code map infection rate of New York City*

Initial findings

SIR

SIR model divide the population into three compartments, susceptible, infectious and recovered. There are two parameters, β and γ. The problem is how to fitting the two parameters to data. We apply the least square method, which allow us to fit β and γ simultaneously. It means that the sum of squared errors is a surface in two dimenions and we are looking for the minimum of this surface. This process is referred to as optimization and, fortunately for us, there are many robust algorithms available for this purpose. One of them, the Nelder-Mead algorithm, is the default in the function $optim$ in R. So we can use it to finish the fitting process. And two times of $optim$ will be applied since the first result might not be the most accurate one.

Before that, we look at the data of cases in the united states. We focus on the susceptible, infectious and recovered population from the dataset[11] confirmed, recovered and death population.

The plot help us get a basic idea of the epidemic condition in the USA. Then we apply the referred method to fit the parameters and finally our result is not very accurate. That is basically because the parameters are always changing with the real

11. Guidotti and Ardia 2020.

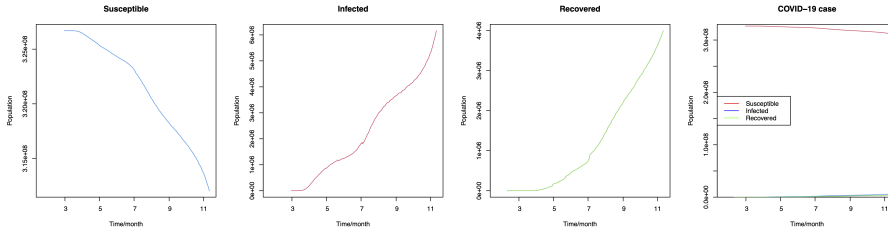

FIGURE 3. *Real SIR cases in the USA.*

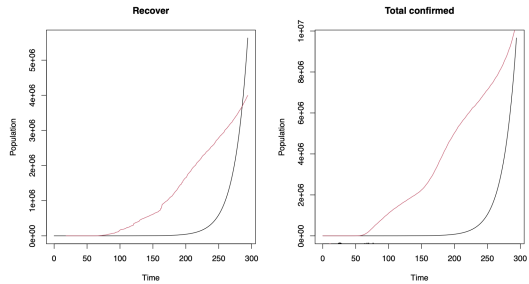

FIGURE 4. *Stable SIR model fitting.*

events like the release of policies and enhance of people's awareness in wearing masks. So we have to find a way to generate a dynamic or times series method fitting the transmission rate and recovery rate. The original way is not applicable. So we design HFSSG to fit the dynamic parameters.

Spatio-Temporal Graph Neural Network

Modelling the Graph

Graphs are natural representations for a wide variety of real-life data. Spatio-temporal graphs are a kind of graph that model connections between nodes as a function of time and space, and have found uses in a wide variety of fields.
To capture the spatio-temporal epidemic/pandemic dynamics, we represent all the input data as a 3D tensor, including the state location, time stamp, and the features such as the usage of masks associated with each location as the third dimensions. To be more specific, we construct on the following.

Graph nodes For attributed graph $G(V, E)$ representing the input data, we define each state be associated with a graph node that feature matrix that contains both static and dynamic features across all the time stamps for that location.

Graph edges In order to model spatial and temporal dependencies, we want to create a graph with different edge types. In the spatial domain, edges represent direct location-to-location movement and are weighted based on normalized mobility flows. In the temporal domain, edges simply represent binary connections to past days.

Hyper parameters and Featuraization

Age and Sexuality To gain some insights as we are constructing the graph neural network, we notice that different aged groups and different sexuality tend to have different symptoms facing the COVID-19. As a result, we decide to extract the feature and design a factor representing those information, since the average values across all aged groups or bi-sexuality won't be useful when training our GNN model.

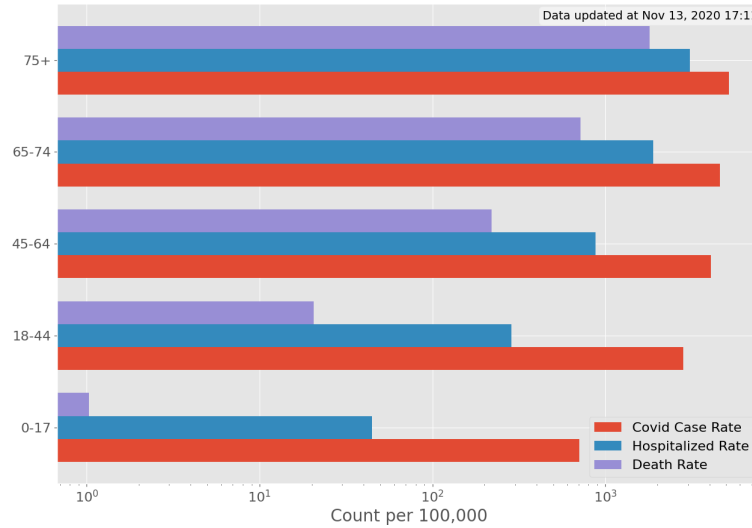

FIGURE 5. *Covid-19 Cases/Hospitalized/Death rate among different age groups*

From the above figure, we find that other than the structural node labels, the node information matrix X also provides opportunities to include explicit features, by concatenating each node's embedding vector to its specific row in X, we can make the HFSSG model simultaneously learn from all types of features.

Use GCN to Extract Spatial Features

With considering the disease transmission status of similar locations, more accurate prediction can be made. Here we apply the Graph Convolution Networks (GCN) model to extract spatial features.

We construct a two-layer GCN. First we build a sliding window. Let $\boldsymbol{X}_t = [\boldsymbol{X}_1, \boldsymbol{X}_2]$ with $\boldsymbol{X}_1 \in \mathbb{R}^{N\times T\times L_I D_1}$, $\boldsymbol{X}_2 \in \mathbb{R}^{N\times T\times L_I D_2}$, where N is the number of nodes,T is the

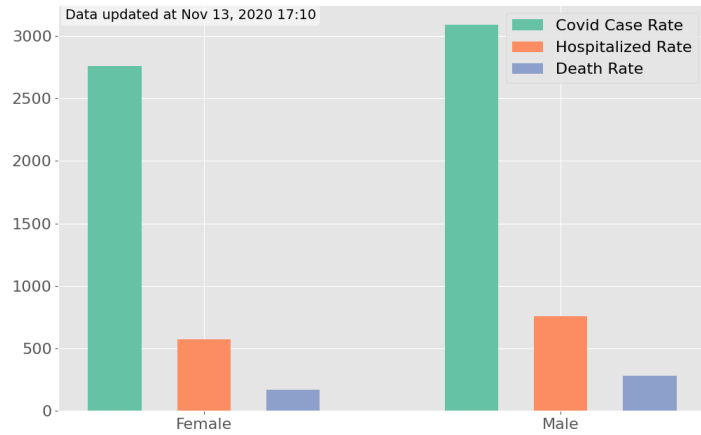

FIGURE 6. *Covid-19 Cases/Hospitalized/Death rate among different sexuality*

number of time stamps, L_1 is the length of the input sliding window, D_1 is the number of static features and D_2 is the number of dynamic features. Then, the two-layer GCN can be expressed as:

$$z_t = \widehat{A}\,\text{Relu}\left(\widehat{A}X_tW_0\right)W_1$$

where A denotes the normalized adjacency matrix of the graph G, W_0, W_1 denote the weight matrix in the first layer and the second layer.

Use RNN to Extract Temporal Features

The node embedding contains spatial features extracted from the graph. We also want to use historical temporal patterns to make a better prediction. We input the node embedding into Gate Recurrent Unit (GRU) in order to learn temporal features. We use max-pooling to integrate node embeddings for each location, which is:

$$\widetilde{Z}_t = \text{maxpool}(\widetilde{Z_t^0}, \widetilde{Z_t^1}, ..., \widetilde{Z_t^N})$$

Then GRU's hidden representation can be calculated as:

$$h_t = \text{GRU}(\widetilde{Z}_1, \widetilde{Z}_2, ..., \widetilde{Z}_t)$$

Long Short Term Memory (LSTM) is a variant of Recurrent Neural Network (RNN) that is used to overcome the limitations of RNN. LSTMs are capable to learn long term dependencies by replacing the hidden layers of RNN with memory cells. Different gate units such as input gate (it), output gate (ot), forget gate (ft) along with the activation function are used to model LSTMs and learn the behavior of temporal correlations. The working procedure of the LSTM cell is also defined mathematically

below, where is logistic sigmoid function, i, f, c, o are input gate, forget gate, memory cell and output gate respectively. W_{xi}, f, c, o are diagonal weight matrices from memory cell to gate units. In this paper, three variants of LSTM are used to carry out experimentation and are explained in further sections.

$$
\begin{aligned}
i_t &= \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right) \\
f_t &= \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \\
c_t &= f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \\
o_t &= \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right) \\
h_t &= o_t \tanh\left(c_t\right)
\end{aligned}
$$

Difficulties so far

We have discovered the weakness of stable parameter fitting in SIR models. So we are working on how to use Graph neural network for dynamic prediction.

- Our first difficulty is to select a proper diagram that can effectively give the parameter predictions. Based on the researches, we know that some neural network like LSTM and RNN have been applied to the combination with SIR or SEIR.
  Our plan is to study the difference between available neural networks and select the most suitable one through experiments.
- Another difficulty for our GNN model is that although we use graph convolution model to extract the spatio and time features, and then we use geographical proximity and demographic similarity between locations, we don't theeffect of each feature.
- We haven't thought of a way to tackle the potential overfitting issue.

Appendix

Visualization of New York Times COVID-19 Dataset

Featurization of Sexuality

```
import csv
import matplotlib.pyplot as plt
import datetime
import numpy as np

plt.style.use('ggplot')

age_data = './boroughs-by-sex.csv'
txt = []
```

```

with open(age_data, 'r', newline='') as file:
    reader = csv.reader(file)
    for line in reader:
        txt.append(line)

header = txt[0]

fig,ax = plt.subplots(figsize=(12.5,7))
spacer = -0.25
cii = 0
for plot_indx in range(4,7):
    data_to_plot,x_range = [],[]
    for jj in range(1,len(txt)-1):
        x_range.append(txt[jj][0])
        data_to_plot.append(float(txt[jj][plot_indx]))
    x_plot = np.arange(0,len(x_range))+spacer
    hist =
    ↪  ax.bar(x_plot,data_to_plot,label=header[plot_indx].replace('_','
    ↪  ').title(),width=0.15,color=plt.cm.Set2(cii))
    spacer+=0.25
    cii+=1

ax.set_xticks(np.arange(0,len(x_range)))
ax.set_xticklabels(x_range)
ax.legend(fontsize=16)
ax.tick_params('both',labelsize=16)
# textbox showing the date the data was processed
txtbox = ax.text(0.0, 0.975, 'Data updated at
↪  '+datetime.datetime.now().strftime('%b %d, %Y %H:%M'),
↪  transform=ax.transAxes, fontsize=14,
        verticalalignment='center', bbox=dict(boxstyle='round',
        ↪  facecolor='w',alpha=0.5))
txtbox.set_x(0.36-(txtbox.figure.bbox.bounds[2]-(txtbox.clipbox.bounds[2]-t
fig.savefig(header[0]+'_in_nyc.png',dpi=300,facecolor='#FCFCFC',bbox_inches
↪  = 'tight')
plt.show()
```

Featurization of different age group

```
import csv
import matplotlib.pyplot as plt
import datetime
import numpy as np
```

```

plt.style.use('ggplot')

age_data = './boroughs-by-age.csv'
txt = []

with open(age_data, 'r', newline='') as file:
    reader = csv.reader(file)
    for line in reader:
        txt.append(line)

header = txt[0]

fig,ax = plt.subplots(figsize=(20,10))
spacer = -0.25
for plot_indx in range(4,7): ## Only focus on BK_CASE_RATE,BK_HOSPITALIZED_RATE,BK_DEATH_RATE
    data_to_plot,x_range = [],[]
    for jj in range(1,len(txt)-1):
        x_range.append(txt[jj][0])
        data_to_plot.append(float(txt[jj][plot_indx]))
    # print(data_to_plot)
    x_plot = np.arange(0,len(x_range))+spacer
    hist = ax.barh(x_plot,data_to_plot,label=header[plot_indx].replace('_',' ').title(),height=0.25,log=True)
    spacer+=0.25

ax.set_xlabel('Count per 100,000',fontsize=20)
ax.set_yticks(np.arange(0,len(x_range)))
ax.set_yticklabels(x_range)
ax.legend(fontsize=16)
ax.tick_params('both',labelsize=16)
# fig.suptitle('COVID-19 in NYC by '+header[0].replace('_',' ').title(),x=0.4,y=0.92,fontsize=18)

txtbox = ax.text(0.0, 0.975, 'Data updated at '+datetime.datetime.now().strftime('%b %d, %Y %H:%M'), transform=ax.transAxes, fontsize=14,
        verticalalignment='center', bbox=dict(boxstyle='round', facecolor='w',alpha=0.5))
txtbox.set_x(1.04-(txtbox.figure.bbox.bounds[2]-(txtbox.clipbox.bounds[2
```

```
fig.savefig(header[0]+'_in_nyc.png',dpi=300,facecolor='#FCFCFC',bbox_inches
↪  = 'tight')
plt.show()
```

Visualization of Traditional SIR Method

```
library (readr)
library("COVID19")
require(deSolve)
usa_data <- covid19('USA')
par(mfrow=c(1,4))
plot(x = usa_data$date, y = 326687501 - usa_data$deaths -
↪  usa_data$confirmed - usa_data$recovered, type = 'l', col =
↪  4,xlab = "Time/month", ylab = 'Population', main =
↪  'Susceptible')
plot(x = usa_data$date, y = usa_data$confirmed -
↪  usa_data$recovered - usa_data$deaths, type = 'l',col = 2,
↪  xlab = "Time/month", ylab = 'Population', main = 'Infected')
plot(x = usa_data$date, y = usa_data$recovered, type = 'l',col =
↪  3, xlab = "Time/month", ylab = 'Population', main =
↪  'Recovered')

plot(x = usa_data$date, y = 326687501 - usa_data$deaths -
↪  usa_data$confirmed - usa_data$recovered,
     type = 'l',col = 2, xlab = "Time/month", ylab =
      ↪  'Population', main = 'COVID-19 case',
     ylim = c(0, 3.3*10E7)
     ,yaxs = "i"
     )
lines(x = usa_data$date, y = usa_data$confirmed -
↪  usa_data$recovered - usa_data$deaths, col = 4)
lines(x = usa_data$date, y = usa_data$recovered, col = 3)
legend("left", legend=c("Susceptible", "Infected","Recovered"),
       col=c("red", "blue", "green"), lty=1)
```

References

Deng, Songgaojun, Shusen Wang, Huzefa Rangwala, Lijing Wang and Yue Ning. 2019. Graph Message Passing with Cross-location Attentions for Long-term ILI Prediction (December).

Guidotti and Ardia. 2020. Covid-19 Data Hub. Available at <https://covid19datahub.io/>.

Kapoor, Amol, Xue Ben, Luyang Liu, Bryan Perozzi, Matt Barnes, Martin Blais and Shawn O'Banion. 2020. Examining COVID-19 Forecasting using Spatio-Temporal Graph Neural Networks. arXiv: 2007.03113 [cs.LG].

Lisphilar. 2020. CPython package for COVID-19 analysis with SIR-derived ODE models. Available at <https://github.com/lisphilar/covid19-sir>.

Pei, Sen, and Jeffrey Shaman. 2020. Initial Simulation of SARS-CoV2 Spread and Intervention Effects in the Continental US. *medRxiv,* https://doi.org/10.1101/2020.03.21.20040303. eprint: https://www.medrxiv.org/content/early/2020/03/27/2020.03.21.20040303.full.pdf. Available at <https://www.medrxiv.org/content/early/2020/03/27/2020.03.21.20040303>.

Xingjie Hao, et al. 2020. Reconstruction of the full transmission dynamics of COVID-19 in Wuhan. *Nature* 584:420–424.

Yang, Zifeng, Zhiqi Zeng, Ke Wang, Sook-San Wong, Wenhua Liang, Mark Zanin, Peng Liu et al. 2020. Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions. *Journal of Thoracic Disease* 12 (3). ISSN: 2077-6624. Available at <http://jtd.amegroups.com/article/view/36385>.

Zheng, Yunling, Zhijian Li, Jack Xin and Guofa Zhou. 2020. A Spatial-Temporal Graph Based Hybrid Infectious Disease Model with Application to COVID-19. *arXiv preprint arXiv:2010.09077*.