COVID-19 Prediction: Models and Their Evaluations

Mingyang Zhang
zhangbruin24@g.ucla.edu
405170429

Rui Deng
ruideng17@gmail.com
205123245

Qingyuan Pan
qingyuan2018pan@sina.com
405101723

Yuxin Wang
yuxinwanghailey@outlook.com
905129084

1 ABSTRACT

During the last few months, COVID-19 prediction has become a popular topic, as people are in an urgent need to know when could mankind stop suffering from the pandemic. In this paper, we illustrate our effort to use different models, including polynomial regression, SIR, LSTM, and ARIMA, to approach this problem. We have experimented with these models to predict the confirmed cases and death cases in the US with a given COVID-19 dataset. We first introduce the background of our problem and its formulation. We then explain our approach to solve the defined problem, where the method explanation contains the overall architecture, model research and selection, and model tuning. We go into the details about the model we chose and how it works. Next, we discuss the performance of our approach with its advantages, reasoning, and limitations. We lastly adopt an ensemble of the linear and polynomial regression models for our final prediction, but we also offer insights on how other models could be applied under different scenarios.

2 INTRODUCTION AND BACKGROUND

With the massive outbreak of COVID-19, the prediction of COVID-19 data has become a popular research topic since March 2020. Presently, the global trend of the coronavirus is still unclear and ever-changing; therefore, it brings difficulties to predict the trend of the COVID-19 accurately. In many aspects, a reliable prediction is important to pandemic control and prevention. Firstly, experts and specialists can make helpful suggestions for disease control based on effective predictions. Secondly, the prediction of COVID-19 is a valuable statistic to assess whether policies used to mitigate the spread of disease are successful.

Researchers all over the world have adopted a large number of different methods to approach this problem. Comparing to traditional epidemiological models like the SIR model, the modern machine learning model provides another insight into this problem. With the help of abundant past confirmed data, machine learning models are able to capture the underlying patterns of data in more complicated scenarios, without the need for explicitly defining and tuning the parameters for pandemic spreading before the prediction.

Our goal in this paper is to perform a prediction task on the given COVID-19 dataset and to obtain the best results for confirmed and death cases determined by the given performance indicators. To achieve this purpose, we have investigated the possible candidate models that are suitable to the nature of this project – a prediction task based on a time series dataset. Some other important procedures, including adopting a proper method to preprocess the dataset, model tuning, and result interpretation, will also be illustrated in the following sections.

For the model selection, we used four different models to predict the confirmed and death cases for each state in the US. Specifically, we primarily base our prediction on polynomial regression, susceptible-infected-recovered (SIR) model, long-short-term memory (LSTM) model, and Auto Regression Integrated Moving Average (ARIMA) model. Due to the different complexity, mechanisms, and assumptions inherent to these models, they yield different results. In order to obtain the best performance, we investigate the feasibility and capability of these models in terms of COVID-19 prediction. A detailed explanation of each model's mechanism and implementation will be introduced in later sections.

3 RELATED WORK

The COVID-19 outbreak is related and similar to the spread of SARS in 2003. They are both caused by corona-viruses that trigger respiratory diseases. The genome of SARS-CoV-2, which is the virus responsible for the COVID-19, and SARS-CoV, which is the virus in the 2003 SARS epidemic, share a genetic similarity of 86 percent (Ni, W.). It is natural to look at the models and methods people took from 17 years ago to fight with SARS. For example, researchers in 2003 have already tried to use the SIR model to predict the SARS' spreading during the SARS pandemic. The spreading of these two person-to-person transmittable diseases both follows the SIR model, which indicates that if we obtain the characteristics of SARS 17 years ago, it might be possible that these two time-series data could be mapped, with algorithms like dynamic Time wrapping (Smith. A. et al). However, the SARS and COVID-19 are still different in severity, extensiveness, and attitude from the society. Therefore, we must try multiple models or combined models to help us in predicting COVID-19.

In another research (Tomara. A. et al), a data-driven forecasting method focusing on LSTM and curve fitting was used to estimate the number of positive cases of COVID-19 in India for a period of 30 days. From their fitting result, it showed that number of positive cases of COVID-19 in India followed a power rule $f(x) = ax^b$. But because of the different government policies and country situation, the trend of spread of COVID-19 in the US is different from that in India. The predicted power rule is improper for measuring COVID-19 in the US.

4 PROBLEM DEFINITION AND FORMALIZATION

The breakout of the COVID-19 is a serious global issue and caused a grief loss of lives across countries. As a reaction to this global pandemic, researchers tried different approaches to predict the trend of COVID-19 cases given the figure of confirmed cases, death cases, and many other important indicators like hospitality rate and recovery rate. Concretely, we formalize our COVID-19 prediction problem as:

Given the 50 states $S = \{S_1, S_2, ..., S_{50}\}$ in America, sort the state name in alphabetical order and arrange them by this order throughout the prediction, i.e. $S_1 = Alabama, S_2 = Alaska, ...S_{50} = Wyoming$. Given the series of cumulative confirmed and death cases in total d days, denoted by $\{TC_{tS_n}, t = [1, d]\}$ and $\{TD_{tS_n}, t = [1, d]\}$, we want to predict the subsequent series of confirmed and death cases $\{TC_{(tS_n}, t = [d + 1, m]\}$ and $\{TD_{tS_n}, t = [d + 1, m]\}$, where m is the final date that we want to predict.

From the definition above, we can see that this problem is a typical supervised prediction problem with time-series data. Moreover, the given time-series is related to transmittable diseases. These insights are important to help us pick the proper algorithms and approaches during the model formalization, implementation, and evaluation.

With the ultimate evaluation metrics defined as MAPE, we looked at the dataset and perform preprocessing on it to obtain the clean and trainable data. Graphing the shape and trend of the data is also helpful in terms of the model selection. Thenbased on pre-screening of the data, we researched and implemented some potential candidate models, from which we could pick up the best fit. After the background investigation and a rough sanity test for each model, we devoted ourselves to the tuning-evaluation loop. We keep tuning our best fit models according to the feedback of the MAPE score. Lastly, we also pay special attention to the prevention of overfitting, where cross validation was taken to help with hyperparameter selection and overfitting prevention.

5 DATA PREPARATION AND PREPROCESSING

Examining the provided data is crucial for model formulation and assessment. The dataset contains the COVID-19 data from the 50 states of the USA, starting on April 12th and ending on November 22nd, 2020. We also downloaded the confirmed and death cases until December 5th to lower the bias and errors in round-2 prediction. With the cases continuously increasing during this time and potentially keeping increasing in the future, it is useful to take a further look at the detail of the dataset.

Firstly, we looked at the training and testing samples. Available features include numbers of confirmed cases, death cases, recovered cases, active cases, tested people, hospitalized people, and the incident rate, where confirmed and death cases are our prediction targets, or the labels. One nice characteristic of this dataset is that all of our features are numerical, which will make it easier to find potential approaches to produce this time series. Meanwhile, we could also focus on finding the models that work better for numerical data instead of categorical data. The following table is a detailed description of each feature:

Province_State	The name of the State within the USA.
Date	The date of the corresponding data, which is in the format of mm-dd-yyyy.
Confirmed	Aggregated case count for the state.
Deaths	Aggregated death toll for the state.
Recovered	Aggregated recovered case for the state.
Active	Aggregated confirmed cases that have not been resolved.
Incident_Rate	Cases per 100,000 persons.
People_tested	Total number of people tested.
People_Hospitalized	Total number of people hospitalized.

We then perform data preprocessing on this dataset. We extract the confirmed cases and deaths for each state separately and store them into subset of data for later training process. We then notice that some fields are empty (i.e. NaN) in our data frame, so we intend to fill those entries with reasonable values. For the 'Recovered' column, we fill the NaN value with 0, because we do not want to overestimate the number of recovered people. For the column of people_hospitalized and Hospitalization_Rate, we fill the empty entries with values from its previous rows, since we assume no new patient is hospitalized given an empty entry.

Based on our visualization of the COVID-19 trend in different states, we find out that in some states, like California and Nevada, the number of confirmed cases grows exponentially, while in others, such as Texas, the speed of increasing in confirmed cases is gradually converging to a more steady rate. It is reasonable that the COVID-19 situation is different in each state, as they differ from each other geographically and culturally. Due to such a difference in the trends of COVID-19 across the country, we decide to create each individual model for each different state.

6 METHODS

After examining the data and visualizing the trends, we observed that the plots of the confirmed cases and the death cases follow regular polynomial patterns, so a polynomial regression model could be an appropriate start to predict COVID-19 in the US. Then, given the scenario of this problem as predicting the spread of disease, our second choice is the classic Susceptible, Infected, and Recovered (SIR) model, since it can properly model the trajectory of an epidemic given its set of differential equations. We then notice, however, fitting the parameters in the differential equation could be a costly and demanding process, and the gap between assumptions in this model and the real-life scenarios could lead to great inaccuracies. Then, we further observe that the data for COVID-19 is temporal in nature. Such time-series data contains some repetitive patterns influenced by government control policies, such as mandatory social distancing, lockdown, reopen, and so on. Therefore, we consider the LSTM model as a good fit for time-series prediction, since its complex neural network structure gives it strong power to learn the long term dependencies and inherent patterns in our

dataset. Another powerful tool we utilized is the ARIMA model, as the COVID-19 data is categorized as time-series data, whose information changes between the previous dates can be captured to predict the future values.

6.1 Polynomial Regression

Before implementing the SIR and LSTM, we firstly used polynomial regression to generate a baseline for our prediction. Even though the limited degree and dimension of polynomial regression restrict its ability to discover complex insights in the dataset, it can still provide a robust interpolation for the trend of COVID-19 in the U.S., since the cumulative cases grow monotonically throughout our prediction interval.

6.1.1 Regression Model. We define a degree N polynomial from $\mathbb{R}$ to $\mathbb{R}$ as $f(x) = b_n x^n + b_{n-1}x^{n-1} + ... + b_2 x^2 + b_1 x + b_0$, where $b_0, b_1, ...b_n$ are real constants. Given the time series (for confirmed cases or death cases) $\{T_{nS_n}, n = [1, d]\}$, each day's date n and its corresponding case C_n is a data point. Our goal is to minimize the mean squared error between our prediction and the ground truth value, so the loss function could be formalized as:

$$J = \frac{1}{d}\sum_n (C_n - (b_n x^n + b_{n-1}x^{n-1} + ... + b_2 x^2 + b_1 x + b_0))^2 \quad (1)$$

Since this loss function is convex, we can use gradient descent to converge to an optimal solution.

6.1.2 Degree Selection. When modeling polynomial regression, the most important feature is the degree of the polynomial. The degree will affect the variance and bias of the model: a low degree will give a higher bias, whereas a high degree leads to overfitting. During training, we notice that polynomials with degree 3 and above are unstable in the long-term, since those polynomials may have an unruly turning point, making the cumulative cases decrease abruptly as time goes on. Then, we decide that quadratic and linear functions are the best fit for the trend of COVID-19 in the US.

6.1.3 Interval Selection. It is perfectly possible, however, even with the best degree selected, polynomial predictions diverge more and more from the true data as time goes on. This is because polynomials are not flexible and sensitive enough to represent the peak and later flattening trend of this epidemic. As we mentioned before, the trend of COVID-19 is fast-changing because of a wealth of policies, and we propose that it is thus more appropriate to predict the future trends only based on the most recent 40 days of the provided time-series. By plotting the trends of those 40 days, we can see that they generally follow a quadratic growth pattern, so we use a quadratic polynomial to fit these data. Moreover, we also adopted the data from the most recent week and used them to adjust the performance of our quadratic polynomial, since the data from the last 7 days is the best predictor for the near future's growth. The details associated with the degree and interval selection for each state will be explained in the later model implementation section.

Although we expected that the polynomial model is only a baseline for our prediction, we still used it for the final round's output. This is because comparing to other models, the polynomial regression fits the data in a clean and explainable way, and it is stable in the long term and resistant to noises in the data. Also, the cases in the U.S. grows monotonically with little disturbances at the beginning of December, so the polynomial is a robust predictor for the cases in the near future. Still, we admit the limited power of polynomials, and we, therefore, implemented other potential models for disease forecasting.

6.2 SIR

6.2.1 Model Selection. Common infectious disease models are divided into SI, SIR, SIRS, and SEIR models according to the types of infectious diseases. Although SEIR takes the incubation period of COVID-19 into account and is therefore a more proper model, we still chose the SIR for future prediction. The most prominent reason is that we do not know what fraction of people are exposed in each state and under different policies, so adopting the exposed criteria will add more inaccuracy and fluctuations to our model. Moreover, given that people are aware of the dominant symptom of COVID-19, we assume that people who are infected can be diagnosed in a short period and stay self-quarantined. Thus, SIR could also be an appropriate choice to forecast future cases of COVID-19.

6.2.2 Parameters. S, I, and R represent the following groups of people respectively:
S: Susceptible means that some people are never infected by the corona virus, but are likely to be infected by COVID-19 as the disease spreads through the population. In our model, we set $S_0 = P_{state}(population) - Infected_{t=1}$, where t = 1 indicates April 12nd, 2020 in our dataset.
I: Infected is the number of people who are infected by the disease and might transmit the disease to a member of S, thereby becoming a new member of Category I. Here, we set $I_0 = Infected_{t=1}$.
R: Recovered, referring to number of people who have been recovered and immune from illness and are thus not able to infect others. Moreover, we also want to count the proportion of death in R, so we add an extra parameter denoting the death rate to the differential equation. In our model, we set $R_0 = Recovered_{t=1}$.
D: Death, referring to number of death cases. Although this parameter is not essential to SIR, we add it here to count the proportion of death in population. Here, $D_0 = Death_{t=1}$.

6.2.3 SIR Model. The model simulates the transmission path of infectious diseases and predicts the spread scale and time of infectious diseases from the Susceptible to the Infected to the Recovered based on the following differential equations:

$$\frac{dS}{dt} = -\beta I \frac{S}{N} \quad (2)$$

$$\frac{dI}{dt} = \beta I \frac{S}{N} - \gamma I - dI \quad (3)$$

$$\frac{dR}{dt} = \gamma I \quad (4)$$

$$\frac{dD}{dt} = dI \quad (5)$$

and the iteration form will be:

$$S_n = s_{n-1} - \beta I_{n-1}\frac{S_{n-1}}{N} \quad (6)$$

$$I_n = I_{n-1} + \beta I_{n-1}\frac{S_{n-1}}{N} - \gamma I_{n-1} - dI_{n-1} \tag{7}$$

$$R_n = R_{n-1} + \gamma I_{n-1} \tag{8}$$

$$D_n = D_{n-1} + dI_{n-1} \tag{9}$$

where N, β, γ, d represent the total population, the probability of being infected, and the probability of recovery, and the probability of death.

6.2.4 Parameter Fitting. We optimized the above parameters in differential equation by minimizing the mean squared error between each prediction and the true value. Let $S_{pred}, R_{pred}, D_{pred} \in \mathbb{R}^d$ denote the prediction based on SIR, and let $S_{true}, R_{true}, D_{true} \in \mathbb{R}^d$ denote the confirmed cases, recovered people, and death cases given in the dataset. We want to minimize:

$$J = \left\|S_{pred} - S_{true}\right\|^2 + \left\|R_{pred} - R_{true}\right\|^2 + \left\|D_{pred} - D_{true}\right\|^2 \tag{10}$$

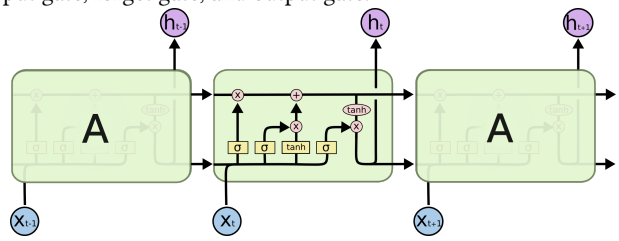
This is a convex loss function, so we can use gradient descent to reach an optimal solution. Still, no matter how well the parameters we fit, the shape of SIR is determined by its differential equations and thus cannot exactly fit patterns of actual cases in the US. Thus, we know that SIR is suitable when we visualize the overall trend of a pandemic, but it is less useful when we want to fit the number of cases exactly in a disease outbreak.

6.3 LSTM

6.3.1 Model Selection. Under the circumstances of real-world data prediction, many theoretical models fail to fit the pandemic data nicely and give poor accuracy. This is because the process of disease transmission is more complicated than the theoretical mathematical models under the perfect assumption. Besides the natural infection and transmission in population, many countries have taken measures to prevent the spread of COVID-19, such as lockdown and restricting transportation. The population movement between states and even countries further increases the complexity and variation in data. All such factors will make traditional statistical and mathematical methods less powerful in terms of COVID-19 prediction. Therefore, more researchers are seeking the help of new technologies, such as machine learning and deep learning, which are more robust at handling complicated and ever-changing real-world scenarios.

Deep learning models such as RNN are widely used in time-series prediction. This kind of model could effectively extract relevant features and feed the activation from the previous time step as input for the current time step. However, one possible disadvantage of RNN is that it suffers from a vanishing/exploding gradient problem during the back-propagation. To solve this problem, LSTM was introduced in 1997 by Hochreiter and Schmidt. LSTM regulates the information stored or deleted in the cell state through well-designed structures called gates and thus learns order dependency in a given time series. In terms of pandemic prediction, LSTM has become a popular choice for many researchers.

6.3.2 LSTM Model. Each LSTM neural unit consists of cell state, input gate, forget gate, and output gate.



First, the LSTM decides what the network needs to discard, and the main function of the forget gate is to determine the current state of the information that are no longer used and need to be forgotten.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{11}$$

The input gate copies the input information of the current neural unit. This function is used to determine what new input information will be updated and stored in the container.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{12}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{13}$$

Then, the cell state can update itself given the information calculated in the forget gate and input gate.

$$C_t = C_{t-1} * f(t) + i_t * \tilde{C}_t \tag{14}$$

Output gate filters information and decides what parts of the cell state will become the output.

$$o_t = \sigma(W_o[h_{t-1}, xt] + b_o) \tag{15}$$

$$h_t = o_t * tanh(C_t) \tag{16}$$

6.3.3 Model Implementation. For the input, we used the data of the last 3 days to predict the data for the next day. We chose the mean absolute percentage error (MAPE) as the loss function. That is to say, we want to minimize the MAPE between our predicted value through LSTM and the true cases in the US.

6.4 ARIMA

6.4.1 Model Selection. Auto Regression Integrated Moving Average, abbreviated as ARIMA, is generally used to forecast the future trend using the existing time series data. Specifically, with existing time-series data, we could capture its lags and the lagged forecast errors to model the underlying equation of the time series, and therefore forecasting its future values. For COVID-19 prediction, ARIMA fits the intrinsic nature of time-series prediction in our task.

6.4.2 ARIMA Model. By convention, the three parameters are named p, d, q, where p is the number of the autoregression terms, d is the number of the nonseasonal differences needed for stationary and q is the number of lagged forecast errors in the prediction equation. The value of p and d will be used to form the underlying equation of the ARIMA model, as it decided how the later data is determined from the previous times. Here, p denotes the number of lags in the model, when Yt only depends on the number of p lags, the model is a pure Auto Regressive model:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \ldots + \beta_p Y_{t-p} + \epsilon_1 \tag{17}$$

The parameter q determines the lagged forecast errors of the model, when Yt only depends on the number of lagged forecast errors, the model is called a pure Moving Average model:

$$Y_t = \alpha + \phi_1\epsilon_{t-1} + \phi_2\epsilon_{t-2} + ... + \phi_q\epsilon_{t-q} \tag{18}$$

Note that the ϵ_t denotes the error from the auto-regressive models of the respective lags at t's term. Then the ARIMA model is just a prediction of Yt given by the linear combination p-lags of Y and q-Lagged forecast errors, along with constant α. Finally the ARIMA model becomes:

$$Y_t = \alpha+\beta_1 Y_{t-1}+\beta_2 Y_{t-2}+...+\beta_p Y_{t-p}+\epsilon_1+\phi_1\epsilon_{t-1}+\phi_2\epsilon_{t-2}+...+\phi_q\epsilon_{t-q} \tag{19}$$

Another parameter d is used to determine the number of differencing needed to make the time series stationary. This term is important to the ARIMA model, as the ARIMA model is a linear model that uses its lags as predictors, which need to be independent of each other in order to achieve the best performance. By obtaining the stationary time series, we could ensure the independence of the predictors. A non-seasonal time series could be made stationary by the differencing process, and d will take an important role in this process by deciding to what extent should we differencing the data. If the d is too small the time series may not be stationary, and if the d is too large, it will also have negative effects on other parameters. (S. Prabhakaran)

7 EXPERIMENTS DESIGN AND EVALUATION

In this section, we will discuss how we evaluated the performances of our models. As we have discussed in the previous section, we choose mean absolute percentage error (MAPE) as the performance measuring metric, mainly due to the project requirement. However, we do agree with the fact that MAPE, with great scale-independency and interpretability, is a reasonable metric used to evaluate the performance of prediction tasks.

Mean absolute percentage error, short for MAPE, is a commonly used metric to measure the accuracy of a trend prediction. MAPE is the average of all the absolute percentage errors:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{Y_i - \hat{Y}_i}{Y_i}| * 100\% \tag{20}$$

It measures how much is the prediction off from the ground truth, measured in percentage. Comparing to other popular metrics such as MSE and MAE, MAPE allows us to reduce the impact brought by outliers and large value differences of data points. If the difference between each label is huge, or if we will have some large outliers, then the error measurement, like MAE or MSE, may run into the problem that for some large value, the prediction error will have a larger impact on the performance measurement comparing to the smaller value. MAPE will balance this impact by dividing each error by the quantity of the data point, therefore each data point will have an equal effect on the error measurement. However, the disadvantage of MAPE also comes from this mechanism. If the data point is too small, then dividing by the data point will enlarge its impact on the measurement.

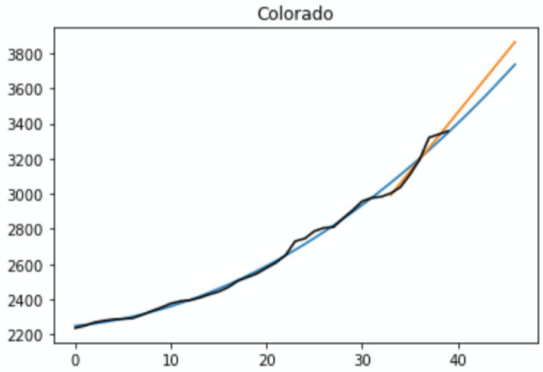
7.1 MODEL IMPLEMENTATION

7.1.1 Polynomial regression. For the implementation of our polynomial regression model, we have utilized PolynomialFeatures and LinearRegression from the sklearn package of python. We first created a new feature matrix that contains the polynomial combinations of the preprocessed features (confirmed cases and death cases) with a specific degree, and the logistic of how we choose these degrees is discussed in the following paragraph. After obtaining this transformed training dataset, we train a linear regression model with this dataset and follow this process we will produce the desired polynomial regression models.

One key point to notice is that the prediction will be affected by our choice of the training dataset, and it is not always the best solution to use as much data as possible. As the limitation of polynomial regression is that we could only fit the given data into one polynomial model, it is important to be careful about what subset of data will be used. Meanwhile, we will create new models trained with data only from one state for each different state. Because the COVID-19 trend in each state is different due to the geographical, cultural, and political differences of each state. For each state, we will also train two individual models for confirmed case prediction and death case prediction separately, as the death cases are not only correlated to the confirmed cases but also affected by other influencers such as population and medical resources.

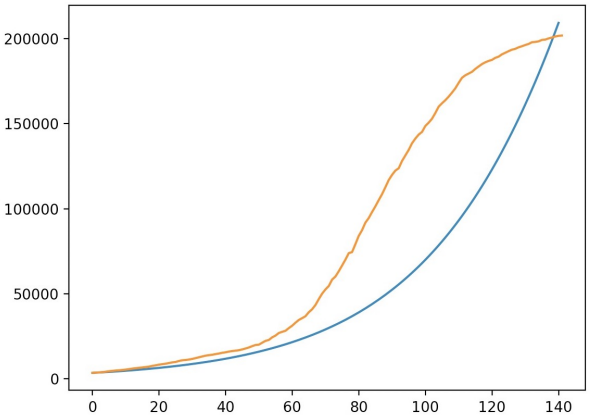
In this paragraph, we will illustrate our logistics of deciding the two important factors of the model, which are the degree of the polynomial model and the duration of the training data, i.e. how many days to use to train the polynomial model for each state. The degree parameter defines the shape of our model: a higher degree indicates a more complex shape with potentially higher variance. By visualizing the trend of confirmed cases and death cases for each state, it is obvious that most states have trends that follow some lower degree polynomials. After testing multiple possibly reasonable candidates of degree, we chose to use the ensemble of degree 1 and degree 2, which fitted most features and future trends for each state. As for the duration of the training data, we knew that the growth of COVID-19 cases might not follow one polynomial model constantly for a long period of time. But because of the government enforcement methods such as lockdown, curfew, and reopen highly influencing the trend of COVID-19, a mixture of several individual models with different degrees in which each model represented one round of breakout starting from different dates outperformed a single model for most of the states. Generally, we employed a combination of a linear model for the past 7 days and a degree 2 polynomial model for the past 40 days as the model for most of the states where the trends were increasing consistently. However, in states such as Hawaii and Alaska, the trends showed greater linearity in the near past days, we then employed a combined linear model of the past 5 days and 7 days. In conclusion, our final model for a mixture of the linear and polynomial model gives an overall MAPE score of 1.97045 for round1 competition on Kaggle.

7.1.2 SIR. As introduced before, we have implemented a relatively simple SIR model, with the population divided into the susceptible,



The above image shows a combination model of a linear model for past 7 days (orange line) and a degree 2 polynomial model for past 40 days (blue line) in Colorado; the black line represents the true death cases

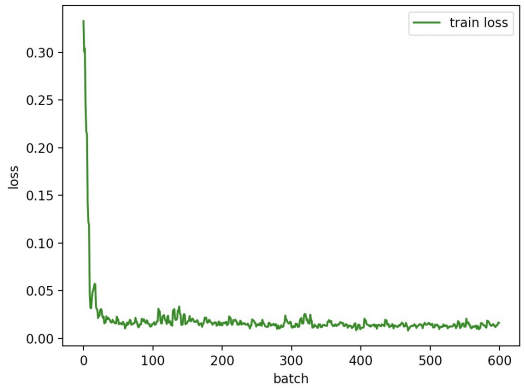
the infected, and the recovered. We also made an ideal assumption that the recovered would not be infected again. Moreover, in this model, we assumed that the number of the total population in a state would not change throughout the pandemic, and the spreading of the epidemic would not be affected by other real-world factors but only by the classic influencers such as the death rate, infection rate, and recovery rate. We used the minimizing function from scipy.optimize library to optimize the loss function (10). This function allows us to use the existing data and the initial value hyper-parameters as input, and it iteratively optimizes the parameters according to the given loss function and the range of parameters. Upon obtaining the best configures, we could plug them back into the SIR Model along with the initial values for the SIR, the number of the total population, and the prediction interval to generate our forecast. With this training and predicting process, we could obtain the desired prediction.



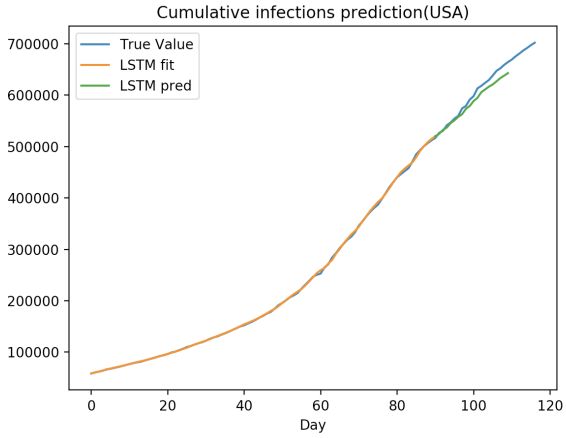
The above image shows a simple prediction generated by our SIR model for Arizona state, where the x-axis indicates the re-indexed dates from the given dataset, and the y-axis represents the accumulative confirmed cases. The orange line indicates the predicted values by the SIR model, and the blue line represents the true cases.

Under a naive assumption, the shape of confirmed cases in SIR is similar to logistic growth and is determined by its differential equation. However, unlike the ideal model, in the real world situation, the total population for each state is not fixed and the stages of a pandemic cannot simply be divided into four differential equations. This explains the gap and difference in shapes between our prediction and the true value. In conclusion, the SIR model is a powerful tool, but in order to generate more realistic prediction results, we need to consider more real world conditions and modify the model accordingly.

7.1.3 LSTM. We implement our LSTM model based on Keras, an open-source neuron network library of python. We tested our model with data of California state as an experiment. The dataset was split into 25% of testing data and 75% of training data. In terms of the LSTM model setup, we used one LSTM layer with 100 neurons and a 0.15 dropout rate, the default tanh activation function, and MAPE as the loss function. We trained the model with 200 iterations.



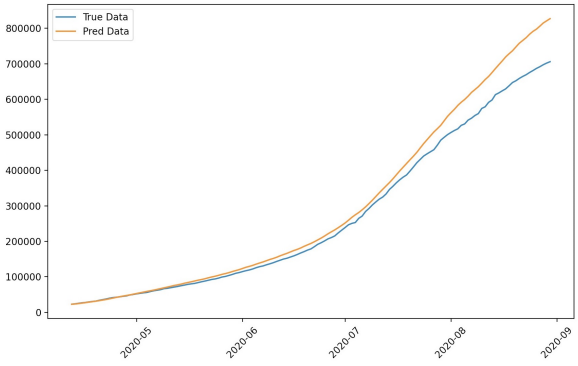
The above figure shows the training loss corresponding to each training iteration, as we could see that the model converge to a steady state, while the training loss stabilized around 0.05 after the 30 iterations.



The above figure shows a training and testing outcome for the cumulative confirmed cases in California, and the prediction is

relatively accurate. However, in practice, we notice that the performance of LSTM is not stable and can be easily affected by noises in the training and testing data. Since we are only given around 200 days of confirmed and death cases, the limited data make LSTM fail to generate a rigorous and robust prediction for individual states in the US.

7.1.4 ARIMA. For our ARIMA model, it is important to determine the proper parameters(p, d, q). For the d term, we have tested a series of potential order of d. By visualizing the result of a different order of differencing, we have found that the data could be made stationary without over differencing when d = 1. For p and q term, we utilized the PACF and ACF plot to help us decide the best fit. By testing a range of the possible number of lags and inspecting the Partial Autocorrelation plot (PACF), we can observe the correlation between the number of lags and its significance. We will choose the smallest number of lags that gives a decent significance, which will be 3 in our case. Similarly, by inspecting the Autocorrelation plot (ACF) we decided that 3 will be a proper number for q in our ARIMA model. After the p d q is determined, we can build our model by utilizing the ARIMA model from the statsmodels library by Python. Passing in the given COVID-19 data and proper p d q value, we can obtain our prediction. Following are the prediction results of our ARIMA model for the number of confirmed cases in California states over time. We could see that the ARIMA model is capable of capturing the general trend of the confirmed cases over time.



Still, the gap between the prediction and real-cases becomes greater as time goes on, since ARIMA accumulates the error when it simply simulates the linear regression and uses previous inaccurate predictions to forecast the future. Thus, we need more tuning and re-weighting or even adjusting the architecture of the algorithm if we want to adopt ARIMA in our prediction task.

7.1.5 Summary of results. By comparing the plotted prediction results and the MAPE generated from all the above models, we have decided that the polynomial regression model achieves the best prediction performance. We summarized two main reasons behind this result. Firstly, the ground truth spreading trend of COVID-19 in the US is increasing smoothly, following a pattern of the polynomial functions. Another reason is that in our model implementation, ARIMA and SIR models made a lot of ideal assumptions about the dataset, which did not apply to the real-world situation and therefore hindered the prediction performance. Also, LSTM would require more effort to fine-tuning the model than the polynomial regression model; therefore, its ability is limited when the dataset is relatively light-weighted. Especially in our prediction task, the number of data and the selectable features are few, so LSTM cannot effectively learn the underlying patterns of our time series.

8 CONCLUSION

In this project, we tried mathematical, machine learning, and algorithm-wise approaches to predict the confirmed and death cases given the COVID-19 dataset. With a relatively small-sized dataset and limited feature selection, the ensembled model of linear and 2-degree polynomial regression achieved the best result with a MAPE of 1.97. Still, this does not imply that polynomial regression is the best model for predicting COVID-19 as a whole. It is only a possibility that the COVID-19 trend may follow the polynomial prediction if the current situation does not change: the factors that could affect the epidemic, such as quarantine policy and local temperature, would not change. To obtain a more robust prediction in the long term, we recommend using the LSTM model because of its superior ability to learn and simulate complex trends in COVID-19. In the future, we believe that fine-tuning the model, introducing more features as input, and making a more realistic assumption will all contribute to a better solution to our prediction problem.

9 TASK DISTRIBUTION FORM

Task	People
Preprocessing data and choosing models	All members
Implementing models and generating predictions	Mingyang Zhang , Rui Deng
Evaluating results, and writing report	Qingyuan Pan, Yuxin Wang, Rui Deng, Mingyang Zhang

REFERENCES

[1] Alazab, M., Awajan, A., Mesleh, A., Abraham, A., Jatana, V., Alhyari, S. (2020). COVID-19 prediction and detection using deep learning. International Journal of Computer Information Systems and Industrial Management Applications, 12, 168-181.
[2] Annelies Wilder-Smith, Calvin J Chiew, Vernon J Lee, Can we contain the COVID-19 outbreak with the same measures as for SARS?, The Lancet Infectious Diseases, Volume 20, Issue 5, 2020, Pages e102-e107, ISSN 1473-3099, https://doi.org/10.1016/S1473-3099(20)30129-8. (http://www.sciencedirect.com/science/article/pii/S1473309920301298)
[3] Anuradha Tomara, Neeraj Gupta(2020) Prediction for the spread of COVID-19 in India and effectiveness ofpreventive measures
[4] Chen, Y. C., Lu, P. E., Chang, C. S. (2020). A Time-dependent SIR model for COVID-19. arXiv preprint arXiv:2003.00122.
[5] Cooper, I., Mondal, A., Antonopoulos, C. G. (2020). A SIR model assumption for the spread of COVID-19 in different communities. Chaos, Solitons Fractals, 139, 110057.
[6] Mandal, M., Jana, S., Nandi, S. K., Khatua, A., Adak, S., Kar, T. K. (2020). A model based study on the dynamics of COVID-19: Prediction and control. Chaos, Solitons Fractals, 109889.
[7] Ni, W., Yang, X., Yang, D. et al. Role of angiotensin-converting enzyme 2 (ACE2) in COVID-19. Crit Care 24, 422 (2020). https://doi.org/10.1186/s13054-020-03120-0
[8] Omer, S. B., Malani, P., Del Rio, C. (2020). The COVID-19 pandemic in the US: a clinical update. Jama, 323(18), 1767-1768.
[9] Peng, L., Yang, W., Zhang, D., Zhuge, C., Hong, L. (2020). Epidemic analysis of COVID-19 in China by dynamical modeling. arXiv preprint arXiv:2002.06563.
[10] Piovella, N. (2020). Analytical solution of SEIR model describing the free spread of the COVID-19 pandemic. Chaos, Solitons Fractals, 140, 110243.
[11] Prabhakaran, S. (2020). ARIMA Model - Complete Guide to Time Series Forecasting in Python | ML+. ML+. Retrieved 18 December 2020, from https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/.

[12] Robert Nau, The mathematical structure of ARIMA models http://people.duke.edu/ rnau/Mathematical_structure_of_ARIMA_models–Robert_Nau.pdf
[13] Shahid, F., Zameer, A., Muneeb, M. (2020). Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. Chaos, Solitons Fractals, 140, 110212.
[14] Tang, Z., Li, X., Li, H. (2020). Prediction of new coronavirus infection based on a modified SEIR model. medRxiv.
[15] Vadyala, S. R., Betgeri, S. N., Sherer, E. A., Amritphale, A. (2020). Prediction of the number of covid-19 confirmed cases based on k-means-lstm. arXiv preprint arXiv:2006.14752.
[16] Vinay Kumar, Reddy Chimmula, Lei Zhang (2020), ATime series forecasting of COVID-19 transmission in Canada using LSTM networks.
[17] Zeroual, A., Harrou, F., Dairi, A., Sun, Y. (2020). Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study. Chaos, Solitons Fractals, 140, 110121.