A Novel Spatiotemporal Method for Predicting Covid-19 Cases

JUNZHE CAI, PETER Z. REVESZ,
University of Nebraska-Lincoln,
Lincoln, NE 68516
USA

Abstract- Prediction methods are important for many applications. In particular, an accurate prediction for the total number of cases for pandemics such as the Covid-19 pandemic could help medical preparedness by providing in time a sucient supply of testing kits, hospital beds and medical personnel. This paper experimentally compares the accuracy of ten prediction methods for the cumulative number of Covid- 19 pandemic cases. These ten methods include three types of neural networks and extrapola- tion methods based on best fit quadratic, best fit cubic and Lagrange interpolation, as well as an extrapolation method proposed by the second author. We also consider the Kriging and inverse distance weighting spatial interpolation methods. We also develop a novel spatiotemporal prediction method by combining temporal and spatial prediction methods. The experiments show that among these ten prediction methods, the spatiotemporal method has the smallest root mean square error and mean absolute error on Covid-19 cumulative data for counties in New York State between May and July, 2020.

Keywords: Covid-19, inverse distance weight- ing, Kriging, neural network, prediction.
Received: April 30, 2021. Revised: June 2, 2021. Accepted: June 4, 2021. Published: June 11, 2021.

1. Introduction

IN many applications, the value of a spatiotemporal variable needs to be predicted for some time in the future based on previously measured data at the same location and neighboring locations. Some well-known applications include the prediction of economic indicators, such as stock prices, GDP or unemployment figures. In this paper, we take a look at predicting the number of cases of the Covid-19 pandemic [1], which is a novel type of pandemic with no well-tested prediction algorithms for it. Earlier epidemics prediction algorithms exist but they often require extra information like infected animals that are not available or applicable in this case [2]. Therefore, we focus on the Covid-19 pandemic in this paper, although our novel spatiotemporal interpolation algorithm may also be applicable to other spatiotemporal interpolation problems [3].

There are only a few publications that use Covid-19 data together with geographic information. Liu et al. [4] analyzes the combination of Covid-19 data and travel data in Wuhan, China and showed that travel restrictions were useful in curbing the spread of the pandemic. Thakar [5] generates an approximate density map for Covid-19 patients using location information such as school or work location from publicly available news articles in Washington State. Wang et al. [6] developed an algorithm that can estimate if a ship contains a risk of Covid-19 infections based on some information about the ships and their travel paths. These works are applicable only when the required patient address or travel data are available. In contrast, our prediction algorithms work without the need for such detailed information. Thomas et al. [7] presented a Covid-19 diffusion model based on interpersonal contact networks. While this may give more accurate predictions than other pandemic models, it requires interpersonal contact information, which is not generally available.

The rest of this paper is organized as follows. Section II. reviews some previously proposed prediction methods. Section III. presents a novel spatiotemporal prediction method. Section IV. presents an experiment that compares the various prediction methods on Covid-19 data from the state of New York. Section gives a discussion of the results. Finally, Section VI. presents some conclusions and future work.

2. Materials and Methods

In this section we review previous prediction methods. The prediction methods include temporal extrapolation methods (Section A.), spatial extrapolation methods (Section B.), and neural networks (Section C.). In addition, Section D. reviews the concept of moving average. Finally, Section E. reviews the error measures used in this paper. Every interpolation method has a function that can be applied to any temporal value even a value higher than all the values in the raw data. In this way, an interpolation method can be also used for extrapolation, that is, for predicting the outcome in the future.

2.1 Temporal Extrapolation Methods

Let y_i be the number of cases of the Covid-19 pandemic at some location i days ago. Hence y_1 is the number of cases yesterday, and y_2 is the number of cases the day before yesterday etc. Then the *Best Fit Cubic* and the *Lagrange* interpolation methods [8] can be used to predict the number of cases of the Covid-19 pandemic at that location. These methods derive interpolation functions into which we can place any future time instance to get a prediction value. In addition, the exponential decay temporal method, which was highly accurate for predicting election outcomes [9], can be used to get an estimate for the current day using the following formula, which assumes that we know the number of cases during the six previous days:

$$y = \frac{y_1}{2} + \frac{y_2}{4} + \frac{y_3}{8} + \frac{y_4}{16} + \frac{y_5}{32} + \frac{y_6}{32} \qquad (1)$$

The above formula can be extended for more numbers of days. The important feature is that the weights are successively diminishing by half except in the last instance, where the last weight is equal to the previous weight. Note that in this way, the sum of all the weights is exactly one. Finally, another prediction method that was proposed by Revesz [10] uses the following formula to predict the number of cases of the Covid-19 pandemic, where t is the number of days ahead from the last data. In other words, if the last data is for yesterday, then predicting for today means $t = 1$ and for tomorrow $t = 2$ etc.

$$y = \left(1 + t + \frac{t^2}{2}\right) y_1 + (t + t^2) y_2 + \frac{t^2}{2} y_3 \qquad (2)$$

2.2 Spatial Extrapolation Methods

Inverse Distance Weighting (IDW) [11] is a common spatial interpolation method. It is used when the interpolated variable at a location has a weighted relationship with its neighbors and when that relationship varies with distance. If a neighbor is closer than another neighbor, then the weight of the former will be higher than the weight of the latter. We use λ_i as the weight, y_i as the interpolated variable, and d_i as the distance to the i^{th} neighbor [12]. Then the Inverse Distance Weighting equation for the interpolated variable y at a location can be written in terms of its neighbors as follows:

$$y = \sum_{i=1}^{N} \lambda_i \times y_i \qquad (3)$$

where the equation for calculating λ_i can be written as follows:

$$\lambda_i = \frac{\left(\frac{1}{d_i}\right)^P}{\sum_{k=1}^{N} \left(\frac{1}{d_i}\right)^P} \qquad (4)$$

The p (power) value can be any number ≥ 1. For simplicity, in this paper we assume that p = 1.

Kriging is based on the work of Krige [13]. Different from IDW, Kriging not only considers the distance, but also find the spatial structure inside the data. The basic formula for Kriging is the following:

$$Z(x_0) = \begin{bmatrix} z_1 & ... & z_n \end{bmatrix} * \begin{bmatrix} w_1 \\ ... \\ w_n \end{bmatrix} \qquad (5)$$

Where $Z(x_0)$ is the predicted value at location x_0, $z_1 ... z_n$ are the values of the neighbors of x_0, and $w_1 ... w_n$ are the weights of the neighbors, which can be calculated as follows:

$$\begin{bmatrix} w_1 \\ ... \\ w_n \end{bmatrix} = \begin{bmatrix} c(x_1, x_1) & ... & c(x_1, x_n) \\ ... & ... & ... \\ c(x_n, x_1) & ... & c(x_n, x_n) \end{bmatrix}^{-1} * \begin{bmatrix} c(x_1, x_0) \\ ... \\ c(x_n, x_0) \end{bmatrix}$$

where $c(x, y)$ is the covariance function, that is, $c(x, y) = Cov(Z(x), Z(y))$.

C. Neural Networks

We use two different types of neural networks in this paper: backpropagation neural networks and recurrent neural networks.

2.3.1 Backpropagation

Backpropagation (BP) is a learning algorithm that has been used very often in neural networks. Backpropagation first appeared in the work of Rumelhart et al. [14] in 1988. Their work shows that applying backpropagation often results in useful discoveries using gradient descent. During the training, when the hidden layer passes the values to the output layer, the backpropagation method will calculate the differences between the hidden layer values and the actual values. Backpropagation will then adjust the weight on the edges between the two layers and repeat passing the values back to hidden layer until the error is small enough to make sure that the neural network can produce an accurate prediction.

Figure 1 shows the example of backpropagation structure. Where:

$$\begin{aligned} Hidden_1 &= In_1 \times W_1 + In_2 \times W_4 + In_3 \times W_7 \\ Out_{Hidden_1} &= \frac{1}{1 + e^{-Hidden_1}} \end{aligned}$$

We can define the values of the output values of the other hidden nodes similarly to the above. Finally, the outputs of the neural network is defined as follows.

$$\begin{aligned} Out_1 &= Out_{Hidden_1} \times W_{10} + Out_{Hidden_2} \times W_{13} \\ &\quad + Out_{Hidden_2} \times W_{16} \end{aligned} \qquad (6)$$

The goal of this step is to find the best weights (W_i) for the neural network to learn.

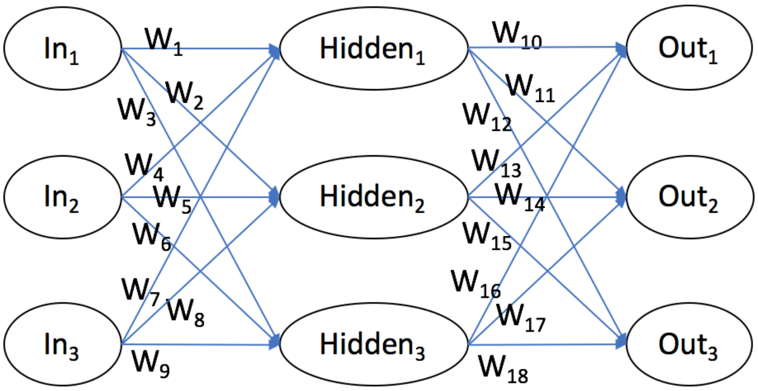

Fig. 1: Backpropagation example.

2.3.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) [15] improve backpropagation with the goal of better predicting the outcomes of a time series, such as in motor control and rhythm detection. Figure 2 shows the architecture of the RNN, which differs from other neural networks in that RNN contains one or more than one loop between nodes. RNN has a limit when dealing with back-propagated error. One of the extensions of RNN called LSTM (Long Short-Term Memory) allows the users to specify a limit. Different from Traditional RNN, LSTM only reads the input from the current time when doing a time series prediction which makes it more efficient than the traditional RNN [15].

LSTM is widely used to forecast data in many areas. Kong et al. [16] used LSTM to forecast short-term resident load. Their experiment showed that among all the prediction methods they selected, LSTM has the most accuracy. Huang et al. [17] used the past PM 2.5 concentration and weather report data to predict the PM 2.5 concentration in the future. The result proves the ability of LSTM to predict PM 2.5. Sagheer et al. [18] developed a model based on LSTM that can deal with most time-series prediction problems. They verified experimentally that their model works well on time series problems regarding petroleum production.

2.4 Moving Average

A moving average is applied for smoothing the raw data. That means that rather than using the raw data for a single day, we use the moving average value for seven days. For example, in the county of Albany in the state of New York, the number of Covid-19 cases for the days from July 1 to July 7 were the following in order: 2112, 2125, 2130, 2145, 2152, 2160 and 2164. Hence the seven day moving average centered on July 4th is the average of these seven values divided by the population of that county, which is 0.3 million, which gives 7008.5 cases per million people.

2.5 Error Measures

To experimentally evaluate the accuracy of the interpolation methods, we use the *Mean Absolute Error* (MAE) and the *Root Mean Square Error* (RMSE) measures, which are defined as follows, where F_i is the predicted value and A_i is the corresponding actual value and N is the number of items:

$$MAE = \frac{\sum_{i=1}^{N} |F_i - A_i|}{N} \tag{7}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (F_i - A_i)^2}{N}} \tag{8}$$

Intuitively, a lower value of these error measures imply a higher quality interpolation and extrapolation or prediction. Conversely, a higher value of these error measures implies a lower quality.

3. Proposed Spatiotemporal Extrapolation Method

In this section we propose a novel spatiotemporal interpolation method that works in general for many types of data, including cumulative Covid-19 pandemic data. Before describing our spatiotemporal extrapolation algorithm, we remark that not all temporal and spatial extrapolation methods can be applied to cumulative data. In fact, we can show the following.

Theorem 1. The exponential decay extrapolation method underestimates the real value when the measured value is monotonically increasing.

Proof: When the measured value is monotonically increasing, then we have the following conditions:

$$y > y_1 > y_2 > y_3 > y_4 > y_5 > y_6 \tag{9}$$

Equation 9 implies the following:

$$\frac{y_1}{2}+\frac{y_2}{4}+\frac{y_3}{8}+\frac{y_4}{16}+\frac{y_5}{32}+\frac{y_6}{32} < \frac{y_1}{2}+\frac{y_1}{4}+\frac{y_1}{8}+\frac{y_1}{16}+\frac{y_1}{32}+\frac{y_1}{32}$$

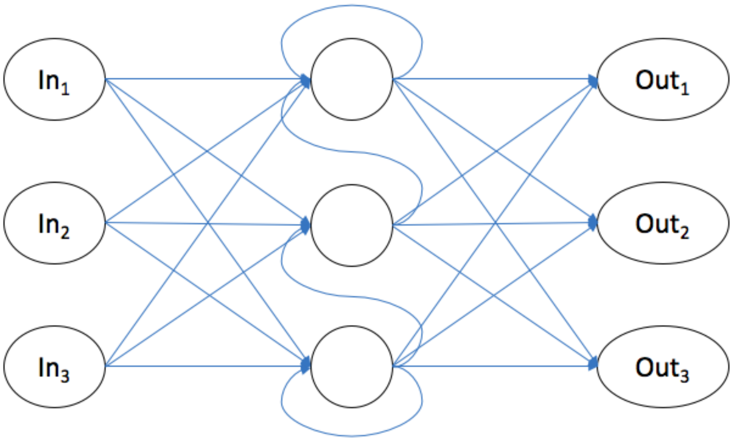

Fig. 2: Architecture of Recurrent Neural Network.

The above implies the following:

$$\frac{y_1}{2} + \frac{y_2}{4} + \frac{y_3}{8} + \frac{y_4}{16} + \frac{y_5}{32} + \frac{y_6}{32} < y_1 \qquad (10)$$

By Equation 1, the exponential decay extrapolation method's estimate for y is the left side of the above inequality. Hence the estimate for y is less than y_1, whereas $y > y_1$ because the measured value is monotonically increasing. Therefore, the exponential decay extrapolation method underestimates the value of y.

Theorem 1 implies that the exponential decay extrapolation method is not applicable for estimating cumulative data, which are inherently monotonically increasing. This theorem serves as a caution in applying known methods to our task.

3.1 Calculation of Distances between Neighboring Counties

Next we describe how we calculate the distances between neighboring locations. In the example below we consider the counties within the State of New York. Second, we calculate the distance between two counties i and j based on their centroids considering that they lie on the surface of the 3-dimensional earth, as follows. First, let R = 6368 kilometers (radius of the earth), and then take:

$$\begin{aligned} x_i &= R \times cos(long_i) \times sin(90^\circ - lat_i) \\ y_i &= R \times sin(long_i) \times sin(90^\circ - lat_i) \\ z_i &= R \times cos(90^\circ - lat_i) \end{aligned} \qquad (11)$$

Similarly, we have:

$$\begin{aligned} x_j &= R \times cos(long_j) \times sin(90^\circ - lat_j) \\ y_j &= R \times sin(long_j) \times sin(90^\circ - lat_j) \\ z_j &= R \times cos(90^\circ - lat_j) \end{aligned} \qquad (12)$$

Finally, the Euclidean distance in 3-dimensions between the two centroids can be found as follows:

$$distance = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2} \quad (13)$$

3.2 Combining Spatial and Temporal Extrapolation Methods to Form Spatiotemporal Methods

Intuitively, the number of cases of the Covid-19 pandemic can be better estimated by considering both temporal and spatial interpolations. If a county C has a very high number of Covid-19 cases, then the situation in its neighbors may not affect the development of the number of cases much and could even be ignored because most residents of C will catch the disease from other residents within county C. Therefore, the best temporal interpolation based on just that state's previous cases, denoted as $E_{t,C}$, likely would give the best prediction for the future.

On the other hand, if a county C has few Covid-19 cases relative to its neighbors, then the situation in its neighbors has to be carefully considered because in that case most residents of C could be infected by neighboring county residents when they travel and meet. Therefore, a spatial interpolation of the neighbors' future cases, denoted as $E_{s,C}$, likely would give the best prediction for the future cases in county C.

Preliminary experiments suggested that the above still needs to be refined because if one of the neighbors experiences an explosion in the number of cases, then it may not immediately cause an explosion in county C too. In other words, there is some time delay instead of an immediate effect. Therefore, in such cases the temporal interpolation $E_{t,S}$, would still likely give the most accurate prediction, while the spatial interpolation $E_{s,S}$ would be likely to give an overestimate of the number of Covid-19 pandemic cases. Therefore, we need to place some limit on the difference between the two estimates and ignore the spatial estimate if it is excessively larger than the temporal estimate. By testing values of multiples of ten, we found that 30 and 280 work best as the lower and upper bound values, respectively. Therefore, we refine the above formula as follows:

$$E_S = \begin{cases} E_{s,C} & \text{if } 30 < E_{s,C} - E_{t,C} < 280 \\ E_{t,C} & \text{otherwise} \end{cases} \qquad (14)$$

4. Results

In this section, we describe a computer experiment that compares several temporal, spatial and spatiotemporal extrapolation methods that are applicable to predicting the number of cumulative Covid-19 cases. This section is organized as follows. Section A. describes the data sources. Section B. describes the implementation of the algorithms that were tested. Section C. explains the experimental procedure and results.

4.1 Data Sources

First, we collected population data for each county of New York State from the *World Population Review* website [19]. Second, we obtained the centroid latitude and longitude of each county from the United State Census Bureau website [20]. Table 1 shows the latitude and the longitude of the centroid and the population of each county of New York State.

Next, we also obtained data about the cumulative number of Covid-19 cases in the counties of New York State during July 2020 from the *New York Times* [21]. The raw data show some fluctuations in the daily increases in the number of Covid-19 cases. Some of these fluctuations may reflect the true expansion of the disease. On the other hand, some fluctuations may be due to the differences between weekdays and weekends when more people are more likely to go for Covid-19 testing. Hence, it makes sense to smoothen the data by taking a moving average. We computed a seven day moving average based on the raw data and divided it by the population of each county. The seven day moving average was calculated as explained in Section D..

4.2 Implementation of the Algorithms

We implemented the temporal, IDW, and the spatiotemporal interpolation methods in MATLAB. For Kriging, we obtained the MATLAB function code from [22]. We obtained a MATLAB implementation of the LSTM recurrent neural network program from [23]. We adjusted the neural network structure to have an input layer with six nodes, a hidden layer with ten nodes, and an output layer with one nodes. We modified the code so that the recurrent neural network can accept six inputs and give one output.

For the backpropagation neural network, we used the program from [24]. We adjusted the neural network structure to have an input layer with six nodes, a hidden layer with ten nodes, and an output layer with one node. We modified the code so that the backpropagation neural network can accept six inputs and give one output.

4.3 Testing Procedure and Results

We did some preliminary experiments to fine tune the parameters used in all of the algorithms. In particular, we compared the accuracies of the methods using the raw, the five days moving average and the seven days moving average data. In general, all methods performed best with the seven days moving average except for the Revesz method. We also compared 3 versus 6 previous days' values as inputs for the Lagrange method with the result that it was more accurate with only 3 inputs. For the neural networks we also considered using a single neural network for each county versus using sixteen different neural networks for each county, where each neural network predicted for a particular number of days ahead between one and sixteen. In general, using sixteen different neural networks was more accurate for most counties. We also experimented with five, ten and fifteen hidden nodes in the neural networks. There was a significant improvement from five to ten hidden nodes but little or no improvement from ten to fifteen hidden nodes. Therefore, we used ten hidden nodes in the hidden layer of all the neural networks.

Table 2 shows the accuracy of training with 50, 100, 200 and 300 epochs for both neural networks. There was a significant improvement from 100 epochs to 200 epochs but little improvement 200 epochs to 300 epochs for backpropagation. There was a significant improvement from 50 epochs to 100 epochs but little improvement from 100 epochs to 200 epochs. and even smaller improvement from 200 to 300 epochs for recurrent neural network. To avoid overtraining, we used 200 epochs for both backpropagation and the recurrent neural network for training and the testing results.

Finally, for fine tuning the parameters of our spatiotemporal method, we tested multiples of ten for possible upper and lower bounds. The lower bound of 30 and the upper bound of 280 produced the most accurate result.

After fine tuning, our goal was to compare how well the various prediction methods predicted the moving average centered on days 7/10 (1 day ahead), 7/11 (2 days ahead), . . . , and 7/25 (16 days ahead). For our testing we divided the prediction methods into two groups based on the number of inputs that they use.

The first group used six inputs, which were the moving average data centered on days from 7/4 to 7/9. The first group included the Best Fit Linear, the Best Fit Quadratic and the Best Fit Cubic methods. The second group used only three inputs, which were the moving average data centered on days 7/7, 7/8 and 7/9. The second group included the Lagrange and the Revesz [10] methods. The Lagrange method was put into the second group because preliminary experiments showed that the Lagrange method with three inputs was more accurate than the Lagrange method with six inputs. On the other hand, the Best Fit Cubic and Best Fit Quadratic methods were better with six inputs than with three inputs. The Revesz method requires three inputs by definition.

A spatial interpolation-based way to predict n days ahead in county C_i the number of cumulative Covid-19 cases is the following two-step process. First, we predict n days ahead in all the neighbors of C_i the number of cumulative Covid-19 cases using the Best Fit Linear extrapolation. Second, we use either IDW or Kriging to predict n days ahead in county C_i the number of cumulative Covid-19 cases.

Our novel spatiotemporal prediction method chooses

Table 1: Latitude, longitude and population (in millions) of the counties in New York State. The data for New York City combine five counties.

County	Latitude	Longitude	Population	County	Latitude	Longitude	Population
Albany	42.58824	-73.97401	0.31	Niagara	43.456731	-78.792142	0.21
Allegany	42.247853	-78.026153	0.05	Oneida	43.242727	-75.434282	0.23
Broome	42.161977	-75.830283	0.19	Onondaga	43.006516	-76.196134	0.46
Cattaraugus	42.239099	-78.662332	0.08	Ontario	42.856357	-77.303497	0.11
Cayuga	43.008546	-76.574587	0.08	Orange	41.40241	-74.306252	0.38
Chautauqua	42.304216	-79.407595	0.13	Orleans	43.502287	-78.229726	0.04
Chemung	42.15528	-76.747179	0.08	Oswego	43.461443	-76.209262	0.12
Chenango	42.478024	-75.602241	0.05	Otsego	42.629776	-75.028841	0.06
Clinton	44.752712	-73.705643	0.08	Putnam	41.427907	-73.743861	0.10
Columbia	42.247729	-73.626806	0.06	Rensselaer	42.710421	-73.513845	0.16
Cortland	42.594039	-76.07624	0.05	Rockland	41.154628	-74.024662	0.33
Delaware	42.193986	-74.966728	0.04	Saratoga	43.106135	-73.855387	0.23
Dutchess	41.75477	-73.740041	0.29	Schenectady	42.817552	-74.043559	0.16
Erie	42.752759	-78.778192	0.92	Schoharie	42.591294	-74.438172	0.03
Essex	44.109601	-73.778431	0.04	Schuyler	42.419776	-76.938603	0.02
Franklin	44.594376	-74.31067	0.05	Seneca	42.782294	-76.827088	0.03
Fulton	43.115609	-74.423678	0.05	St. Lawrence	44.488112	-75.074311	0.11
Genesee	43.00091	-78.192778	0.06	Steuben	42.266725	-77.385525	0.10
Greene	42.279821	-74.142025	0.05	Suffolk	40.943554	-72.692218	1.48
Hamilton	43.657879	-74.502456	0.00	Sullivan	41.719993	-74.771577	0.08
Herkimer	43.407489	-75.011683	0.06	Tioga	42.178057	-76.297456	0.05
Jefferson	43.996389	-76.052968	0.11	Tompkins	42.453006	-76.473483	0.10
Lewis	43.782681	-75.44414	0.03	Ulster	41.947212	-74.265458	0.18
Livingston	42.727485	-77.769779	0.06	Warren	43.555105	-73.838139	0.06
Madison	42.910026	-75.663575	0.07	Washington	43.312377	-73.439428	0.06
Monroe	43.464484	-77.664658	0.74	Wayne	43.458758	-77.063164	0.09
Montgomery	42.900891	-74.435357	0.05	Westchester	41.152686	-73.745753	0.97
Nassau	40.729612	-73.589414	1.36	Wyoming	42.701363	-78.228567	0.04
New York C.	40.776642	-73.970187	8.18	Yates	42.638237	-77.104324	0.02

Table 2: Training accuracy

	Number of epochs			
	50	100	200	300
BP	76.37%	89.88%	95.58%	95.83%
RNN	87.33%	97.02%	98.54%	99.12%

between either the above IDW-based prediction or the prediction n days ahead in county C_i of the number of cumulative Covid-19 cases using the Best Fit Linear extrapolation. The choice is guided by the conditions described in Section B.. The spatial interpolation-based and the novel spatiotemporal prediction methods are classified as belonging to the first group because they use the Best Fit Linear extrapolation method.

For the backpropagation and the recurrent neural networks, we collected raw data from 6/1 to 6/28 too. Then for each county, we generated moving average data divided by the population of the county from 6/4 to 6/25. Next we trained 16 separate neural networks for each county on the following set of training data, which are sequences with length 7:

For one day ahead, the training data includes:

1) input: 6/4 - 6/9 output: 6/10

2) input: 6/5 - 6/10 output: 6/11

3) input: 6/6 - 6/11 output: 6/12

For testing, we use 7/4 - 7/9 as input and expect the neural network to output data for 7/10.

For two days ahead, the training data includes:

1) input: 6/4 - 6/9 output: 6/11

2) input: 6/5 - 6/10 output: 6/12

3) input: 6/6 - 6/11 output: 6/13

For testing, we use 7/4 - 7/9 as input and expect the neural network to output data for 7/11.

We continue in this way until 16 days ahead, where the training data includes:

1) input: 6/4 - 6/9 output: 6/25

2) input: 6/5 - 6/10 output: 6/26

3) input: 6/6 - 6/11 output: 6/27

For testing, we use 7/4 - 7/9 as input and expect the neural network to output data for 7/25.

Table 3 shows the MAE of each day for the training.

Table 3: The MAEs of neural network training

Method	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
BP	350.34	78.03	478.80	305.96	148.13	595.34	189.46	41.98	22.22	144.01	252.39	256.51	340.12	435.33	628.30	115.00
RNN	101.20	101.48	101.75	102.02	102.29	102.56	102.84	103.11	103.38	103.65	103.94	104.23	104.54	104.85	105.16	105.48

Table 4: The RMSEs of the prediction methods

Type	Method	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Temporal	BP	764.00	752.14	766.07	779.29	755.01	776.73	760.16	803.16	865.63	791.65	807.15	797.66	825.19	813.24	838.31	824.05
Temporal	RNN	739.05	737.25	736.12	739.49	734.88	725.25	709.70	706.03	693.19	673.95	660.75	664.77	648.16	621.20	619.60	610.54
Temporal	Lagrange	5.37	14.00	26.37	41.62	61.34	85.30	115.56	148.90	186.94	228.66	274.96	323.36	375.64	430.90	490.15	553.12
Temporal	Revesz	3.93	11.11	21.90	35.42	53.27	75.30	103.52	134.72	170.73	210.31	254.58	300.98	351.24	404.64	461.87	522.89
Temporal	Cubic	5.21	15.68	33.48	58.37	93.23	140.86	203.05	279.79	374.79	490.02	625.62	782.50	963.68	1170.01	1403.81	1668.21
Temporal	Quadratic	6.19	14.64	27.13	43.15	63.81	88.13	115.28	143.85	175.38	209.64	247.25	287.34	330.52	376.87	428.01	483.01
Temporal	Linear	10.46	18.65	27.99	38.28	47.67	56.23	64.05	72.80	79.96	85.56	92.34	101.31	111.24	122.12	132.42	142.67
Spatial	Kriging	8375.16	8378.97	8383.33	8388.87	8394.33	8400.20	8405.39	8410.06	8412.73	8414.49	8415.58	8416.22	8416.68	8417.35	8418.45	8419.48
Spatial	IDW	3095.06	3096.19	3097.57	3099.77	3103.11	3107.46	3111.47	3115.72	3120.70	3125.66	3130.19	3133.49	3136.67	3139.77	3142.59	3145.07
Spatiotemp.	ST	17.74	23.75	30.46	38.49	45.93	52.73	58.68	65.76	71.99	77.42	84.11	92.94	102.85	113.87	123.84	133.71

Table 5: The MAEs of the prediction methods

Type	Method	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Temporal	BP	607.27	622.97	622.86	623.81	610.59	631.40	622.78	637.66	687.62	639.76	653.17	645.82	662.54	673.61	690.58	676.13
Temporal	RNN	589.97	593.11	596.91	602.48	601.95	596.83	587.78	585.18	574.30	569.64	554.03	561.33	543.66	527.51	525.57	523.69
Temporal	Lagrange	3.56	9.42	17.56	27.97	41.81	58.24	78.11	99.69	124.45	151.33	180.32	211.49	244.58	279.46	316.52	355.91
Temporal	Revesz	2.65	7.64	15.02	24.23	36.76	51.75	70.59	90.99	114.58	140.01	168.08	198.13	230.07	263.94	299.85	338.04
Temporal	Cubic	7.33	21.17	44.84	78.47	125.58	190.35	275.01	379.09	506.55	659.14	839.51	1047.27	1286.93	1559.27	1867.56	2214.14
Temporal	Quadratic	4.37	10.35	19.34	31.21	46.26	63.74	83.43	103.68	126.21	150.58	177.11	205.77	236.75	269.58	305.65	344.20
Temporal	Linear	7.01	12.94	19.99	27.07	34.13	41.21	47.70	54.52	60.32	64.46	69.33	76.27	83.59	91.32	99.75	108.71
Spatial	Kriging	3951.21	3960.78	3970.85	3981.79	3992.73	4003.48	4013.01	4021.52	4028.28	4033.79	4038.63	4043.29	4047.75	4052.93	4059.25	4065.45
Spatial	IDW	2003.26	2005.42	2007.66	2010.24	2013.33	2017.80	2022.77	2027.74	2032.73	2036.99	2040.92	2043.76	2046.84	2049.61	2052.07	2054.57
Spatiotemp.	ST	9.69	15.36	21.34	27.09	32.78	38.77	44.38	50.53	55.85	59.71	64.30	71.16	78.39	86.04	94.19	102.87

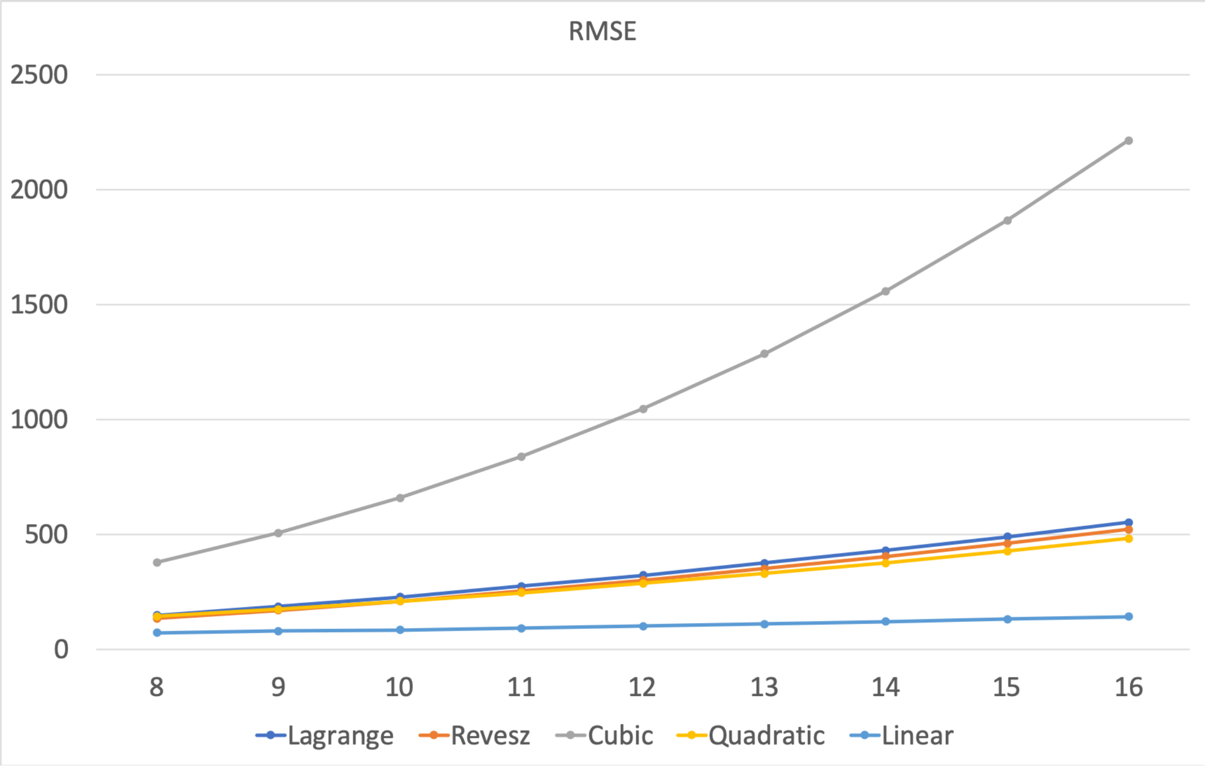

Fig. 3: RMSE of Temporal Methods.

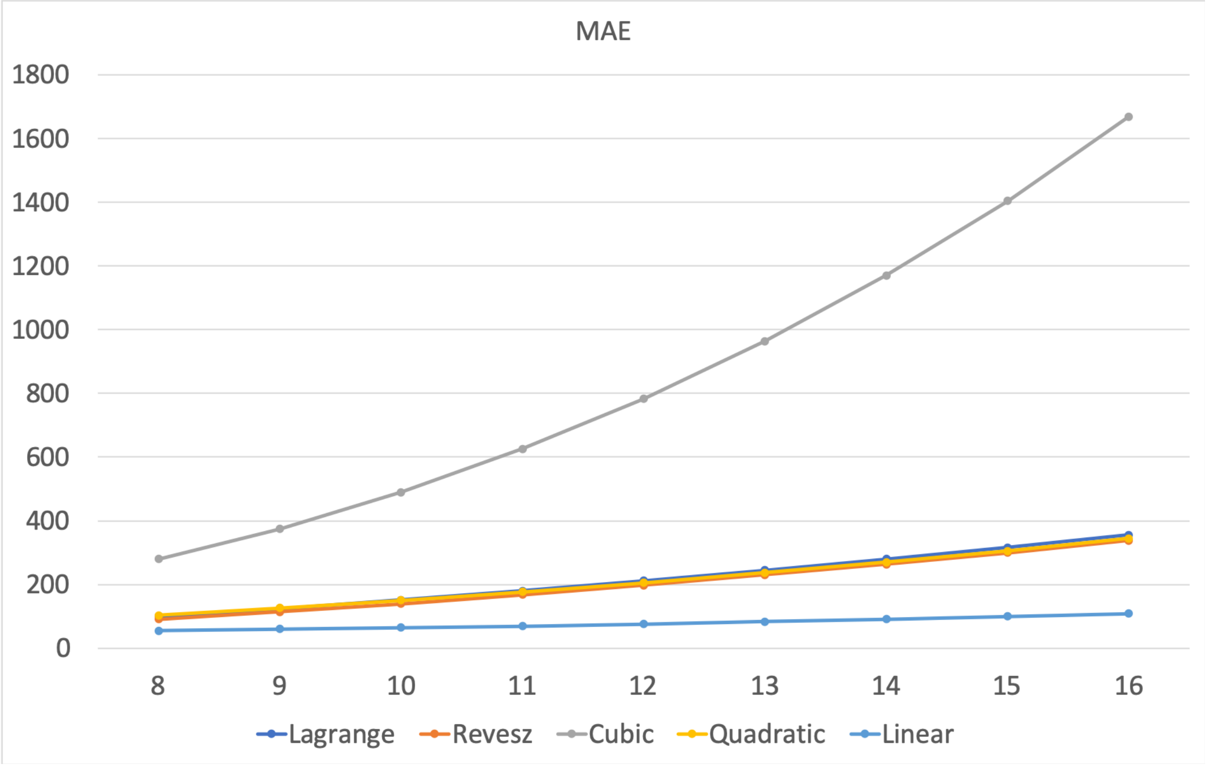

Fig. 4: MAE of Temporal Methods.

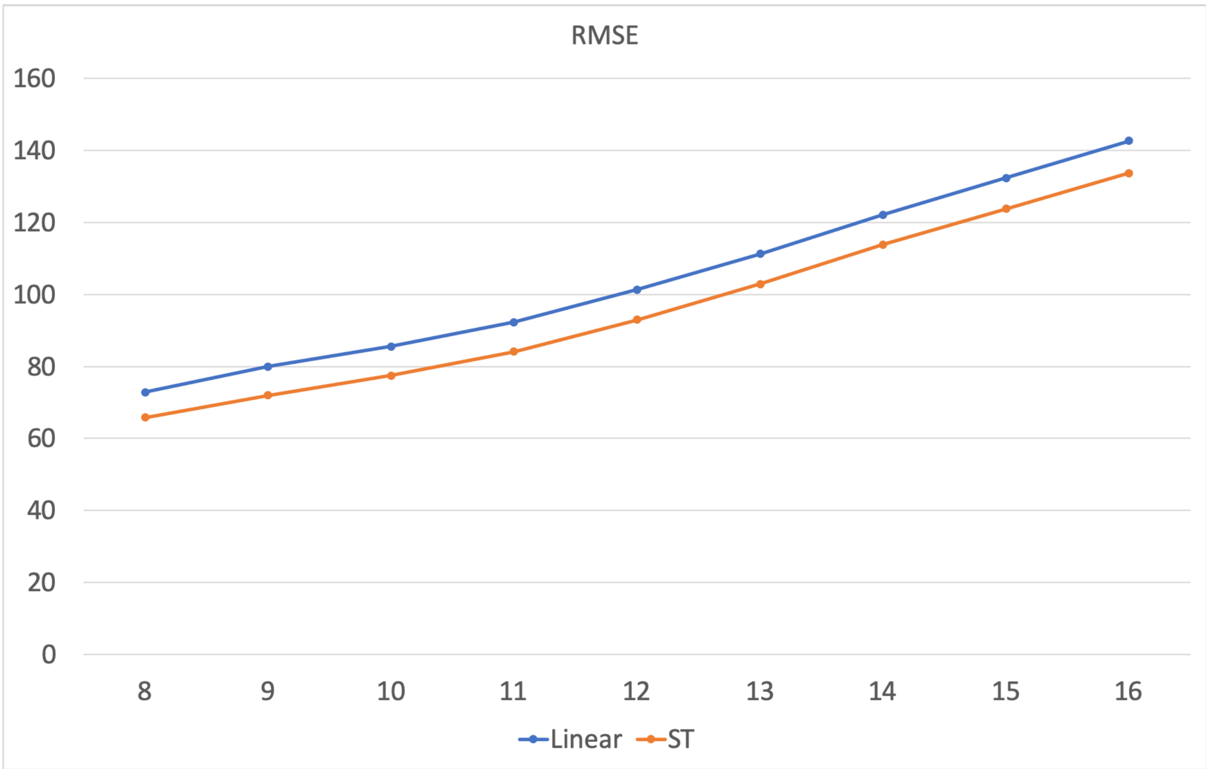

Fig. 5: RMSE of spatiotemporal Methods.

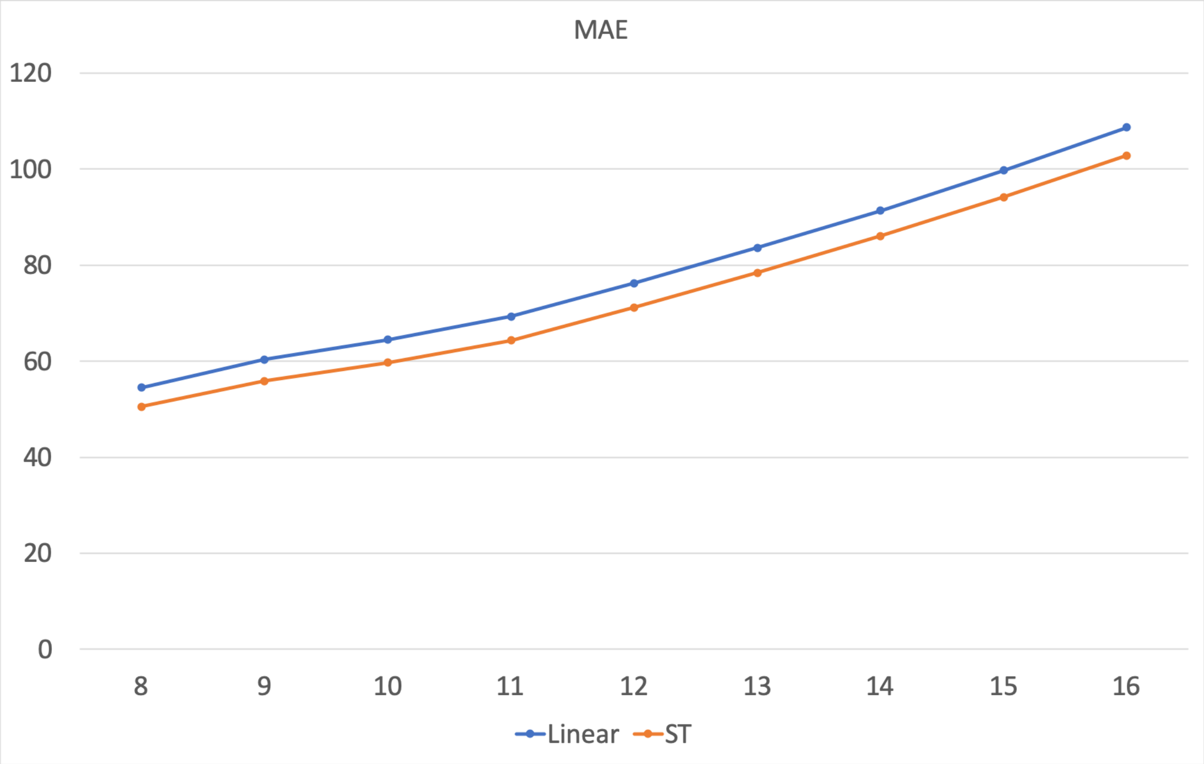

Fig. 6: MAE of spatiotemporal Methods.

5. Discussion

There are 58 counties and 16 neural networks for each, which is a total of 928 sequences in the training data set. During testing, we gave as an input to the neural network the moving average data centered on days 7/4 ... 7/9.

For all methods, we compared the predictions for the moving average centered on days 7/10 (1 day ahead), 7/11 (2 days ahead), . . . , and 7/25 (16 days ahead) with the actual values. We evaluated the root mean square error (RMSE) and the mean absolute error (MAE) measures for each prediction method as defined in Section E..

Table 4 shows the root mean square error (RMSE) for each prediction method when they were used to predict 1-16 days ahead. Similarly, Table 5 shows the mean absolute error (MAE) for each prediction method when they were used to predict 1-16 days ahead.

The average of the seven day moving averages centered on July 25 is 7952.15. Hence the ST method's MAE of 102.87 is equivalent to about a 1.29 percent error. For testing the spatial interpolation, we use the predict result of the Best Fit Linear since it has the highest accuracy among all temporal method we tested. The result for the spatial Interpolation shows that for IDW, the predict result for Dutchess and Tompkins counties, the overall result for IDW has lower error than the the Best Fit Linear method in those three states. For Kriging, the predict result for Tompkins and Yates county, the overall result for Kriging has lower error than Revesz method in those two states but IDW has the lowest error. The Figures 5 and 6 show the RMSE and MAE of the combined spatiotemporal method. The experiment indicates that our spatiotemporal prediction method works well for cumulative Covid-19 cases.

We analyzed different upper bounds and lower bounds to find the best way of combining the results of the best fit linear and IDW methods. After searching by a step size of 10 in the range from 10 to 350, we found that the lower bound of 30 and the upper bound of 280 gave the best combination of the two methods in terms of the lowest RMSE and MAE among all the upper bounds and lower bounds we tested.

We also analyzed the results for New York State in further detail by considering separately the counties that are mostly metropolitan areas versus the other counties that are mostly rural areas. For the spatiotemporal method, Tables 6 and 7 show the MAEs and RMSEs of the mostly metropolitan counties, which are Erie, Monroe, New York City, Onondaga, Schenectady and Westchester, and rest of the counties, which are mostly rural. Table 6 also shows the absolute difference between the metropolitan and the rural MAEs. The absolute differences are always below 17 with a mean of 9.7 over the sixteen days. The average number of Covid-19 cases for all counties ranged from 6603.73 to 6950.04.

The last row of Table 6 shows that the absolute differences in the MAEs make up only a small percent of the average number of Covid-19 cases. These percentages fluctuate slightly from 0.10% for one day ahead to 0.02% for twelve days ahead and to 0.24% for sixteen days ahead. The last row of Table 7 shows that the absolute differences in the RMSEs also make up only a small percent of the average number of Covid-19 cases and also fluctuate slightly from 0.22% for one day ahead to 0.04% for thirteen days ahead and to 0.31% for sixteen days ahead. Hence according to both the MAE and the RMSE measures there was no significant differences between the mostly metropolitan and the mostly rural counties within New York State. This suggests that similar accuracies can be obtained when the method is applied to other states, including mostly metropolitan states such as Rhode Island and mostly rural states like most states in the Mid-West.

6. Conclusions and Future Work

This paper compared ten prediction methods for cumulative Covid-19 cases in the counties of New York State. The number of methods was greatly extended compared to our earlier conference paper [25] and a smoothening of the raw data using a seven day moving averages and a new neural networks testing procedure was also introduced. One of the methods is a novel spatiotemporal method that combines a temporal extrapolation method with the IDW spatial interpolation method. Overall, this novel spatiotemporal prediction method was the best according to both the MAE and the RMSE error measures.

It remains to be seen whether the prediction method can be further improved. Generally, the spatial prediction methods are more accurate with denser spatial locations with measurement data. Hence the IDW method could improve if we have more than a single location for each county. Each county may be subdivided into smaller districts with their own separate measurements. With increased accuracy of the IDW method, our spatiotemporal interpolation method could also improve.

In addition, further improvements could be obtained by a direct assessment of test results from population surveys, as is done in the REal-time Assessment of Community Transmission (REACT) study in England [26]. The REACT study collected not only samples but in a survey additional information about the subjects, such as age, race, gender, occupation, mobility, and contact with infected persons. Such information facilitates a deeper analysis of the pandemic. For example, Ward et al. [26] could identify a two-to three-fold higher prevalence rate among health and care workers compared with non-essential workers and similarly higher prevalence rate in people of Black or South Asian than white ethnicity. Such detailed results have important implications for effectively fighting the pandemic. However, such detailed data is not available in all locations, for example in New York State. In such situations, our method can still give a prediction of the total number of expected cases without breaking down the cases by occupation and race. It would be an interesting future work to extend our method to the case when such detailed data is available.

Finally, we would like to suggest some future work. Our spatiotemporal algorithm was fine tuned by finding

Table 6: The spatiotemporal method's MAEs for the Metropolitan and Rural counties

County	1	2	3	4	5	6	7	8
Metropolitan MAE	3.65	5.92	10.76	16.26	22.29	28.68	34.58	39.23
Rural MAE	10.39	16.45	22.56	28.34	33.99	39.93	45.51	51.83
Absolute Difference of MAEs	6.73	10.53	11.80	12.09	11.70	11.25	10.93	12.60
Average Number of Cases	6603.74	6627.28	6652.34	6677.43	6701.85	6725.55	6748.49	6771.51
Percentage Difference	0.10%	0.16%	0.18%	0.18%	0.17%	0.17%	0.16%	0.19%
County	9	10	11	12	13	14	15	16
Metropolitan MAE	45.38	52.06	60.62	69.88	81.41	93.24	106.20	118.10
Rural MAE	57.06	60.59	64.72	71.30	78.04	85.21	92.80	101.11
Absolute Difference of MAEs	11.68	8.53	4.11	1.42	3.36	8.03	13.39	16.99
Average Number of Cases	6793.62	6814.36	6835.62	6858.13	6881.07	6904.41	6927.87	6950.04
Percentage Difference	0.17%	0.13%	0.06%	0.02%	0.05%	0.12%	0.19%	0.24%

Table 7: The spatiotemporal method's RMSEs for the Metropolitan and Rural counties

County	1	2	3	4	5	6	7	8
Metropolitan RMSE	4.01	6.97	13.04	20.45	28.99	38.51	47.03	54.04
Rural RMSE	18.68	24.97	31.87	40.05	47.50	54.13	59.87	66.98
Absolute Difference of RMSEs	14.68	18.00	18.83	19.60	18.52	15.62	12.84	12.94
Average Number of Cases	6603.74	6627.28	6652.34	6677.43	6701.85	6725.55	6748.49	6771.51
Percentage Difference	0.22%	0.27%	0.28%	0.29%	0.28%	0.23%	0.19%	0.19%
County	9	10	11	12	13	14	15	16
Metropolitan RMSE	60.11	66.30	75.22	85.54	100.28	117.22	135.66	153.01
Rural RMSE	73.24	78.60	85.08	93.76	103.15	113.48	122.41	131.30
Absolute Difference of RMSEs	13.13	12.30	9.86	8.22	2.87	3.74	13.25	21.71
Average Number of Cases	6793.62	6814.36	6835.62	6858.13	6881.07	6904.41	6927.87	6950.04
Percentage Difference	0.19%	0.18%	0.14%	0.12%	0.04%	0.05%	0.19%	0.31%

the lower and upper bound values of 30 and 280. These values may be dependent on the characteristics of the raw data in each state. It would be good to build a program that generation different set of upper bounds and lower bounds and to automatically find the best set of upper bound and lower bound that produce the lowest RMSE and MAE.

In addition, if scientists find a vaccine against the Covid-19 virus, then a more complex model could be developed that takes into consideration the percentage of the population that was vaccinated or already had the disease and developed some immunity to it.

A further complication that the mutation of the Covid-19 virus with the introduction of new strains. The newer strains may be more virulent and deadly than the original strain of the Covid-19 virus. Moreover, people who have been vaccinated for the original strain may not have an immunity for the emerging strain. The degree of immunity against the new strains is vaccine dependent, that is, some of the currently available vaccines by Pfizer, Moderna, Johnson & Johnson provide varied degrees of immunity against the new strains. On the other hand, the pharmaceutical companies are expected to develop newer vaccines against Covid-19. Many epidemiologists believe that the Covid-19 pandemic could be a recurring pandemic that would need a new vaccination each year similar to the common flu.

While the consideration of these extra statistics and complications could refine each of the prediction methods, we expect the improvements to be about the same percentage for all of the prediction methods. Hence our main conclusion that spatiotemporal extrapolation yields the most accurate prediction method will likely continue to hold with these extension too. We hope that the main ideas behind our spatiotemporal prediction method will be adopted by epidemiologists in their work.

References

[1] C. Huang, Y Wang, X. Li, L. Ren, J. Zhao, Y. Hu, L. Zhang, G. Fan, J. Xu, X. Gu, *et al.* Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *The Lancet* **2020**, *395*, 497–506.

[2] P. Z. Revesz, and S. Wu, Spatiotemporal reasoning about epidemiological data. *Artificial Intelligent Medicine* **2006**, *38*, 157–170. doi:black10.1016/j.artmed.2006.05.001.

[3] L. Li, and P. Z. Revesz, Interpolation methods for spatio-temporal geographic data. *Computers, Environment and Urban Systems* **2004**, *28*, 201–227. doi:black10.1016/S0198-9715(03)00018-8.

[4] Y. Liu, Z. He, and X. Zhou, Space-Time Variation and Spatial Differentiation of COVID-19 Confirmed Cases in Hubei Province Based on Extended GWR. *ISPRS Int. J. Geo Inf.* **2020**, *9*, 536.

[5] V. Thakar, Unfolding Events in Space and Time: Geospatial Insights into COVID-19 Diffusion in

Washington State during the Initial Stage of the Outbreak. *ISPRS Int. J. Geo Inf.* **2020**, *9*, 382.

[6] Z. Wang, M. Yao, C. Meng, and C. Claramunt, Risk Assessment of the Overseas Imported COVID-19 of Ocean-Going Ships Based on AIS and Infection Data. *ISPRS Int. J. Geo Inf.* **2020**, *9*, 351.

[7] L. J. Thomas, P. Huang, F. Yin, X. I. Luo, Z. W. Almquist, J. R. Hipp, and C. T. Butts, Spatial heterogeneity can lead to substantial local variations in COVID-19 timing and severity. *Proceedings of the National Academy of Sciences* **2020**, *117*, 24180–24187,

[8] R. L. Burden, J. D. Faires, and A. C. Reynolds, *Numerical Analysis*, 2001.

[9] J. Gao, and P. Z. Revesz, Voting prediction using new spatiotemporal interpolation methods. Proceedings of the 7th Annual International Conference on Digital Government Research, DG.O 2006, San Diego, California, USA, May 21-24, 2006; Fortes, J.A.B.; Macintosh, A., Eds. Digital Government Research Center, 2006, Vol. 151, *ACM International Conference Proceeding Series*, pp. 293–300. doi:black10.1145/1146598.1146678.

[10] P. Z. Revesz, Data mining citations to predict emerging scientific leaders and citation curves. *International Journal of Education and Information Technologies* **2017**, *11*, 171–179.

[11] D. Shepard, A two-dimensional interpolation function for irregularly-spaced data. Proceedings of the 23rd ACM National Conference; ACM Press: New York, NY, USA, 1968; p. 517?524.

[12] P. Z. Revesz, *Introduction to Databases: From Biological to Spatio-Temporal*; Springer, 2010.

[13] D. Krige, A statistical approach to some mine valuations and allied problems at the Witwatersrand. University of Witwatersrand. Master's thesis, 1951.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning representations by back-propagating errors. In *Neurocomputing: Foundations of Research*; MIT Press: Cambridge, MA, USA, 1988; p. 696?699.

[15] F. Gers, F. Long short-term memory in recurrent neural networks. PhD thesis, University of Hanover, 2001.

[16] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Transactions on Smart Grid* **2019**, *10*, 841–851.

[17] C. J. Huang, and P. H. Kuo, A deep CNN-LSTM model for particulate matter (PM2. 5) forecasting in smart cities. *Sensors* **2018**, *18*, 2220.

[18] A. Sagheer, and M. Kotb, Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing* **2019**, *323*, 203–213.

[19] World Population Review. *Population of Counties in New York (2020)*, 2020.

[20] United States Census Bureau. *Gazetteer Files*, 2019.

[21] M. Smith, K. Yourish, S. Almukhtar, *et al.* *Coronavirus (COVID-19) Data in the United States, The New York Times.*, 2020.

[22] MathWorks. *Ordinary Kriging*, 2020.

[23] MathWorks. *Time Series Forecasting Using Deep Learning*, 2020.

[24] A. Srinivas, *machine-learning-made-easy/backpropogation.py*, 2020.

[25] J. Cai, and P. Z. Revesz, A novel spatiotemporal interpolation algorithm and its application to the COVID-19 pandemic. Proceedings of the 24th Symposium on International Database Engineering & Applications, 2020, ACM Press, pp. 1–10. doi:blackhttps://doi.org/10.1145/3410566.3410602.

[26] H. Ward, C. Atchison, M. Whitaker, K. E. Ainslie, J. Elliott, L. Okell, R. Redd, D. Ashby, C. A. Donnelly, and W. Barclay, SARS-CoV-2 antibody prevalence in England following the first peak of the pandemic. *Nature communications* **2021**, *12*, 1-8.

Peter Z. Revesz holds a Ph.D. degree in computer science from Brown University, Providence, Rhode Island, USA in 1991.

He was a postdoctoral fellow at the University of Toronto before joining the University of Nebraska-Lincoln, where he is a Professor in the Department of Computer Science and Engineering.

Dr. Revesz is an expert in databases, data mining, big data analytics and bioinformatics. He is the author of the textbooks *Introduction to Databases: From Biological to Spatio-Temporal* (Springer, 2010) and *Introduction to Constraint Databases* (Springer, 2002). Dr. Revesz held visiting appointments at the IBM T. J. Watson Research Center, INRIA, the Max Planck Institute for Computer Science, the University of Athens, the University of Hasselt, the U.S. Air Force Office of Scientific Research and the U.S. Department of State.

Dr. Revesz is a recipient of an AAAS Science and Technology Policy Fellowship, a J. William Fulbright Scholarship, an Alexander von Humboldt Research Fellowship, a Jefferson Science Fellowship, a National Science Foundation CAREER award.

Junzhe Chen is a graduate student in the Department of Computer Science and Engineering at the University of Nebraska-Lincoln.

Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

Dr. Revesz designed the study, proved Theorem 1 and wrote most of the paper. Junzhe Chen carried out the experiments, drew the figures and contributed to writing about the algorithms.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)