Georgia Institute of Technology
Institute for Data Engineering and Science
COV-IDEAS-20 REU

Overview of the Effectiveness of COVID-19 Prediction Models

Frank D'Agostino
Roshan Pulapura
Yue (Amanda) Yao

Advisor: Dr. Eric Schwartz

June-August 2020

Contents

Abstract

As the novel coronavirus SARS-CoV-2 continues to spread across the United States and around the world, governments and businesses are turning to pandemic forecasting models to inform their decisions on closing public spaces, enforcing quarantine measures, providing financial assistance, and more. Over the past several months, academics and private researchers have developed thousands of models to predict the trajectory. Unfortunately, many of the earlier models failed to predict the current trend of the pandemic, and amongst the newer models, there is significant disagreement as to what the future of COVID-19 will look like. Therefore, the goal of this project was to categorize and assess the effectiveness of this wide variety of models to determine how they compare to one another and how well they have done thus far in predicting new cases. Of the models we evaluated, most belonged to at least one of: compartmental (SIR-based) models, statistical models (ARIMA, ARGO, Bayesian), or deep learning/neural networks. Using data from the Johns Hopkins University Coronavirus database, Google Trends, Oxford, and Apple, we tested basic implementations of several models within these three categories. Ultimately, we developed and tested SIR, SIRD, SIRD with Dynamic R_t, Naïve Bayes, Bayesian Standard Time Series with MCMC, ARIMA, ARGONet and LSTM. In addition to our own models, we calculated a variety of error measurements for models developed by other researchers, as provided by the Center for Disease Control. In summation, due to the numerous factors contributing to the spread of the disease beyond government response and general epidemic dynamics, we determined that building a consistently dependable model that can make long-term forecasts can be impractical without having reasonable assumptions about the future values of these various parameters.

1 Introduction

1.1 Overview

The novel coronavirus SARS-CoV-2 is currently the top focus of scientists and policymakers as it grows rampant in the United States and other countries around the world. Commonly referred to as COVID-19, the disease was first identified in Wuhan (Hubei), China [9]. Based on its exponential spread in China, the World Health Organization (WHO) Emergency Committee declared a global health emergency on January 30, 2020 [30]. The outbreak eventually reached other countries such as the United States, and it was officially declared a pandemic by the WHO on March 11, 2020. This sparked countries to strengthen lock-downs, reduce travel, and increase public health communication [20].

When compared to SARS-CoV-1 or MERS-Cov, COVID-19 seems to spread much easier (with a basic reproduction value of 2 to 3.5) and can spread from asymptomatic individuals [32]. During this unprecedented public health emergency, making time-sensitive decisions optimally is crucial. As a result, mathematicians and epidemiologists have been producing a wide array of predictive models to forecast infections and deaths. For example, many

researchers have implemented a standard epidemiological compartmental model, where populations are split into susceptible, infected, and removed (SIR) groups, and transmission is modeled using differential equations. Thus, this type of model is often referred to as a SIR epidemiological model [25]. A more in-depth explanation of the standard SIR model will be provided in Section 2.1. However, modeling groups across the world have given conflicting predictions and suggestions for how to best handle the pandemic, causing people to lose trust in predictive COVID-19 models [11]. Therefore, it is vital to understand how to improve these predictions and how to better aggregate models to give reliable results.

While the wide variations may be concerning to some, modelers are all too familiar with the uncertainty that is inherent to all models. Specifically within the context of COVID-19, the task of forecasting parameters of the pandemic is a layered problem. According to "Why is it difficult to accurately predict the COVID-19 epidemic?", possible answers for the wide range of predictions are that there may not have been enough data at the start of the pandemic, not enough reliable figures due to data collection/recording inconsistencies, and/or not enough people being confirmed out of everyone who is actually infected. The difference between cases and infections changes significantly across viral infections, especially ones that spread through air droplets such as SARS-CoV-2 [24]. The unreliability of ground-truth data, coupled with incomplete knowledge of the virus itself, magnifies the uncertainty of COVID-19 models [31].

Despite uncertainties, models are essential and require constant work to achieve the ability to forecast COVID-19 and future diseases effectively. As more groups work on and create models to predict impacts of COVID-19, researchers are utilizing different mathematical techniques. Understanding the strengths and weaknesses of different mathematical techniques can be beneficial in crafting models to answer specific questions policymakers and people want answered. For example, researchers have found that simpler models can produce more accurate forecasts when compared to complex models [24]. As a result, the goal of our work was to better understand the broad array of modeling techniques and how their strengths can be leveraged to best predict COVID-19 impacts.

1.2 Categorization of COVID-19 Models

To assess the breadth of mathematical models, we conducted an extensive literature review of over 40 papers and categorized the types of models being implemented. The database of papers can be found in the "Data and Code Availability," Section 5. We found that broadly, we could split models we found into three categories: statistical/probabilistic, compartmental, and machine learning. While not completely comprehensive, we feel these categories help compartmentalize the models into their purest forms. Of course, there can be significant overlap in these categories as well. Unique combinations of these techniques can be implemented to achieve more interesting results. However, as studies have shown that simpler models perform significantly better than complex models for COVID-19 forecasting, we opted to focus on fundamental models to observe their efficiencies [24]. A tree outlining our

classification scheme can be seen in Figure 1.

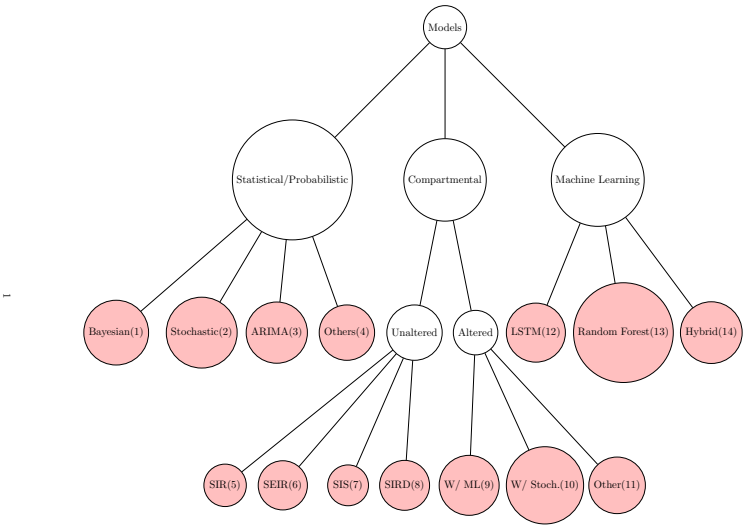


Figure 1: Classification tree of different mathematical modeling techniques for COVID-19. After combing through more than 40 papers designing and implementing prediction models for COVID, we chose this breakdown of model types based on our findings. The nodes highlighted in pink represent the 14 types we felt covered all of the models we found, such that each model could be labeled as such.

2 Models

2.1 SIR

As briefly mentioned in the introduction, the standard SIR epidemiological model is a type of compartmental model. Simply, the population is split into three "compartments" of susceptible, infected, and removed, and ordinary differential equations depict the flow in and out of each compartment. Through this dynamic process, researchers can model the size of each group as a function of time. Susceptible people are people that are not yet infected and are typical, healthy adults. Infected people are people that are carrying COVID-19 and are actively able to spread it to other susceptible individuals. Recovered people are people that recovered and now have immunity to the disease (meaning they cannot get infected again), where the person can no longer infect susceptible people.

There are typically two parameters in a standard SIR model. First, there is the effective contact rate of disease, which describes the transmission of the disease throughout the population. This is typically represented by β. The second parameter is the amount of time someone is contagious for, which is simply captured by the rate of recovery γ [29]. Each

group at time t is represented by $S(t), I(t)$, and $R(t)$ for susceptible, infected, and removed, respectively. The entire population is represented by N. At all times, the sum of groups must equal the population such that $S(t) + I(t) + R(t) = N$ [8]. The ordinary differential equations describing the model are as follows:

$$\frac{dS}{dt} = -\frac{\beta SI}{N}$$
$$\frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I$$
$$\frac{dR}{dt} = \gamma I$$

This model describes the rate of growth and decay of each compartment at each time step. These are typically the standard equations used in a SIR model [2]. The growth of the model over time largely depends on the initial conditions. If only 1 person is infected at the start of the model, then there's a small chance the infection could spread to the rest of the population. Additionally, if the population is huge, then starting with a large amount of infected people might still not be enough to spur a "pandemic" in the model.

While a SIR model is a famous epidemiological that is simple, easy to understand, and viable for many situations, it has limitations. Made in 1927, its simplicity leaves little room for the model to take into account other factors and is not very dynamic [29]. Reducing diverse populations, intricate social dynamics, and fickle viral properties to three compartments can oversimplify the issue and produce results that are not representative of the situation. Another large disadvantage is that a SIR model ignores any latency period in which a person carries a virus but is not symptomatic. In regards to the COVID pandemic, it also does not account for government interventions (such as social distancing) nor hospital response/diagnosis procedures [8].

In order to find the best β and γ values to describe the system, we can utilize the least-squares optimization strategy. This involves calculating the squared distance between each ground-truth point and the point at time t of our model, and then minimizing the error based on the inputted parameters. Given data $\{(x_1, y_1), (x_2, y_2), ...(x_N, y_N)\}$, we can define the error as:

$$E(a, b) = \sum_{n=1}^{N} (y_n - (ax_n - b))^2$$

If we differentiate this error function E and set it equal to 0, we can find the values of a and b in which the error is minimal. By simplifying these equations, we can create a system of equations and solve for a and b by using matrices [19]. Using Python, the least-squares optimization was implemented using the open-source *scipy* package. The following function was written to pass in the parameters, solve the SIR model using *odeint*, and minimize the error and optimize the parameters:

```
from scipy.integrate import odeint
from scipy.optimize import minimize

params = Parameters()
params.add('beta', value=beta, min=0, max=1)
params.add('gamma', value=gamma, min=0, max=1)

def ode_sol(t, init, params):
    beta, gamma = params['beta'].value, params['gamma'].value
    out = odeint(SIR, init, t, args=(beta, gamma))
    return out

def error(param, init, tspan, data):
    sol = ode_sol(tspan, init, param)
    return (sol[:, 0:3] - data).ravel()

result = minimize(error, params, args=(init, tspan, data),
                  method='leastsq')
```

The t and *tspan* variables are the list of times (usually representative of days), *init* is the array of initial S, I, R group values, *data* is the array of ground-truth data that the model is being trained on, and *params* are the inputted first guesses for the β and γ parameters.

The model was run and trained on United States case data, with the model iterating on each prediction day. Due to the limitations and simplicity of the model, only 1 week predictions were made to reduce the error of making long-term forecasts. The full extent of code can be found in the "Data and Code Availability," Section 5.

2.2 SIRD

Compartmental models are versatile because they allow new compartments to be added to better describe or denote subgroups within a population. For example, some modelers use SEIR modelers, which split the infected group in SIR models into E (exposed) and I (infected) groups. In a SEIR model, exposed individuals are ones that carry the disease but are not yet showing symptoms or are infectious, while infected people can spread the disease to susceptible individuals [14]. However, the novel nature of COVID-19 has made understanding the transmission dynamics complicated. Studies have shown that presymptomatic individuals can also spread the virus, contradicting the built-in behavior of a SEIR model [21]. Additionally, the lack of ground-truth data to determine the amount of exposed individuals make optimizing the model difficult.

Therefore, we decided to focus on a SIRD model, which rather splits the "removed" group into recovered and dead. Unlike exposure data, hospitals can provide much more factual data on

who recovered and died from COVID-19, making optimization and validation a much simpler task. The SIRD also has three parameters, which are the effective contact rate of disease β, recovery rate γ, and death rate from disease ν. With the entire population represented by N, the sum of groups must equal the population such that $S(t) + I(t) + R(t) + D(t) = N$ [3]. The ordinary differential equations describing the model are as follows:

$$\begin{aligned}
\frac{dS}{dt} &= -\frac{\beta SI}{N} \\
\frac{dI}{dt} &= \frac{\beta SI}{N} - (\gamma + \nu)I \\
\frac{dR}{dt} &= \gamma I \\
\frac{dD}{dt} &= \nu I
\end{aligned}$$

Like the SIR model, the system of equations describes the growth and decay rate of each group for a given time step [3]. For optimization, the least-squares method was also used to optimize for the parameters β, γ, and ν. The Python code and packages were analogous to the SIR model, except the data contained ground truth susceptible people, infections, recoveries, and deaths. As such, more granular ground-truth data was hypothesized to improve the model forecasts significantly.

2.3 Dynamic R_t

Epidemiologists typically describe and compare virulent diseases through a value called the basic reproduction number, typically depicted by R_0. This represents the number of infections that would occur from a single person in a population where everyone is able to get the disease. If the value is greater than 1, disease spread usually grows, and if it is less than 1, the disease is not sustainable and dies out [7]. Typically, for standard SIR models, the basic reproduction number is the ratio of the contact rate and the recovery rate, such that $R_0 = \frac{\beta}{\gamma}$. For a SIRD model, the basic reproduction number is rather represented by $R_0 = \frac{\beta}{\gamma+\nu}$, where β is the contact rate, γ the recovery rate, and ν the death rate from disease [17].

However, a single reproduction number to describe the dynamics of a pandemic can be problematic and not account for the complexity of policy and public health awareness. As such, several groups have been focusing on the time-changing reproduction factor, $R_e(t)$, which is how many people an infected person will on average infect in a day t [28]. This dynamic reproduction factor can better account for the ever-changing policies and lockdown measures that states and countries are implementing.

In order to integrate government intervention and social response into forecasts and ultimately the $R_e(t)$, data from both Oxford and Apple were utilized. The Oxford COVID-19 Government Response Tracker is a publicly available database that keeps track of time series data regarding policies countries are enacting in regards to health, the economy, pandemic restrictions, and overall stringency in response plans [5]. The Apple Mobility Data is time

series data of relative Apple Maps travel requests split into walking mobility, driving mobility, and transit mobility. This data is relative to the baseline date of January 13th, 2020, and can serve as an indicator of the public's adhesion to health guidelines and restrictions. The data is publicly accessible here: `https://www.apple.com/covid19/mobility`.

To produce a R_t value, a multiple linear regression was run on the policy and mobility data. These datasets served as the predictors (independent variables), while the $R_e(t)$ value at each time t was the single dependent variable, or criterion [1]. A plot of some of the predictors and the United States daily case count can be seen in Figure 2a. Once the regression was conducted and the coefficients calculated, R_t values at each time step were produced by predicting the values via the regression. The code and outputs for this implementation can be found on the GitHub page in Section 5.

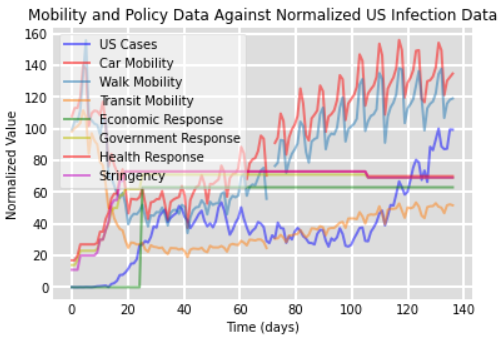
Another group of researchers released an interactive website (`https://rt.live/`) where they were tracking R_t values for every state in the United States using a generative Bayesian model [28]. To assess different approaches, all data was grabbed from the site. Using a weighted average dependent upon the proportion of cases from each state at each day, a R_t series was calculated for the United States.

To implement the R_t value into a SIR or SIRD model, the R_t series needed to be continuous to account for small increments in the time steps. As such, the *scipy.interpolate* library was used to conduct a spline fit algorithm, which can interpolate the discrete data points and produce a continuous set of points [18]. The produced spline fit of the R_t values can be seen in Figure 2b. A $beta\_func$ function can be introduced to calculate the β parameter based on a given R_t, γ, and ν parameter within the model as follows:

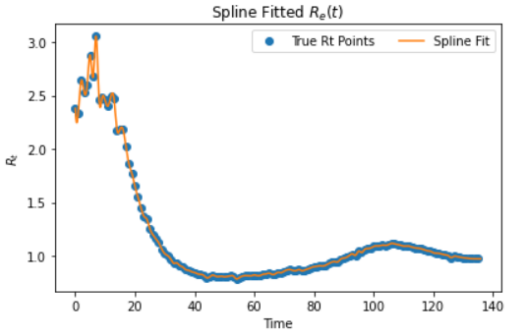
```
from scipy.interpolate import CubicSpline
cs = CubicSpline(x, y)

def beta_func(t, recov, death):
    return cs(t)*(recov+death)

def SIRD(y, t, beta_func, rec, death):
    S, I, R, D = y
    N = S+I+R+D
    # Ordinary differential equations
    dS = (-beta_func(t, rec, death) * S * I)/N
    dI = ((beta_func(t, rec, death) * S * I)/N) - ((rec+death) * I)
    dR = rec * I
    dD = death * I
    return(dS, dI, dR, dD)
```

In this case, x is the time series of the R_t array, and y is the R_t series itself. The $beta\_func$ function then runs the cubic spline on the inputted time value t and multiplies it by the

γ (*recov*) and ν (*death*) parameters to get the β parameter. The differential equations are then altered to include the function.



(a) Mobility and Policy Data Plot



(b) Rt.live Based R_t Value Plot

Figure 2: (a) Plot of the relationships between the Oxford Government Response Tracker data, Apple mobility data, and United States normalized daily new cases from Johns Hopkins University. Clearly, as cases increased, mobility also drastically increased, contradicting a rise in more stringent government response. This could be due to the fact that a stronger government response could not be effective if strict enforcement of policies was not implemented. (b) Plot of the discrete and continuous $R_e(t)$ values, calculated by a weighted average from rt.live state-level R_t values. Clearly, COVID-19 was extremely infectious early on in many states, eventually rising below 1 as initial responses and public perception was strong. Eventually however, the reproduction factor went slightly above 1 again. The increased initial population of infected people could account for the large spike in cases saw in middle-late July.

2.4 Random Forest

Straying from standard epidemiological compartmental models, there are a variety of machine learning techniques in order to perform classification or regression tasks. One such technique is a Random Forest algorithm. To begin with, a decision tree is a flow-chart type structure where each node in a tree represents a classification (above or below a certain value). Someone can trace down the decision tree and eventually reach an end result [12]. A random forest is an ensemble of these decision (or regression) trees, randomly sampling data with replacement and generating large decision trees for each sample. These iterations occur and then the average of results from the group of trees constitute the final prediction [27].

Similarly to the SIRD model, data from both Oxford and Apple were utilized [5]. The Apple mobility data is publicly accessible here: `https://www.apple.com/covid19/mobility`. More information regarding these data sets can be found in Section 2.3. For the Oxford data, the full extent of variables were utilized, ranging from ordinal scales of work and school lock downs, public gathering restrictions, economic stimulus packages, and public health guidelines. The model was run using the *sklearn* package, and can be seen as follows:

```
# Import random forest model
from sklearn.ensemble import RandomForestRegressor

# Begin model
rf = RandomForestRegressor(n_estimators=1000, random_state=42)

# Train model on training data
rf.fit(train_features, train_labels)
```

The *train_features* and *train_labels* are the selected training predictors and criterion, in which the model is then fitted and eventually run on test data to forecast U.S. cases. The *sklearn* package also allows us to visualize the impact of each variable on the model, which can be seen in Figure 7.

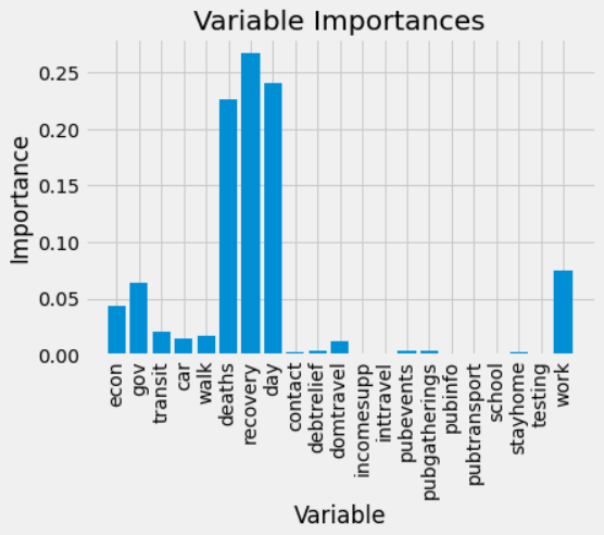


Figure 3: Bar plot of the relative importance of each variable in the random forest model. The top three predictors are the day of infection, the deaths, and the number of recoveries. The fourth best predictor was the ordinal scale of workplace lockdowns.

Random forest is limited by the quantity and quality of data provided, and as evident by Figure 7, many of the factors had almost no importance in swaying the model. Additionally, these factors are not completely comprehensive of every factor that impacts COVID-19 infections. In several models, random forest regressions performed poorly when compared to other machine learning methods [23]. Therefore, this method can be limited.

2.5 Long Short-Term Memory (LSTM)

In order to test the effectiveness of a deep learning model in comparison to the pure statistical models and compartmental models, we developed an LSTM (Long Short-Term Memory)

neural network. LSTM is the primary deep learning algorithm used to make time-series forecasts, and it has been used by several other COVID-19 research projects [33].

LSTM is a recurrent neural network, meaning that it differs from a standard neural network in that it can use information from previous points in time to inform classification/regression on later points in time. In the case of COVID-19, for example, this could mean that RNNs are capable of predicting the number of daily cases (based on factors such quarantine measures and unemployment aid) differently depending on whether surrounding time points are showing a plateau in number or cases or a rapid increase. LSTM is generally considered the most sophisticated RNN, and it improves upon the standard RNNs by allowing for greatly increased long-term memory. Essentially, LSTM introduces a "forget" gate into its cell states, triggered by the activation function tanh, which allows the LSTM to determine whether or not information should be remembered, thereby reducing the amount memory and time required remember long-term time dependencies, and eliminating the exponential propagation of errors across long chains of cells [6].

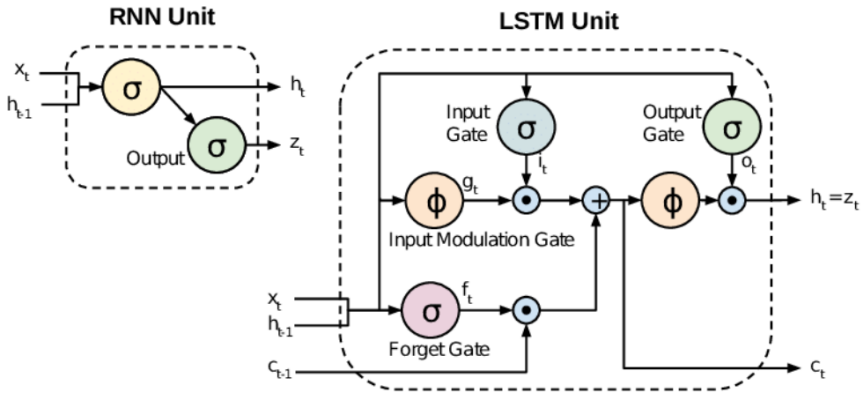


Figure 4: This diagram represents the difference between an LSTM cell and a standard RNN cell. As described, the LSTM includes a "forget" gate which allows it to be selective in which data to preserve later in the sequence and which to forget.

Our LSTM was implemented in TensorFlow using Keras. LSTMs work best when the information is dependent on lots of different variables—unlike the compartmental models, for example, which can only handle three or four variables at a time in the actual differential equations (S,E, I,R,D, etc) without becoming unwieldy. Therefore, we included from the JHU dataset the basic forecasting values—confirmed cases, incident cases, daily deaths, recovered cases, and hospitalizations—as well as additional parameters from the Oxford Coronavirus Government Response Tracker, which included school closings, workplace closings, cancellation of public events, stay at home orders, restrictions on internal movement, travel bans, income support, debt/contract relief, international support, public information campaigns, testing policy, contact tracing, emergency investment in healthcare, vaccine research, etc. All of these factors would also impact the rate at which people spread the virus to one another.

The dataset, which spanned from January 23, 2020 to August 2, 2020 was separated out into training and testing datasets at a given pivot date. We tested numerous values for the pivot date (starting late June and going through late July) to see if having additional training information after the late June resurgence of cases affected the results. The data were then scaled using the MinMaxScaler from the scikit-learn package to the range (0,1) and the inputs and outputs were sectioned into date ranges—we tried several values for the length of the date range and found that 20 was the most effective. The model itself included four LSTM layers, each with 50 units. Each of these layers also included a dropout of 0.2 to regularize/prevent overfitting. The Adam optimizer was selected for solving, and mean squared error was selected as the loss function. A batch size of 32 was used for 100 epochs. Ultimately, the predictions were re-scaled and plotted against the actual values for the parameters (incident cases and cumulative cases) for the dates in the testing set. The LSTM was successful in predicting the trends of both daily incident cases and cumulative cases when larger training sets were used, as is demonstrated in the graphs below.

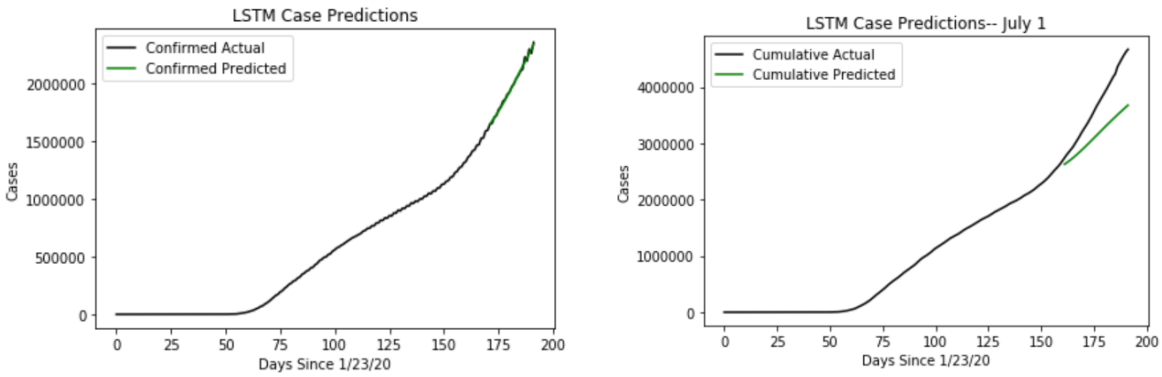


Figure 5: Plots of the number of cumulative cases in the United States vs. time in elapsed days since Jan 23. Predicted values are shown in green and actual measurements are shown in black. The left plot shows the predictions when trained until July 11, while the right plot shows the predictions when trained until July 1.

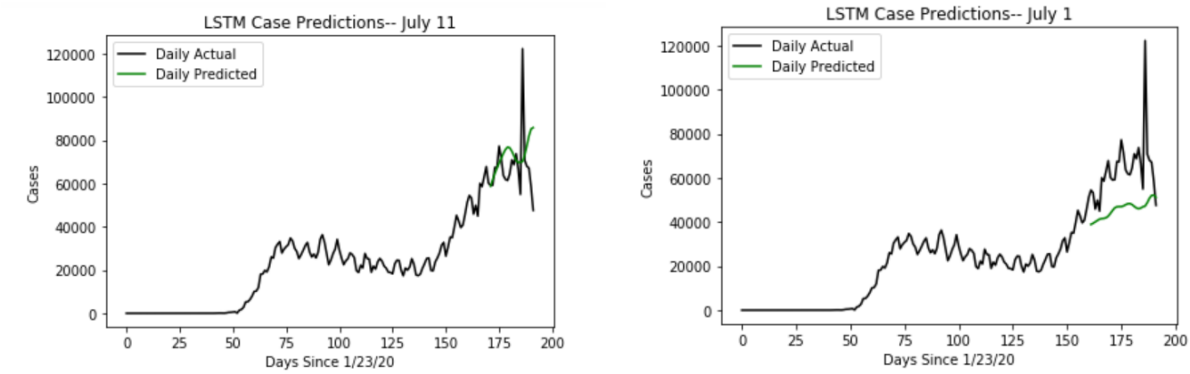


Figure 6: Plots of the number of incident daily cases in the United States vs. time in elapsed days since Jan 23. Predicted values are shown in green and actual measurements are shown in black. The left plot shows the predictions when trained until July 11, while the right plot shows the predictions when trained until July 1.

These results can be explained both in terms of the actual structure of an LSTM and the nature of the coronavirus data. That is, in late June and early July, the United States saw a rapid surge of cases, meaning that the data began to trend very differently from the trends in late May and early June. When the LSTM is trained on data in a close date range that captures the new trend, it successfully extrapolates that trend further. But when it is trained on dates where that spike in cases is completely absent, it was unable to predict the spike based on the values of the additional parameters (i.e. the Oxford data). This difficulty has been reported by several other researchers in the COVID-19 deep learning research space. In fact, Dr. Aditya Prakash, a Professor of Computer Science at Georgia Tech and co-advisor to this project, found that the inability of basic LSTMs to predict future spikes when these trends do not occur in the past was a primary motivation for the DeepCOVID model.

2.6 Bayesian Markov Chain Monte Carlo

In addition to compartmental models and deep learning models, this project also analyzed several pure statistical methods including the Bayesian Markov Chain Monte Carlo (MCMC) technique as well as ARIMA (next section). In the literature, we found that Bayesian techniques were often not used in isolation for predicting the number of cases or deaths on a given day. Rather, they were commonly used for parameter estimation in various compartmental models. Some examples of prior research in this space include the paper by Wang et. al.using Bayesian techniques to estimate a dynamic R_t value, and the paper by Yi et. al. using Bayesian techniques to estimate what percentage of the population should be tested to ensure that the rate of transmission does not exceed a certain maximum value [26, 10]. However, for this project, we attempted to see how well a purely Bayesian forecast

performed without the restrictions put forth by the SIR model.

$$P(\theta|\mathbf{D}) = \frac{P(\theta)P(\mathbf{D}|\theta)}{P(\mathbf{D})} \tag{1}$$

Bayes's Theorem is given in equation 1. Essentially, it is a way of updating a distribution based on new information. If θ is the parameter vector, then $P(\theta)$ is the prior distribution of θ. The joint distribution $P(\mathbf{D}|\theta)$ represents the likelihood of the data $\mathbf{D}$ occurring given θ. Therefore, the posterior distribution– that is, the probability of θ given the data $\mathbf{D}$, is represented by Bayes's Theorem above as a function of the prior and the likelihood.

Markov Chain Monte Carlo algorithms use Bayes's theorem to make predictions from a target distribution. Starting from a prior distribution, this algorithm repeatedly applies Bayes's theorem to obtain a posterior distribution, and then taking that as the new prior distribution, applies Bayes's theorem again and so forth. Ultimately this Markov Chain converges to a given distribution, which is the target distribution.

Since the COVID data is in time series, we used the Bayesian Standard Time Series (BSTS) model to predict daily cases for each day. The BSTS model is represented as follows:

$$y_t = \mu_t + \tau_t + \beta^T x_t + \epsilon_t$$
$$\mu_t + 1 = \mu_t + \delta_t + \eta_{0t}$$
$$\delta_{t+1} = \delta_t + \eta_{1t}$$
$$\tau_{t+1} = \sum_{s=1}^{S-1} \tau_t + \eta_{2t}$$

where y is the output, μ is the trend, τ is the seasonal pattern, $\beta^T x$ is the regression term, and η and ϵ are errors, in this case taken to be normally distributed.

This model was implemented in the probabilistic programming language Stan, using the Python package PyStan. We split the input data into a training set, from January 23 through July 21, and a testing set from July 22 through August 2. Note that the testing set had to be quite small for the Bayesian model (unlike LSTM or the compartmental models) because each new data point is dependent on the value of the previous point, since it is a Markov Chain. The farther away the model moves from the actual ground truth data, the less accurate it will be.

In terms of the actual Stan program, we created vectors for x and y (training) as well as for μ and δ (given in the code as u and v). We used a normal distribution for y as well as the u and v errors. We updated u and v within the parameter transformation block according to the recursive equations given above. In the generated quantities block, we created vectors

for the forecast values of y, u, and v and used the same formulas as above to update u, v, and y for the forecast indices.

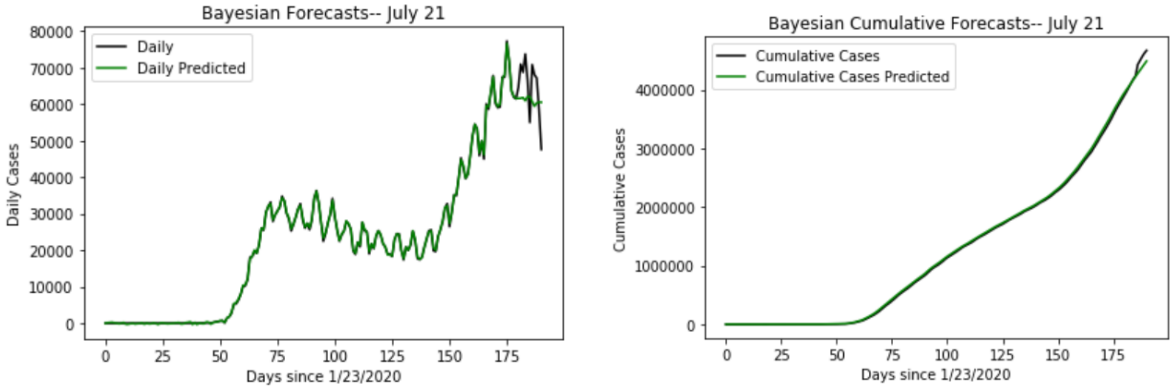


Figure 7: Plots of the number of incident daily cases in the United States vs. time in elapsed days since Jan 23. Predicted values are shown in green and actual measurements are shown in black. The left plot shows the daily predictions when trained until July 21, while the right plot shows the cumulative predictions when trained until July 21.

Ultimately, the Bayesian model fit the training data almost exactly. However, while it did capture the general downward trend of the testing data and closely predicted the cumulative cases for the last 10 days, it failed to produce the same fine-grained accuracy as with the training data. As mentioned earlier, the Markov Chain algorithm grows less and less accurate as it gets further away from ground truth data, since each state directly depends on the previous state. So it is to be expected that the Bayesian model would not maintain that same degree of accuracy, and that it would be more useful for predicting short term forecasts rather than months in advance the way the compartmental models do.

2.7 ARIMA

AutoRegressive Integrated Moving Average (ARIMA) model is a class of model that captures a suite of different standard temporal structures of time series data. This acronym is descriptive, capturing the key aspects of the model itself. Briefly, they are:

- AR: Autoregression. A model that uses the dependent relationship between an observation and some number of lagged observations.
- I: Integrated. The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.
- MA: Moving Average. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

The parameters of the ARIMA model are defined as follows:

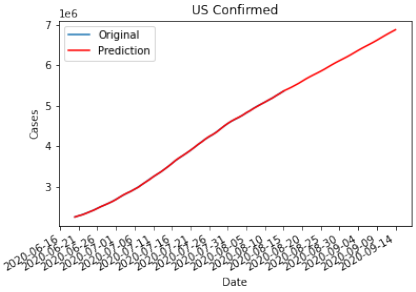
- p: The number of lag observations included in the model, also called the lag order.
- d: The number of times that the raw observations are differenced, also called the degree of differencing.
- q: The size of the moving average window, also called the order of moving average.

These parameters are noted by ARIMA(p,d,q). The general formula for an ARIMA model is as follows:

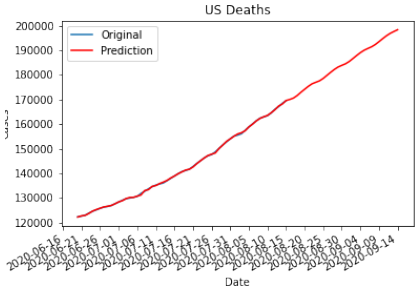
$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + ... + \beta_p Y_{t-p} + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + ... + \phi_q \epsilon_{t-q} + \epsilon_t$$

The ARIMA model is commonly used in time series prediction generally, and is also an effective model in COVID-19 prediction. Perone used ARIMA with AIC and MAE as parameter selecting methods [22]; Chakraborty and Ghosh added a Wavelet-based forecasting model to estimate error based on ARIMA [4]; Kumar et al. combined ARIMA with VAR (Vector Auto Average) to predict the spatial extinct while using remote sensing data and for the purpose of creation of GIS map of worldwide on cumulative cases, deaths, and recovers and GLM (Generalized Logistic Growth Model) measure oscillates some multiple peak parameters inferred in sub-epidemic and pandemic conditions to determine the projected outcomes [13].

For our research, we used the general form of ARIMA model, with Grid Search based on MSE to determine best parameter for every state in US and also the whole country. We used 66% of data as training data and the rest to be testing data. For predictions, we used 1-day ahead forecast and we also 30-day ahead forecast. Result predictions include four categories: cumulative confirmed cases, cumulative deaths, incident cases, incident deaths. The best trained parameters up until August 12th are in the file: "./ARIMA/best_cfg.csv" (due to time limitations, not all of the states were trained). A sample of plots can be seen in Figure 8, 9, 10, and 11. The rest of the plots produced can be found in: "./ARIMA/" under each state folder on the GitHub (see 5). The ARIMA model is a pretty effective model on COVID predictions.

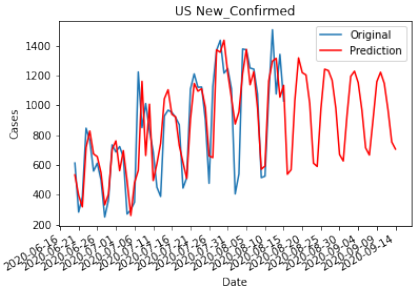


(a) US Cumulative Confirmed Cases

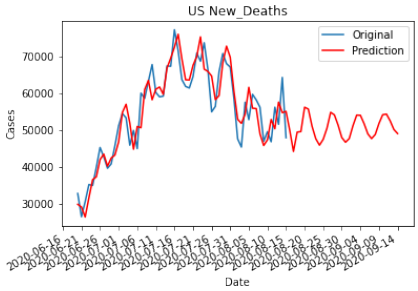


(b) US Cumulative Deaths

Figure 8: Plots of the ARIMA model predictions on US cumulative cases and deaths. The cumulative plot makes small changes difficult to see, but the model seems to predict the trend fairly well.

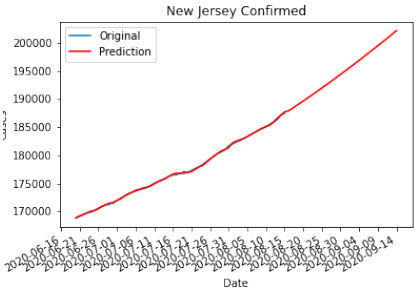


(a) US Incident Cases

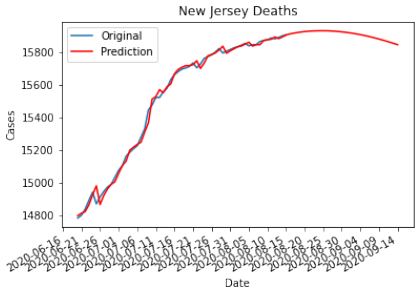


(b) US Incident Deaths

Figure 9: Plots of the ARIMA model predictions on US incident cases and deaths, which are daily recorded totals. The model grabs the altered variation in data over time well.

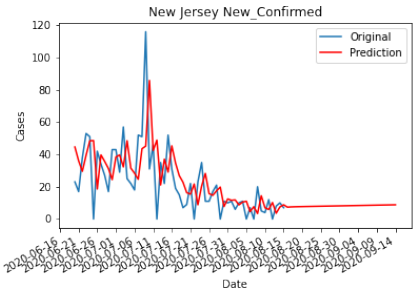


(a) NY Cumulative Confirmed Cases

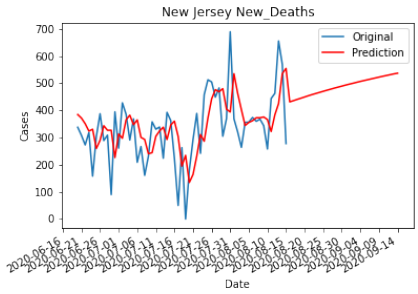


(b) NJ Cumulative Deaths

Figure 10: Plots of the ARIMA model predictions on New Jersey cumulative cases and deaths. On a state level, the predictions seem to still hold up, but perform worse than country wide predictions.



(a) NJ Incident Cases



(b) NJ Incident Deaths

Figure 11: Plots of the ARIMA model predictions on US cumulative cases and deaths. The predictions perform much worse, fitting the trained data well but drastically flattening when predicting past the end date.

2.8 ARGO

During our summer research, we had the pleasure to have Professor Shihao Yang from Georgia Institute of Technology talk about his novel influenza epidemics prediction model called ARGO (AutoRegression with GOogle search data) [34]. ARGO uses the power of big data, which not only incorporates the seasonality in influenza epidemics, but also captures changes in people's online search behavior over time (using Google Trend). It has proven to be a good model for influenza prediction. The team also published a second paper with an augmented ARGO model for state-level influenza epidemics prediction incorporating additional geo-related data [16]. We also found several papers applying ARGO to COVID-19 pandemic prediction. One such paper was "A Machine Learning Methodology for Real-Time Forecasting of the 2019-2020 COVID-19 Outbreak Using Internet Searches, News Alerts, and Estimates from Mechanistic Models" [15].

For an educational exercise and due to data access limitations, we replicated a simple modified version of ARGO. We used lasso regression on AR model with Google Trend information incorporated. We gave a time frame of 1 to 15 days lag for AR and 5 days lag for Google search information. Intuitively, we assume people's COVID-related behavior will be influenced by their Google searches, which indicates what they care about during that time frame in the past 5 days. The regression function being used is:

$$Y_t = \mu + \sum_{j=1}^{15} \alpha_j Y_{t-j} + \sum_{j=1}^{5} \sum_{i=1}^{71} \beta_{ji} X_{i(t-j)} + \epsilon_t$$

$$argmin_{\mu,\alpha,\beta} \sum_t (Y_t - \sum_{j=1}^{15} \alpha_j Y_{t-j} - \sum_{j=1}^{5} \sum_{i=1}^{71} \beta_{ji} X_{i(t-j)})^2 + \lambda \sum_{i,j} |\beta_{ji}|$$

Where $X_{i(t-j)}$ represents the j-day lag of the i^{th} Google search term. We pick 71 Google search terms in total, including some terms such as "covid fever", "how long does covid last", "fever and cough", "covid symptoms", and "covid recovery time."

We only ran the model on the following states since Google trend does not have accessible API usage. The states are as follows: "US", "New York", "California", "Texas", "New Jersey", "Washington", "Massachusetts", "Michigan." A section of a heatmap describing some of the search terms and their effectiveness can be seen in Figure 12. The model was run to predict incident cases and incident deaths. The results can be found in Figures 13 and 14.

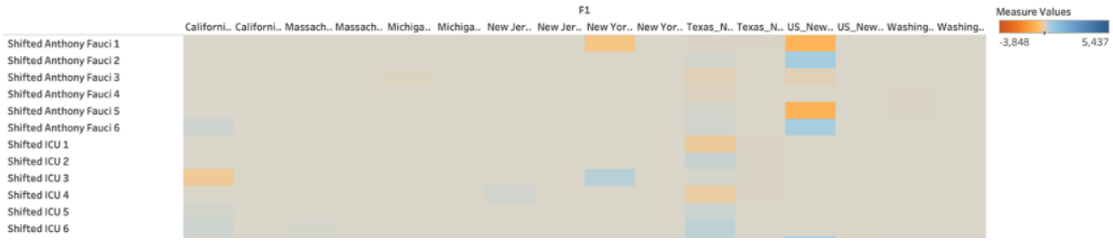
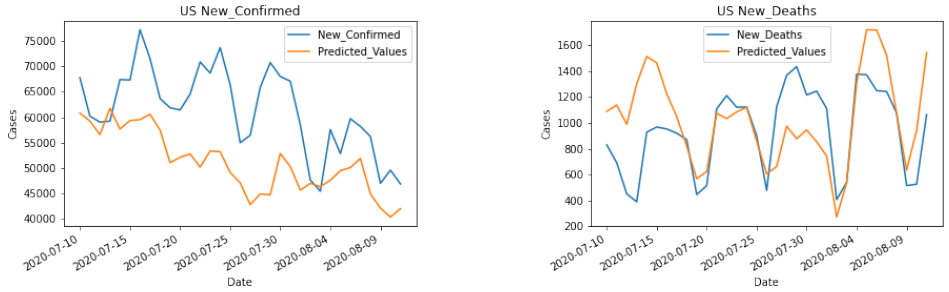


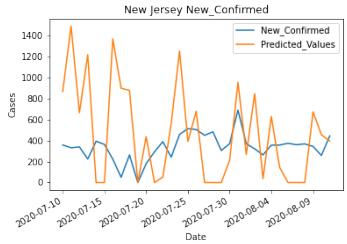
Figure 12: Coefficient heatmap of some of the search terms used in the ARGO model. The full-sized image can be found at: "./ARGOnet/coef_heatmap.png" on our GitHub.



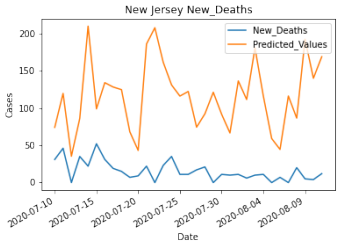
(a) US Incident Confirmed Cases

(b) US Incident Deaths

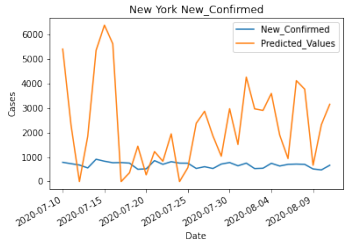
Figure 13: Plots of the ARGO predictions for the United States on daily new confirmed cases and COVID-19 related deaths. For both incident deaths and cases, the model follows the general trend of data fairly well. The model seems to perform much better for incident deaths. However, it seems to consistently under predict incident cases.



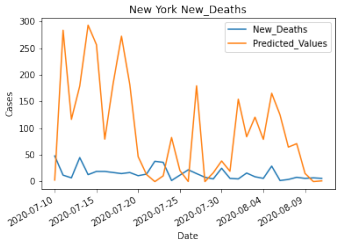
(a) NJ Incident Confirmed Cases



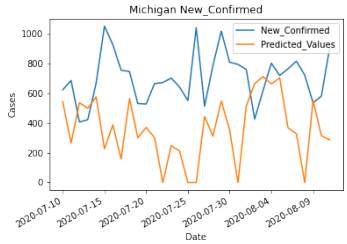
(b) NJ Incident Deaths



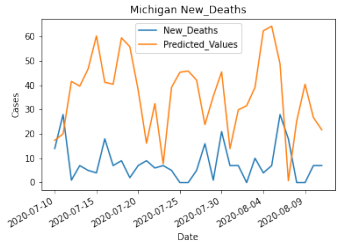
(c) NY Incident Confirmed Cases



(d) NY Incident Deaths



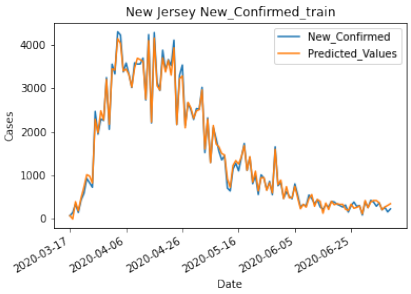
(e) MI Incident Confirmed Cases



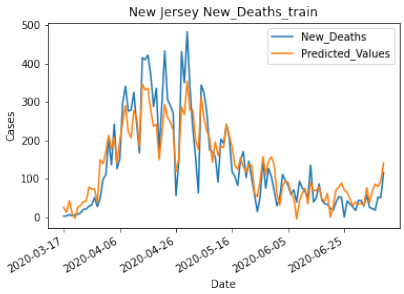
(f) MI Incident Deaths

Figure 14: State level ARGO predictions perform worse than overall United States predictions. This suggests that more geo-related data is needed to better predict smaller populations of people. The model seems to perform better for incident cases rather than daily confirmed deaths.

From the plots, we realize our simple modified version of ARGO performs poorly. It somewhat captures the national level data but it fails to capture and predict on state-level data. However, if we look at how the model fits training data, it fits actually fairly well. Plots for New Jersey can be seen in Figure 15.



(a) NJ Incident Confirmed Cases



(b) NJ Incident Deaths

Figure 15: Plots of the training data for the ARGO model on the state of New Jersey for both incident confirmed cases and incident deaths. The model fits very well, following the variable pattern over time and capturing the general trends of the data.

We believe the following reasons contribute to the overfitting we observed:

- From the original ARGO that Shihao's team is using, we are missing two parts of important data: ILI reports and healthcare/hospitalization related data. These are all good indicators for people's behavior, and could drastically improve the model.
- Google search terms that were being used could be more carefully chosen, preferrably by a professional who has experience in search term trends, linguistic impacts, and methodologies for selecting the best phrases.
- Geo-related and region-specific data like mobility data, county level policies, etc.

For future implementations, we would try to incorporate ILI reports, healthcare/hospitalization related data, geo-related data, etc. and more selectively choose representative Google search terms.

3 Evaluation and Visualization of Predictions

3.1 Evaluation Methods

We will be evaluating our prediction models and prediction models from CDC COVID forecast GitHub. We will evaluate each model with MSE (mean squared error), MAE (mean absolute error), and MAPE (mean absolute percentage error). They are simply calculated by averaging every existing prediction. We also developed a visualization on predictions. Here is a brief overview on how to use our error implementation:

- **Clean Prediction Data.ipynb**: Cleans all necessary data, including:
  - Clean prediction data from CDC COVID forecast GitHub.
  - Get ground truth data from JHU COVID GitHub.

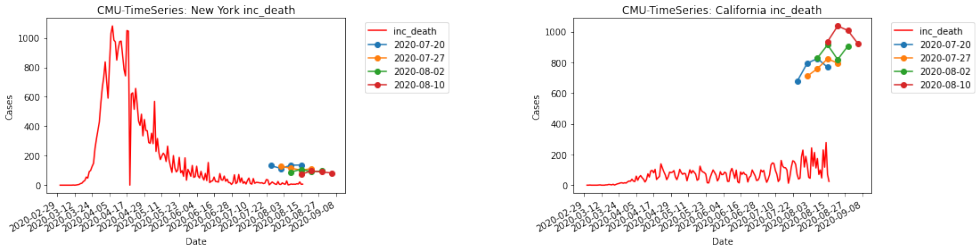
- Clean our prediction data to match the format that is being used in visualization. (Users can ignore this part, since their model and outputs will most likely be structured differently than ours.)

- **Calculate MSE, MAE, MAPE.ipynb**:
  - Calculate MSE, MAE, MAPE for every past prediction.

- **Prediction Visualization Model Measurements.ipynb**:
  - Call $plot\_model\_state(state, model)$ to plot a single model visualization of a specific state. There will be four possible plots if all the predictions exist: cumulative confirmed cases, cumulative deaths, incident cases, and incident deaths. Each line (or maybe only a dot, depending on the input) on each plot will represent the model's predictions made on a specific day.
  - Call $measurement()$ to finish the visualization on all existing models and return four Comma Separated Value (CSV) files (cumulative confirmed cases, cumulative deaths, incident cases, and incident deaths) containing the MSE, MAE, and MAPE of each inputted model.

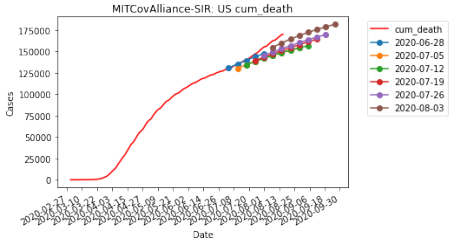
3.2 Sample CDC Model Visualization

We will now show some visualizations of models of US/New York/California (if it exists) that have similar, but more complex versions of the models we produced to give readers a better understanding of how much better or worse a complex model based on simple methods can perform. The rest of the visualizations can be found at: "./Measurements" under each model's folder.

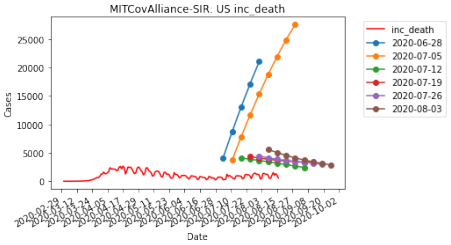


(a) NY Incident Deaths (b) CA Incident Deaths

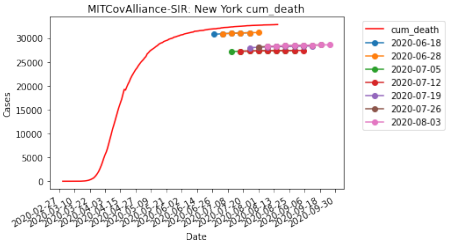
Figure 16: CMU-TimeSeries: Autoregressive time-series model



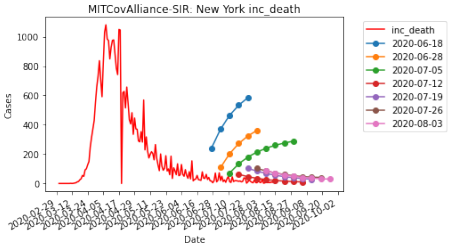
(a) US Cumulative Deaths



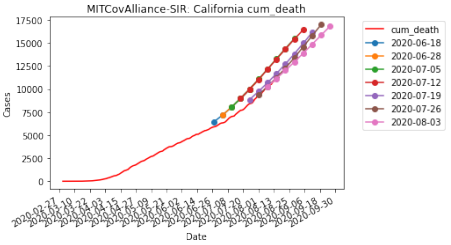
(b) US Incident Deaths



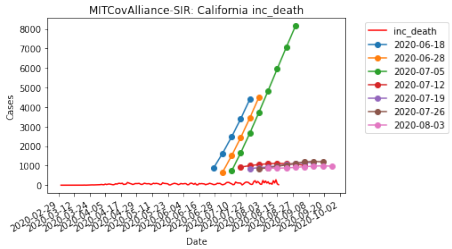
(c) NY Cumulative Deaths



(d) NY Incident Deaths

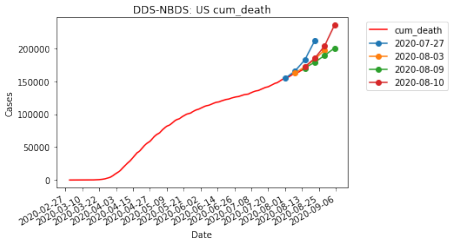


(e) CA Cumulative Deaths

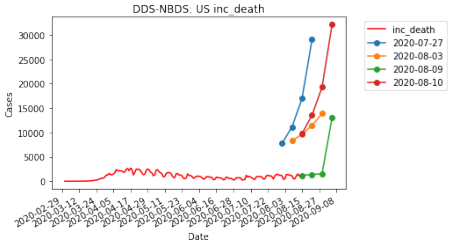


(f) CA Incident Deaths

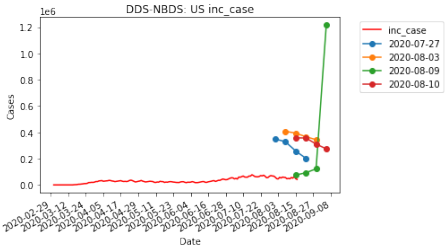
Figure 17: MITCovAlliance-SIR: SIR model



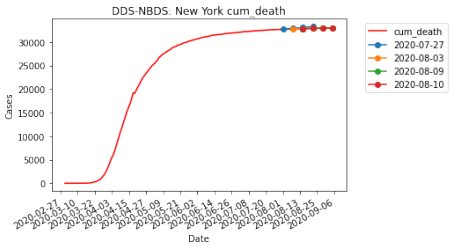
(a) US Cumulative Deaths



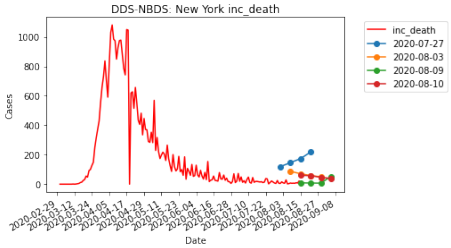
(b) US Incident Deaths



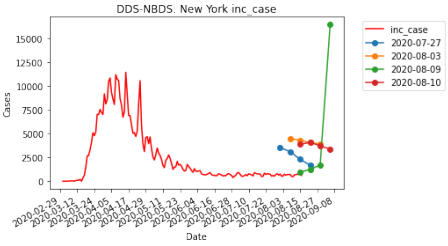
(c) US Incident Cases



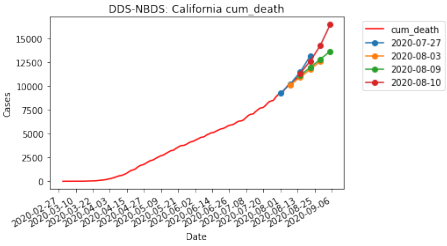
(d) NY Cumulative Deaths



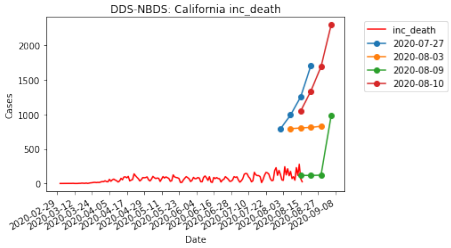
(e) NY Incident Deaths



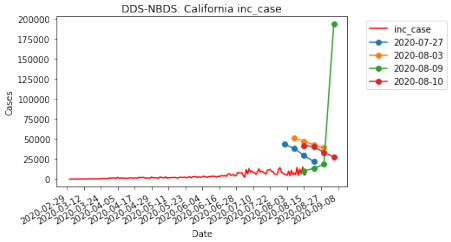
(f) NY Incident Cases



(g) CA Cumulative Deaths



(h) CA Incident Deaths

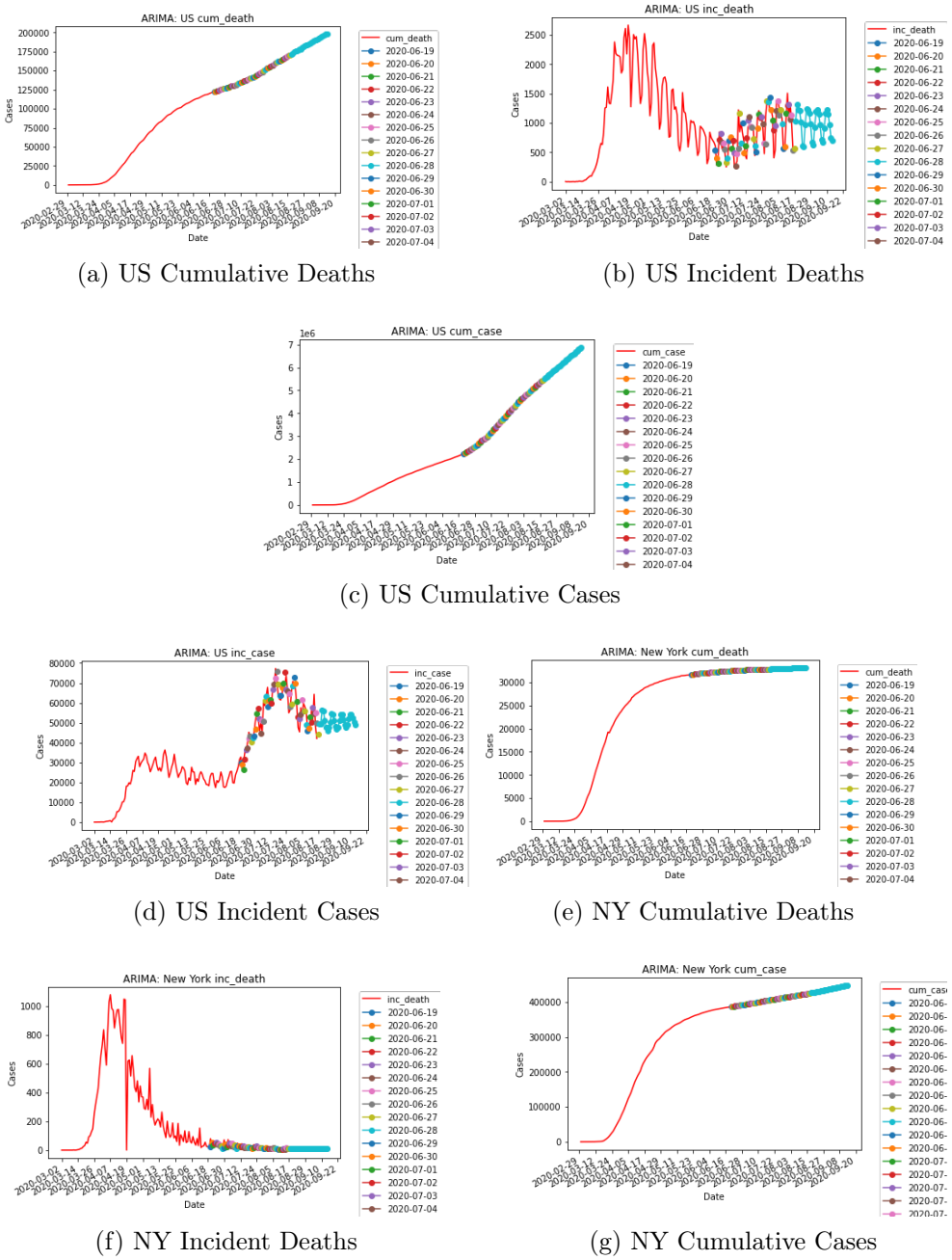


(i) CA Incident Cases

Figure 18: DDS-NBDS: Bayesian hierarchical model

3.3 Our Model Visualizations

We will show visualizations of US/New York (if it exists for that model). The rest of the plots, along with their full legends, can be found at: "./Measurements" under each model's folder.



(a) US Cumulative Deaths

(b) US Incident Deaths

(c) US Cumulative Cases

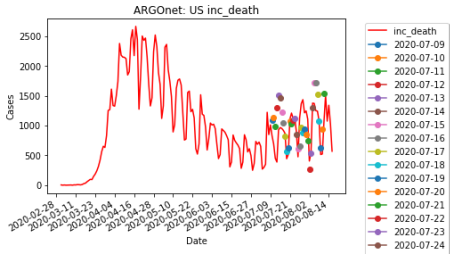
(d) US Incident Cases

(e) NY Cumulative Deaths

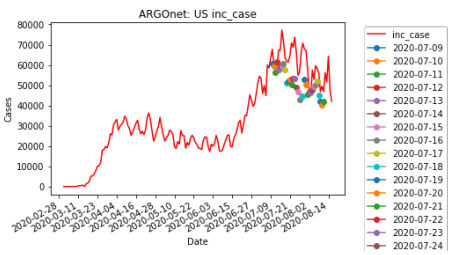
(f) NY Incident Deaths

(g) NY Cumulative Cases

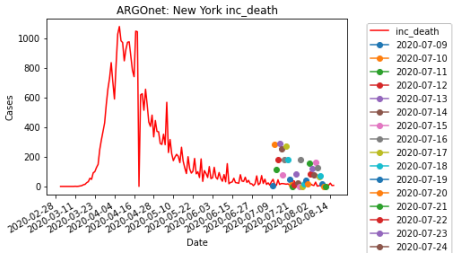
Figure 19: ARIMA: AutoRegressive Integrated Moving Average Model



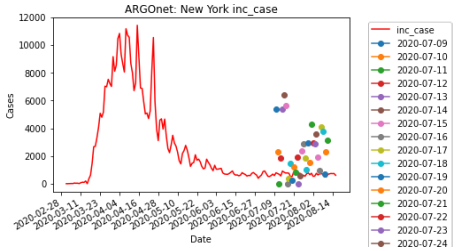
(a) US Incident Deaths



(b) US Incident Cases

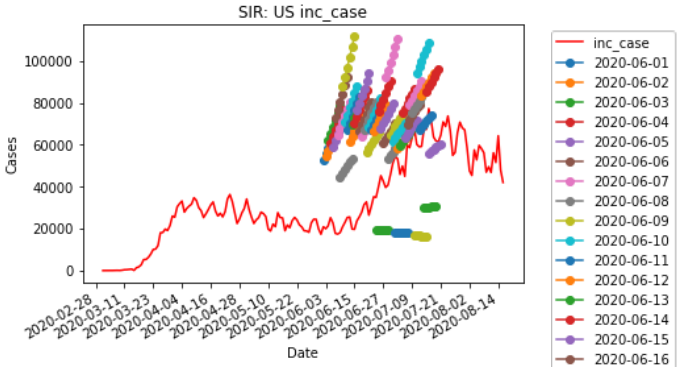


(c) NY Incident Deaths



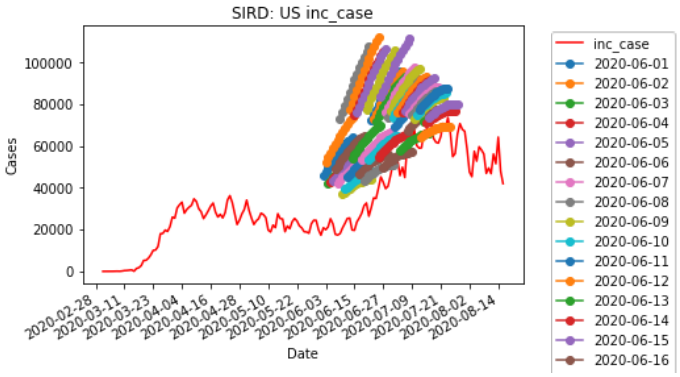
(d) NY Incident Cases

Figure 20: ARGO: AutoRegression with GOogle search data Model



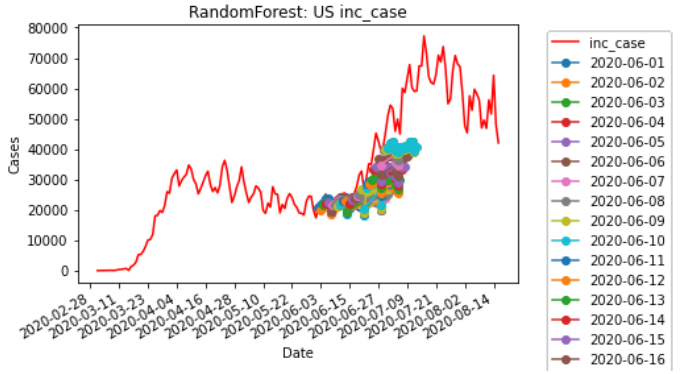
(a) US Incident Cases

Figure 21: SIR Model



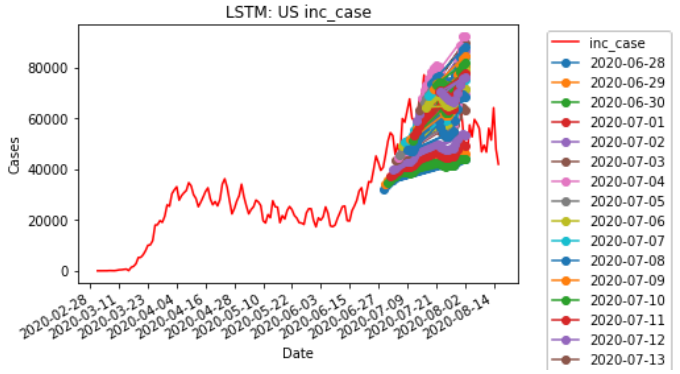
(a) US Incident Cases

Figure 22: SIRD Model



(a) US Incident Cases

Figure 23: RandomForestRegressor Model



(a) US Incident Cases

Figure 24: Long Short-Term Memory (LSTM) Model

3.4 Evaluation Summary

```
---------------------------- Model Evaluation Summary (State-Level) --------------------------------
------------------- Models from CDC COVID forecast Github (State-Level) ----------------------------
................................... Cumulative Deaths ..............................................
              Max              Max_model      Min  Min_model         Mean
MSE   3.25333e+07            CU-nochange  1947.01  CEID-Walk  1.48495e+06
MAE       2007.61  UChicago-CovidIL_30_+  20.0261  CEID-Walk      379.282
MAPE          inf        LANL-GrowthRate  1.48332  CEID-Walk          inf
................................... Incident Deaths ................................................
              Max              Max_model      Min        Min_model     Mean
MSE   1.66624e+06            UT-Mobility  22034.2  LANL-GrowthRate   235522
MAE       1029.12  UChicago-CovidIL_30_+  58.5765  LANL-GrowthRate  246.691
MAPE          inf              CEID-Walk  386.018          STH-3PU      inf
................................... Incident Cases .................................................
             Max             Max_model          Min  Min_model        Mean
MSE   2.5777e+09  CovidActNow-SEIR_CAN  9.36451e+07   DDS-NBDS  3.7964e+08
MAE      15002.9  CovidActNow-SEIR_CAN      4648.83   DDS-NBDS     7570.43
MAPE         inf             CEID-Walk          inf  CEID-Walk         inf
```

(a) Summary: CDC state level

```
--------------------------------- Our Models (State-Level) ----------------------------------------

................................... Cumulative Deaths ..............................................

              MSE       MAE      MAPE
model
ARIMA  163.760094  6.188776  0.643204

................................... Cumulative Cases ...............................................

                MSE         MAE     MAPE
model
ARIMA  222792.650826  200.417447  0.52727

................................... Incident Deaths ................................................

                MSE        MAE  MAPE
model
ARGOnet  9005.078864  64.623925   inf
ARIMA     147.648599   5.942969   inf

................................... Incident Cases .................................................

                 MSE          MAE  MAPE
model
ARGOnet  5.369518e+06  1396.584963   inf
ARIMA    2.066900e+05   195.629417   inf
```

(b) Summary: GT COV-IDEAS state level

```
---------------------------- Model Evaluation Summary (National-Level, US) --------------------------------
-------------------- Models from CDC COVID forecast Github (National-Level, US) ---------------------------
..................................... Cumulative Deaths .............................................
              Max    Max_model        Min  Min_model         Mean
MSE   2.84051e+10  CU-nochange    9837.47  CEID-Walk  7.85641e+08
MAE         35645  CU-nochange     99.184  CEID-Walk      6998.97
MAPE      37.7534  CU-nochange  0.0585222  CEID-Walk       5.9167
..................................... Incident Deaths .............................................
              Max        Max_model         Min       Min_model         Mean
MSE   1.96458e+08      UT-Mobility  3.5226e+06  SWC-TerminusCM  4.07469e+07
MAE        8692.3  JHU_IDD-CovidSP     1696.86  SWC-TerminusCM      5314.99
MAPE      1219.41  JHU_IDD-CovidSP     246.865         STH-3PU      589.653
..................................... Incident Cases .............................................
              Max      Max_model         Min          Min_model         Mean
MSE   5.57373e+11  IHME-CurveFit  5.6957e+10  COVIDhub-baseline  1.63977e+11
MAE        745837  IHME-CurveFit      212896  COVIDhub-baseline       379898
MAPE      1427.07  IHME-CurveFit     462.403           DDS-NBDS      723.669
```

(c) Summary: CDC US

```
------------------------------- Our Models (National-Level, US) -----------------------------------

..................................... Cumulative Deaths .............................................

                MSE         MAE      MAPE
model
ARIMA  36746.350551  138.873163  0.097337

..................................... Cumulative Cases .............................................

                MSE          MAE      MAPE
model
ARIMA  2.732722e+07  4194.814975  0.110252

..................................... Incident Deaths .............................................

                   MSE         MAE       MAPE
model
ARGOnet  123026.760351  276.294129  35.833584
ARIMA     36843.803218  139.287988  21.074121

..................................... Incident Cases .............................................

                       MSE           MAE        MAPE
model
ARGOnet       1.512322e+08  10500.765931   16.429372
ARIMA         2.383418e+07   4060.522650    7.827785
LSTM          2.135998e+08  11655.785705   19.097412
RandomForest  1.400161e+08   9299.423269   22.742351
SIR           1.487566e+09  34351.376461  116.347729
SIRD          1.038242e+09  27530.784738   81.736522
```

(d) Summary: GT COV-IDEAS US

4 Conclusion

Since the compartmental models inherently include only one peak, the SIR-based models struggled to capture the April and July peaks of COVID-19 but were more successful in predicting the number of cases after the second peak occurred. On the other hand, the statistical models such as ARIMA were pretty successful in capturing time series trends but was less successful in capturing large spikes. Our simple version of ARGONet was only able to capture overall trends on a national level but failed on a state level since we were missing

important data source like ILI reports, hospitalization, and geo-related data. Likewise, there was significant overfitting with the Bayesian model, which matched the training data almost exactly but struggled to predict later peaks based on earlier information. Finally, although the LSTM failed to predict future cases when trained on a smaller date range, its performance increased significantly with a larger training set.

5 Data and Code Availability

5.1 Literature Database

The database of COVID-19 modeling papers can be found here: `https://docs.google.com/spreadsheets/d/1uwGd9sBX-mVtcIacEAp3HJBkY63gl4887qk14nVq9Gs/edit?usp=sharing`. Each paper has a general broad classification of its model type, as well as the main parameters it utilized. Also includes the DOI link for a majority of papers, their country of focus, and any links to GitHub pages or code files. The spreadsheet also contains CDC models.

5.2 GitHub

The GitHub page can be found here: `https://github.com/GTIdeas2020REU/gt-covid19-modeling`. All code and models created by us can be found and is publicly accessible at this link. Additionally, all CSV files of our predictions are provided in the same format as the COVID-19 Forecast Hub, which can be found here: `https://github.com/reichlab/covid19-forecast-hub`. Predictions from this page was also pulled to aid in our analysis.

5.3 Graphs

The full array of graphs and figures can be found on our GitHub page here: `https://github.com/GTIdeas2020REU/gt-covid19-modeling`. If referenced in this paper, it is in the directory specified.

References

[1] Leona S Aiken, Stephen G West, Steven C Pitts, Amanda N Baraldi, and Ingrid C Wurpts. Multiple linear regression. *Handbook of Psychology, Second Edition*, 2, 2012.

[2] Fred Brauer. Compartmental models in epidemiology. In *Mathematical epidemiology*, pages 19–79. Springer, 2008.

[3] Cem Cakmakli and Yasin Simsek. Bridging the covid-19 data and the epidemiological model using time varying parameter sird model. *arXiv preprint arXiv:2007.02726*, 2020.

[4] Tanujit Chakraborty and Indrajit Ghosh. Real-time forecasts and risk assessment of novel coronavirus (covid-19) cases: A data-driven analysis. *Chaos, Solitons & Fractals*, page 109850, 2020.

[5] Oxford COVID. Government response tracker. *BBC Research Coronavirus: The world in lockdown in maps and charts https://www. bbc. com/news/world-52103747*, 19.

[6] Raj Dandekar and George Barbastathis. Neural network aided quarantine control model estimation of global covid-19 spread. 2020.

[7] Klaus Dietz. The estimation of the basic reproduction number for infectious diseases. *Statistical methods in medical research*, 2(1):23–41, 1993.

[8] Giuseppe Gaeta. A simple sir model with a large set of asymptomatic infectives. *arXiv preprint arXiv:2003.08720*, 2020.

[9] Giulia Giordano, Franco Blanchini, Raffaele Bruno, Patrizio Colaneri, Alessandro Di Filippo, Angela Di Matteo, and Marta Colaneri. Modelling the covid-19 epidemic and implementation of population-wide interventions in italy. *Nature Medicine*, pages 1–6, 2020.

[10] Dennis Kon-Jin Lin Chun-Ming Yu Grace Yi, Wenqing He. Covid-19: Should we test everyone? 2020.

[11] Inga Holmdahl and Caroline Buckee. Wrong but useful—what covid-19 epidemiologic models can and cannot tell us. *New England Journal of Medicine*, 2020.

[12] Bogumił Kamiński, Michał Jakubczyk, and Przemysław Szufel. A framework for sensitivity analysis of decision trees. *Central European journal of operations research*, 26(1):135–159, 2018.

[13] Pavan Kumar, Himangshu Kalita, Shashikanta Patairiya, Yagya Datt Sharma, Chintan Nanda, Meenu Rani, Jamal Rahmani, and Akshaya Srikanth Bhagavathula. Forecasting the dynamics of covid-19 pandemic in top 15 countries in april 2020: Arima model with machine learning approach. *medRxiv*, 2020.

[14] Michael Y Li and James S Muldowney. Global stability for the seir model in epidemiology. *Mathematical biosciences*, 125(2):155–164, 1995.

[15] Dianbo Liu, Leonardo Clemente, Canelle Poirier, Xiyu Ding, Matteo Chinazzi, Jessica T Davis, Alessandro Vespignani, and Mauricio Santillana. A machine learning methodology for real-time forecasting of the 2019-2020 covid-19 outbreak using internet searches, news alerts, and estimates from mechanistic models. *arXiv preprint arXiv:2004.04019*, 2020.

[16] Fred S Lu, Mohammad W Hattab, Cesar Leonardo Clemente, Matthew Biggerstaff, and Mauricio Santillana. Improved state-level influenza nowcasting in the united states leveraging internet-based data and network approaches. *Nature communications*, 10(1):1–10, 2019.

[17] Junling Ma. Estimating epidemic exponential growth rate and basic reproduction number. *Infectious Disease Modelling*, 5:129–141, 2020.

[18] Ari Maller. Interpolation. 2004.

[19] Steven J Miller. The method of least squares. *Mathematics Department Brown University*, 8:1–7, 2006.

[20] Richard A Neher, Robert Dyrdak, Valentin Druelle, Emma B Hodcroft, and Jan Albert. Potential impact of seasonal forcing on a sars-cov-2 pandemic. *Swiss medical weekly*, 150(1112), 2020.

[21] Hiroshi Nishiura, Natalie M Linton, and Andrei R Akhmetzhanov. Serial interval of novel coronavirus (covid-19) infections. *International journal of infectious diseases*, 2020.

[22] Gaetano Perone. An arima model to forecast the spread and the final size of covid-2019 epidemic in italy. *medRxiv*, 2020.

[23] Matheus Henrique Dal Molin Ribeiro, Ramon Gomes da Silva, Viviana Cocco Mariani, and Leandro dos Santos Coelho. Short-term forecasting covid-19 cumulative confirmed cases: Perspectives for brazil. *Chaos, Solitons & Fractals*, page 109853, 2020.

[24] Weston C Roda, Marie B Varughese, Donglin Han, and Michael Y Li. Why is it difficult to accurately predict the covid-19 epidemic? *Infectious Disease Modelling*, 2020.

[25] Junkichi Satsuma, R Willox, A Ramani, B Grammaticos, and AS Carstea. Extending the sir epidemic model. *Physica A: Statistical Mechanics and its Applications*, 336(3-4):369–375, 2004.

[26] Ling Li-Philip Nadler Rossella Arcucci Yuan Huang Zhongzhao Teng Yike Guo Shuo Wang, Xian Yang. A bayesian updating scheme for pandemics: Estimating the infection dynamics of covid-19. 2020.

[27] Vladimir Svetnik, Andy Liaw, Christopher Tong, J Christopher Culberson, Robert P Sheridan, and Bradley P Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43(6):1947–1958, 2003.

[28] Kevin Systrom, Thomas Vladek, and Mike Krieger. Project title. `https://github.com/rtcovidlive/covid-model`, 2020.

[29] Juliana Tolles and ThaiBinh Luong. Modeling epidemics with compartmental models. *Jama*, 2020.

[30] Thirumalaisamy P. Velavan and Christian G. Meyer. The covid-19 epidemic. *Tropical Medicine & International Health*, 25(3):278–280, 2020.

[31] Meimei Wang and Steffen Flessa. Modelling covid-19 under uncertainty: what can we expect? *The European Journal of Health Economics*, page 1, 2020.

[32] Yixuan Wang, Yuyi Wang, Yan Chen, and Qingsong Qin. Unique epidemiological and clinical features of the emerging 2019 novel coronavirus pneumonia (covid-19) implicate special control measures. *Journal of medical virology*, 92(6):568–576, 2020.

[33] Bingjie Yan, Xiangyan Tang, Boyi Liu, Jun Wang, Yize Zhou, Guopeng Zheng, Qi Zou, Yao Lu, Wenxuan Tu, and Naixue Xiong. An improved method for the fitting and prediction of the number of covid-19 confirmed cases based on lstm.

[34] Shihao Yang, Mauricio Santillana, and Samuel C Kou. Accurate estimation of influenza epidemics using google search data via argo. *Proceedings of the National Academy of Sciences*, 112(47):14473–14478, 2015.