Modeling the COVID-19 Outbreak in Various States in India

Abhishek Bhattacharjee[1]

[1]Dept. of Bioengineering, University of Illinois at Urbana-Champaign, Urbana, IL, 61801

Dedicated to my grandmother, Nilima Bhattacharjee, for being proud of me and encouraging me every step of the way

Abstract

COVID-19 is a novel coronavirus that poses a major health risk to the world. This disease has had dramatic economic and health effects around the world and demands a fast-moving and effective response. One such component of that response includes the modeling and forecasting of the disease spread. This must be done in order to effectively track the spread of the disease and allocate resources accordingly. This type of modeling work is especially important in a place like India where resources are strained, and large populations present easy transmission routes.

This study explores various modeling and forecasting approaches that could be used to track the disease as well as its mechanics in various regions in India. Five modeling methods were used in this study and include the Exponential Smoothing Model, ARIMA Model, LSTM Model, SIR Compartmental Model, and a Prophet Library Model. All five models were used to generate 7-day forecasts with all known data as well as validation forecasts using 70% of the available data for training and 30% for testing. The SIR Model and Prophet Model were specifically used to perform a more in-depth analysis of disease spread, tackling topics such as trend changepoints, and population adjusted spread rates. With regards to inputs, the Exponential Smoothing Model, ARIMA Model, and LSTM model all relied on a univariate approach, with case count data as the only input variable. The SIR and Prophet Models utilized a multivariate approach with the SIR Model using last known population, recovery counts and death counts as extra regressors, and the Prophet Model using Google Human Mobility data as an extra factor.

The study found that the Exponential Smoothing and ARIMA Models present themselves as weak options for modeling this pandemic due to their tendency to underpredict the trend, especially in the earlier stages of a pandemic where exponential growth is observed. The LSTM model was also found to be weak due to the relatively low number of data points (disease data only numbers in the 100s of data points). This led to under or over fitting, causing the model to routinely underpredict. The SIR and Prophet Models were found to be extremely accurate and precise; they did not suffer from under/overprediction and had a combined 1-2% error.

With regards to the analysis of disease spread and prevalence, the study found two results of note. First, the spread of the disease did not appear to correspond with any specific holidays or events in India. The changepoint analysis tool was used to perform this analysis and changepoint dates were not found to correspond with major holidays or religious observances in India. The second major detail found through this analysis involves regions that are most vulnerable to COVID-19. It was found that rural regions felt the effects of the pandemic later but sustained higher population adjusted rates of transmission.

With the results of this study, it would be recommended that local agencies in India adopt the use of an SIR or Prophet library model to generate simple and accurate predictions to guide COVID response resource allocation. It would also be recommended that resources be diverted to rural regions in India to prevent sustained transmission in those areas. These recommendations are extremely relevant considering the COVID vaccine rollout and should be used as a tool with the vaccine to control COVID-19.

Key Words: COVID-19, modeling, forecasting, ARIMA, exponential smoothing, SIR, Prophet, LSTM

1. Introduction and Motivations

In the current COVID-19 pandemic, modeling and understanding the behavior of the virus within a region is crucial to mounting an effective response. Even a basic understanding of the factors that contribute to the pandemic can be used by local authorities to inform their decisions and act to combat the pandemic. Important factors include trends in the virus's spread with regards to age, health, geography, and numerous other factors. In addition to understanding the transmission tendencies of the disease, accurate forecasts are also crucial to a region's ability to respond to pandemic conditions. Such forecasts allow health authorities to plan testing regimens, equip medical centers, and make informed public policy decisions. The pandemic itself is also personal to me as my family and close friends in India have been adversely affected, with notable decreases in quality of life, access to healthcare and stark differences in the severity of those conditions. It is this incredible variation of conditions within India that drove the exploration into this topic. It is imperative to understand why the disease transmits the way it does and to accurately forecast based on this insight. Being able to accomplish this task will allow for an effective response to the pandemic regardless of regional transmission variability.

In this study, the early onset of the pandemic within various states in India was studied and compared to general trends from the USA and Europe, as reported in published literature. I sought to understand the relationship between the geographic makeup of a populace and viral spread with regards to states in India. The states that were selected for analysis have different geographic locations and different rural/urban population proportions to help with this goal. A Susceptible-Infected-Recovered (SIR) model was used alongside a Prophet linear model to analyze the transmission mechanics of the disease. These models rely on multiple input variables to generate a trend and accurately model the COVID-19 outbreak. These models give information pertaining to how fast the disease is transmitting and when the transmission rate changes. The SIR model and Prophet model's predictive capabilities were also tested and compared against the simpler Exponential Smoothing (ES) model and AutoRegressive-Integrated-Moving-Average (ARIMA) model, as well as an LSTM deep learning model. The forecasting abilities of the models were tested over various time

frames to see how effective each model was at predicting future case counts. Additionally, the study was conducted during the initial stages of the pandemic, granting key insights into data quality and practices.

2. Data Analysis and Manipulation

A. Exploratory Data Analysis

In this study five principal models were utilized to generate forecasts while two of those five models were used to additionally study the mechanics of the outbreak. The main factor analyzed was the transmission of the disease with respect to the rural/urban population breakdown of the state being studied. The *ExploreData* package in R was used to perform an automated EDA (in conjunction with manual analysis) and to produce visual representations of the condition of the data.

The Indian COVID-19 outbreak data that were used to train the models was obtained through the covidregionaldata R dataset found on CRAN. The dataset sorts global COVID-19 data by administrative levels, sorting by country first and then by level 1 and level 2 regions [1]. The set contained numerous observation types for each administrative level with the ones relevant to this study being the total cumulative case count (listed as "cases total), the cumulative death count (listed as "deaths total") and cumulative recovered count (listed as "recovered total"). These observation types were chosen as they were the ones required to construct the models. It should be noted that the dataset is incomplete with regards to numerous data points and levels of administrative organization. Discrepancies found through a preliminary EDA, include the lack of level 2 administrative region data for countries like India, lack of hospitalization and testing counts across various countries, and the lack of total cumulative recovery count for cases within the United States. These findings are visually summarized in Figures 1-3. Additionally, the dataset had a large number of values that were either 0 or 1 at timepoints before the beginning of the widespread outbreak in India.

The other dataset utilized in this study is the Google Human Mobility report data, released on the 8th of August 2020. This dataset was specifically needed by the Prophet model setup used by this study to function as an additional regressor [2,3]. This dataset contains location data trends for numerous geographic location categories including retail space visits, workspace visits, park spaces visits and others [2]. The main categories used for this project include workspace visits (listed as "percent_work_change") and retail space visits (listed as "percent_retail_change"). These two observation types were chosen due to the impact of the COVID-19 outbreak on them; this is illustrated in Figure 4. These two factors demonstrated the largest changes from normal in terms of visit activity when compared to other location types within a state.

Additionally, population forecasts and demographic breakdowns calculated with Indian census data were obtained for the various states that were studied. These were obtained from an Indian Government census outlet and used to fit the SIR model [4].

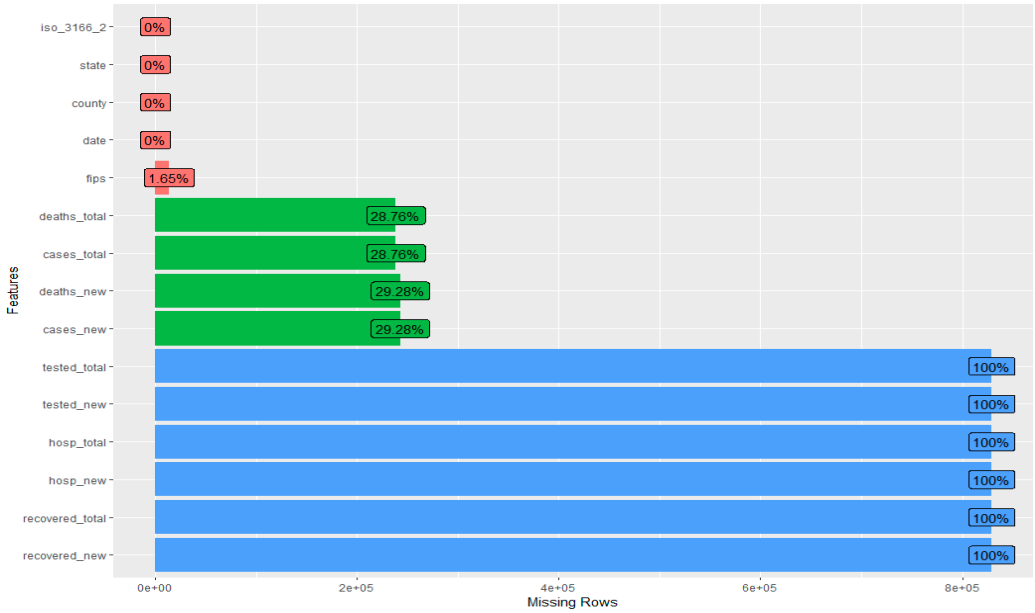

Figure 2: Missing Data Overview. *The plot_missing function from the* ExploreData *package was used to scan for missing data. The function was run on a characteristic two administrative level nation (USA); data with two administrative levels can be organized by county or state and is characteristic of most developed countries' datasets. As shown above, all testing data and hospitalization data (tested_total, tested_new, hosp_total, hosp_new) is missing, rendering the columns unusable. In the case of the USA, recovery data is also missing. Additionally, death and case data are partially missing.*

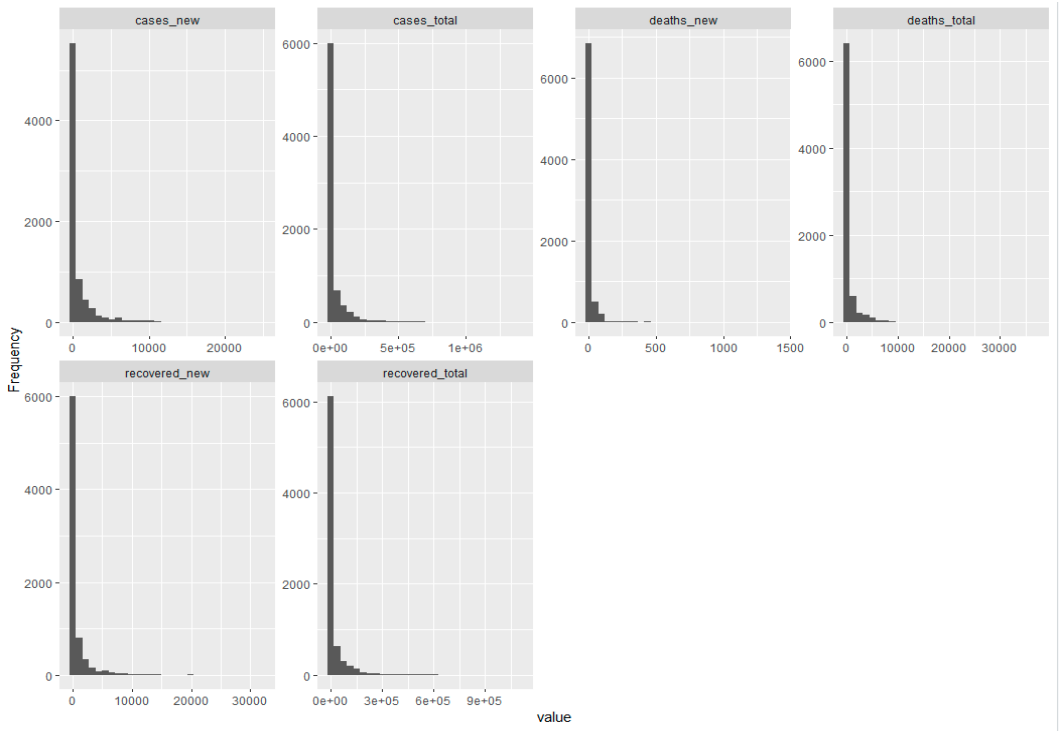

Figure 1: Value Distributions in the Data. *The plot_histogram() function was used to visualize the frequency of values reported in the data. Many 0 values are reported in the data for all observation types.*

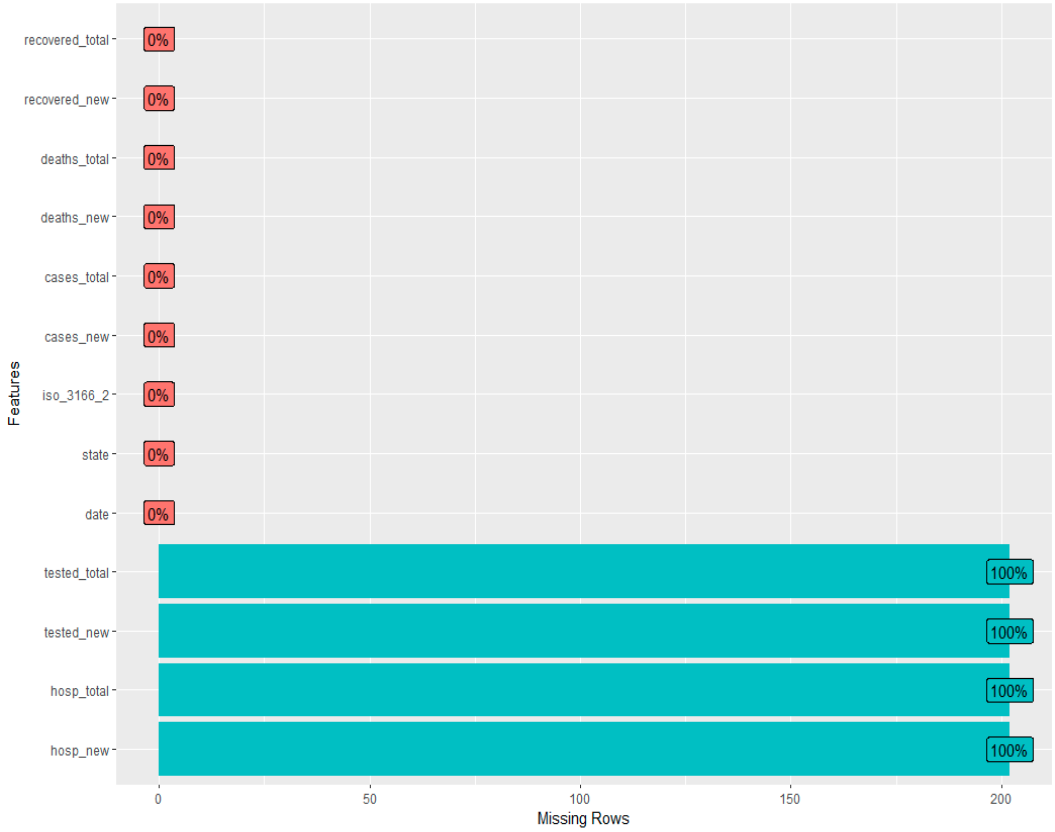

Figure 3: Missing Data Overview for India. *The plot_missing function was run on Indian subsection of the dataset; this country lacked level 2 administrative regions and only organized data by states within the country. As shown above, all testing data and hospitalization data (tested_total, tested_new, hosp_total, hosp_new) is missing, rendering the columns unusable. Additionally, the patchwork nature of data at the onset of the outbreak for most countries provided calls into question the validity of the first few data points in*

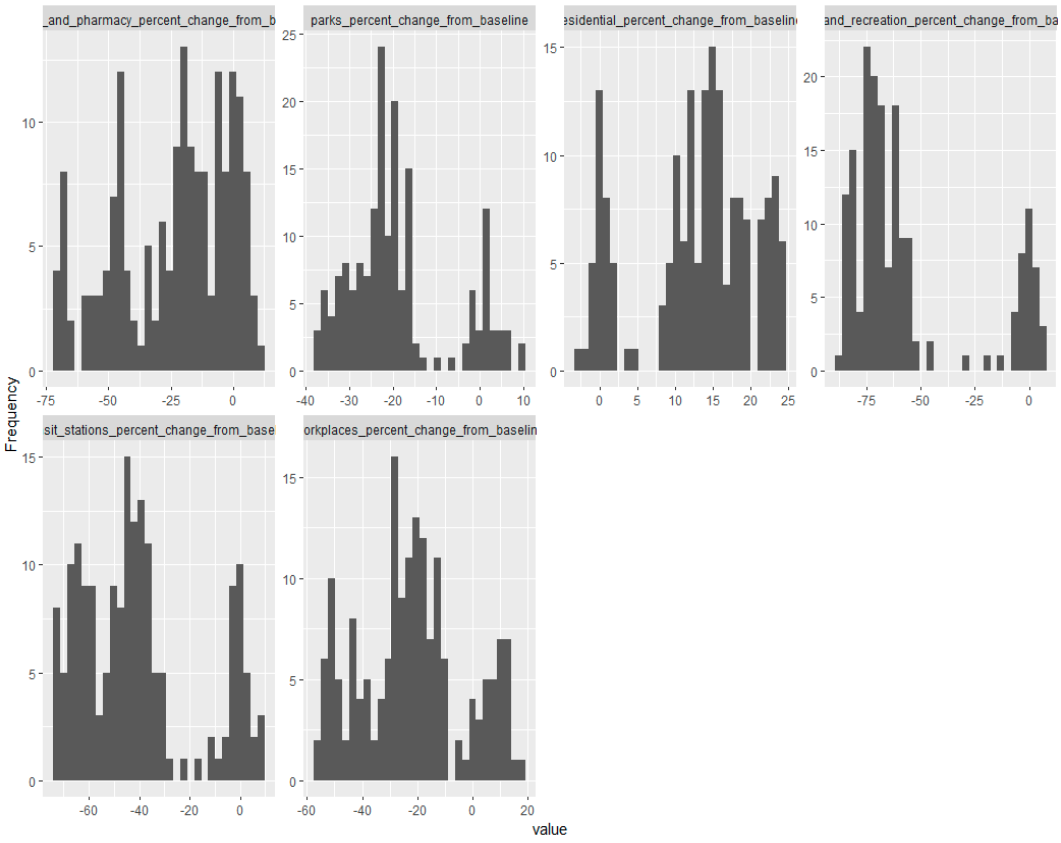

Figure 4: Frequency of Values in Mobility Report: *The above output of plot_histogram shows the frequencies of various values in the Google Human Mobility Report for the state of Assam, India. The visualization shows that retail/recreation, workplace, and transit station data columns all have a high frequency of negative values pointing to sustained, negative mobility in those spaces throughout the*

B. Data Cleaning

With both data sets obtained, the first activity done on the data was cleaning and trimming the data. The covidregionaldata and Google Human Mobility Report datasets were uploaded into their respective R scripts either from a .csv file or as a package. Both sets had the starting and ending observations trimmed off to match in size and to remove unusable data. The first 40 days of data for the outbreak were clipped due to the questionable quality of the data and its lack of use for calculating model parameters during an outbreak (values were all 0). Deficiencies in reporting by the Indian government of historical case data, especially before the month of May 2020 are pointed out by Vasudevan et. al. in their data quality and reporting study [5]. Due to these factors, the earlier data points were removed. This resulted in the clipping of the disease outbreak data and mobility data up until April 22nd, 2020. Additionally, the Google Human Mobility Report is released monthly, with the report used for this study containing data up until August 7th, 2020. As a result, the end of the disease outbreak data must be trimmed back to August 7th, 2020. With this trimming completed, the data used for every model had a start date of April 22nd, 2020 and end date of August 7th, 2020, yielding 108 usable days of data.

3. Models and Methodology

A. Overview

The models utilized in the study the Susceptible-Infected-Recovered (SIR) model, a customized Prophet model, an Exponential Smoothing (ES) forecast model, an AutoRegressive-Integrated-Moving-Average (ARIMA) model, and an LSTM deep learning model. Of these models, all five were used to generate and test forecasts to predict COVID-19 case counts on a 7-day basis. Only two of the models were used to quantitatively analyze the transmission trends of the COVID-19 pandemic; these models were the SIR model and the Prophet model. These two models were useful for extracting insights into the transmission mechanics of the virus as well as timepoints where the underlying transmission mechanics changed. An abbreviated glossary defining key terms for each model can be found at the end of the section. Additionally, each set of models came with slightly different visualization schemes, all based on the ggplot library. To discuss all models equally, they were compared on the basis of their actual prediction (trend) and the 95% confidence interval for the prediction; most of which are shared across all models.

B. Input and Output Variables

Over the course of the modeling process, three specific input variables were used across the different models and specific outputs were collected. These relationships are listed below.

Inputs

covidregionaldata Dataset	Case count data used in all models. Recovery and Death data used additionally in the SIR Model
Mobility Data	Google Human Mobility data used in the Prophet Model
Population Data	Indian Census Population data used in the SIR Model

Outputs

7-day Forecast	7-day case count forecast collected for all models. Error is also collected.
β	Implied transmission coefficient collected from SIR Model. Covered in sections E and G.
Changepoint Chart	Changepoint vs. time charts collected from the Prophet model. Covered in sections F and G.

C. Exponential Smoothing and ARIMA Models

The most rudimentary models that were used in this study include the ES and ARIMA models. Both models, in the context of this study, are general curve fitting models, fitted with univariate time series data, that can be used to generate predictions.

The ES model utilized in the study was obtained from the *stats* package and was implemented through the HoltWinters() function. This function allows the fitting of a triple exponential smoothing model [6]. Exponential smoothing models are based on the principle of weighting the effect of past data less than the effect of recent data using the exponential window function[7]. The most rudimentary of these models can make a prediction without a trend or a seasonal component, relying only on the smoothing effect of past observations. Triple exponential smoothing models allow for a model that can smooth the effect of past observations, track a trend, and track seasonality (Figure 5) [7]. The specific implementation of the model that was used relied mainly on the smoothing parameter and trend parameter, foregoing the seasonality option. When training the model, the HoltWinters() function automatically assigns values for the smoothing and trend parameters based on the data [6,8]. The HoltWinters() function allows for user input start values for the trend and smoothing parameters, these values were left blank and the function was allowed to find values on its own. The data that were used for training the model were the case count data spanning 140 days. These data were converted from a tibble to a time series (.ts) object to be accepted by the HoltWinters() function [6]. The generated model was then fed into the forecast() function from the *forecast* package to generate a 7-day prediction. The prediction was automatically computed with 80% and 95% confidence intervals.

$$\begin{aligned}\hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t+h-m(k+1)} \\ \ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \\ s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}\end{aligned}$$

Figure 5: HoltWinters Additive Triple Exponential Smoothing. *The HoltWinters() function used in the ES model relies on the above equations where l_t accounts for smoothed constant trend data, b_t accounts for a smoothed non-constant trend, s_t accounts for seasonal factors, and $\hat{y}$ is the total of all three factors. Since the specific model used in this study omitted seasonality, s_t remained unused [9].*

The ARIMA model is commonly used as an alternative to the ES model for univariate time series forecasting. The specific ARIMA model that was used in the study was implemented through the auto.arima() function from the *forecast* package. ARIMA models are a family of univariate models that use the data as their own regressor (AR), replace the observations with differences of observations (I), and whose regression error is a linear combination of previous error terms (MA), all to varying degrees [10]. Within the model, the parameters that dictate the degree of influence of the AR, I, and MA portions of the model are denoted by p, d, and q respectively. These parameters are automatically configured by the auto.arima() function as it trains itself on the supplied data [11]. The d value was checked manually to make sure it was appropriate as exponentially shaped data requires double differencing (d = 2) which was applicable to the data used. The data that were fed into the auto.arima() function for training were converted from a tibble to a .ts object so the function

would accept it. The model allows for p, q, and d values to be specified by the user, these were left unfilled, and the q values were instead checked when the models were generated [10]. The model generated by the auto.arima() function was fed into the forecast() function to generate a 7-day prediction with 80% and 95% confidence intervals.

D. LSTM Model

The last strictly univariate model that was utilized in this study was an LSTM deep learning model. This neural network model is characterized by functional subunits called cells which can maintain a memory state. This information is controlled input gates which control the flow of information in, output gates which control the flow of information out, and forget gates which control the information that gets deleted (Figure 6). These cells are arranged in layers which can be customized to hold different numbers of cells [12]. The model used in this study arranged the cells into two layers composed of 32 cells each followed a dense layer. The dense layer is a fully connected layer that follows the LSTM recurrent layers and aids in outputting a prediction. Before training, the data were formatted such that the model would be able to use the it. The data were lagged by three days (to favor recency and newer observations), differenced, and rescaled to fit the range [-1,1] (this is the range of the sigmoid function) [12]. After modification, the input set was reshaped into a 3-dimensional input and fed into the model for training. The model was set up using the LSTM tools found in the *keras* library. The data were trained for 12 epochs for the best chance at a usable model; the training duration was determined with preliminary testing as training for less than 12 epochs was found to underfit and training more than 12 led to overfitting. The model was then fed into a customized prediction function, built around the predict() function, that utilized the last known value in the training data as a basis for the next prediction. This predicted value then had its differencing and scaling reversed to make it comparable to actual case count values.

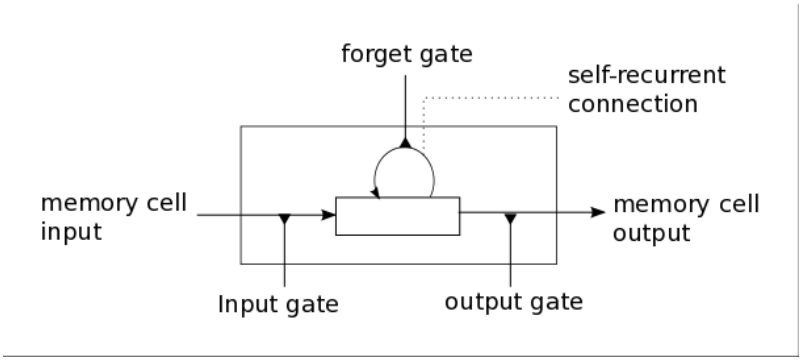

Figure 6: Characteristic LSTM Memory Cell: *The characteristic LSTM memory cell is similar to a conventional recurrent cell but also contains a forget gate, allowing the block to forget information, changing the cell state and what the output can be [13].*

E. SIR Model

The first of the multi-input models that were utilized was the SIR model. This model is used to model three populations during an epidemic situation; a susceptible (S) population, infected (I) population, and recovered/dead (R) population. This model setup assumes that the infected population gains immunity either permanently or longer than the epidemic lasts. The three populations are tracked via three differential equations that track the changes (dS, dI, dR) in the respective populations. The equations rely on the values of the S, I, R populations as well as the implied transmission coefficient β and implied recovery coefficient γ (Figure 7) [14]. These values, within the context of the model, are dependent on the characteristics of the disease itself as well as the conditions in which it is allowed to spread. β dictates how quickly the disease spreads and how the S and I populations change. γ dictates how quickly individuals recover or die, directing how the I and R populations change [14, 15]. It should be noted that these constants are population adjusted and the β constant describes a person-to-person transmission rate. As a result, if an outbreak is simulated and β is held constant, logistic growth in total case counts will still occur since the actual case counts depend on β *and* population. Changing β changes the nature of the growth. The custom model implemented in this study is a discrete time model, solving the differential equations using the training data, over 1-day intervals to create a table of S, I, and R values as well as dS, dI, dR values and β/γ values. The model then averages the last 7 β values and last 7 γ values to create averages that are used as a basis for prediction. These are then applied to the last known S/I/R and dS/dI/dR values to generate the first predicted values. These values are then re-applied to generate the next day, repeating until a 7-day prediction is generated. The predictions then have a 95% confidence interval generated for them using a two-sided t-test method with a sample size of 7 (number of β/γ used in prediction). Additionally, the β values could be extracted for analysis [15,16].

$$\frac{dS}{dt} = -\frac{\beta IS}{N},$$

$$\frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I,$$

$$\frac{dR}{dt} = \gamma I,$$

Figure 7: SIR Model Equations. *The SIR model tracks three populations (**S**usceptible, **I**nfected, and **R**ecovered) through three ordinary differential equations. The additional parameters β (transmission coefficient) and γ (recovery coefficient) account for the physical characteristics of the disease. β modulates the rate of infection and γ modulates the rate of recovery and death [17].*

F. Prophet Library Model

The last model utilized in this study was a custom setup Prophet model, implemented through the *Prophet* library. To set up this multivariate model, additional regressor functionality was enabled through the add_regressor() function and set to an additive type regressor (Figures 8,9) [3,18]. Changepoint functionality was also taken advantage of; referring to the model's ability to detect significant changes in a trend, mark them in relation to time and use them guide modeling of the trend. The model's changepoint detection time frame and changepoint response were modified to make the model more responsive to recent trend shifts; this was accomplished with the changepoint.range and changepoint.prior.scale parameters [19]. Then the additional regressor dataset was constructed. In the case of this model, the cleaned Google Mobility Report data were used.

The last seven days of the mobility report were taken and extrapolated in a repeated manner to match the number of days being predicted; these values were taken and appended to the original table of values. This was done due to the Prophet library requiring the values of extra regressors to be known in the future, within the prediction time frame [20,21,22]. The model was then trained using the cleaned case count data and original, cleaned mobility data. The prediction was then obtained using the predict() function with the appended mobility dataset (contains past and future values) as an extra parameter. Additionally, the changepoint detection mechanics were run on the model to find when in time significant trend shifts occurred (significant enough to influence the model) [23].

$$g(t) = (k + \mathbf{a}(t)^\intercal \boldsymbol{\delta})t + (m + \mathbf{a}(t)^\intercal \boldsymbol{\gamma}),$$

Figure 9: Core Linear Model in Prophet. *The model shown above is the core model used by Prophet to calculate a trend and forecasts; this version of the model was utilized by the study. g(t) represents the overall composite trend while k represents the base trend and $a(t)^T\delta$ accounts for accumulated trend shifts over time. Additionally, m represents the offset parameter calculated for the model while $a(t)^T\gamma$ modulates offset changes throughout the model [18].*

$$h(t) = Z(t)\boldsymbol{\kappa}.$$

Figure 8: Holiday and Additional Regressors Functionality in Prophet. *Prophet uses the above function to model and store the effects of holidays and additional regressors. Z(t) represents a matrix of holidays/regressors and their dates of occurrence in the past and future The κ parameter stores the effects of each occurrence of each holiday or regressor value. Given a time value, the holiday occurrences are used to track the effect of each occurrence through κ and summed up to generate h(t). h(t) is added to g(t) (Figure 8) to generate the overall trend [3,18].*

G. Section Notation

ES	HoltWinters Triple **E**xponential **S**moothing Model
ARIMA	**A**uto**R**egressive **I**ntegrated **M**oving-**A**verage Model
LSTM	**L**ong **S**hort-**T**erm **M**emory Model
SIR	**S**usceptible-**I**nfected-**R**ecovered Model
Prophet	Prophet Linear Model with Additive Seasonality, Holidays, and Regressors from the Prophet Library
β	Transmission Coefficient: Unitless constant used to modulate daily infection rate in SIR model
γ	Recovery Coefficient: Unitless constant used to modulate daily recovery and death rate in SIR model
p	Lag Parameter: Defines the number of lags in days used by the AutoRegressive portion of the ARIMA model

d	Differencing Parameter: Defines the number of times the model data was differenced.
q	Moving Average Order: Defines the order of the Moving Average portion of the model.

4. Results

The study sought to test the modeling and predictive capabilities of various models. The 5 Indian states tracked through this study included Maharashtra, Uttar Pradesh, Tamil Nadu, Rajasthan, and Assam. The SIR and Prophet models were also used to analyze the transmission behavior of the virus with relation to geography and time.

The SIR models were run on each state and had their β values extracted for analysis. Each state had a large variance in their distribution of β values from the SIR model, pointing to discrepancies in data collection at the state level. As shown in the table below, the average β values calculated for all the states generally seemed to correspond to a larger rural population fraction (for all states except Uttar Pradesh) and a lower absolute hospital bed count. Additionally, states with internationally connected urban centers (Chennai, Tamil Nadu and Mumbai, Maharashtra) tended to have higher average initial β values (during the first 7 days studied) but lower average final β values (over the final 7 days studied). These locations are also known to be the initial outbreak centers in India. Plotting β values against time for each state reflects this observation and individual values are noted in the table below and Figures 10-14.

*- Indicates value was calculated including missing/zero data points

State	Urban/Rural Population % [4]	Total Hospital Bed Count [24]	Average β	Average Initial β	Average Final β
Maharashtra	45.22%/54.78%	231,739	0.082	0.095	0.066
Uttar Pradesh	77.73%/22.27%	281,402	0.114	0.066	0.102
Tamil Nadu	48.4%/51.60%	155,375	0.117	0.086	0.100
Rajasthan	24.87%/75.13%	93,176	0.117	0.052	0.092
Assam	14.10%/85.90%	24,178	0.160	0.038*	0.169

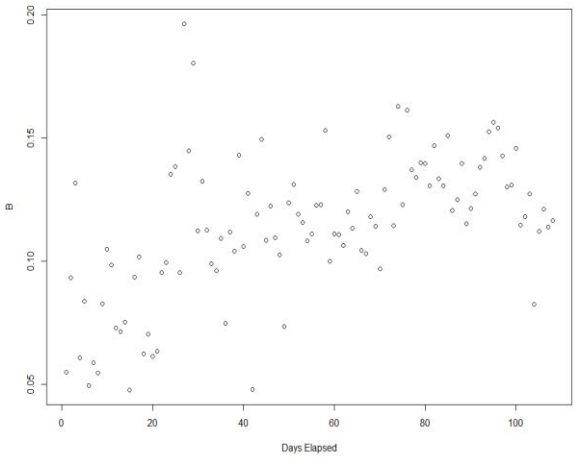

Figure 12: β Values for Uttar Pradesh. *Uttar Pradesh is one of the hardest hit states in the pandemic and holds a large hotspot, Delhi. The data for U.P. shows a large amount of variation and an initially increasing trend that later plateaus. This implies difficulties in data collection and some degree of success in reducing the speed at which COVID-19 is spreading within the state.*

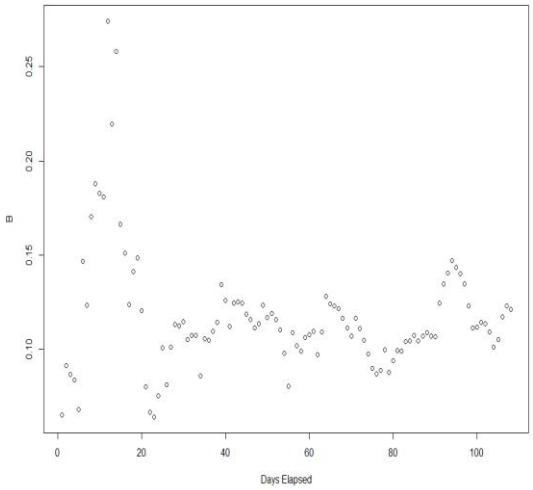

Figure 11: β Values for Tamil Nadu. *Tamil Nadu deviates from the other states in the β values that were calculated for it. It displays a relatively low amount for variation in the observations for most of the time poinsts and shows an oscillating by mostly unchanging trend.*

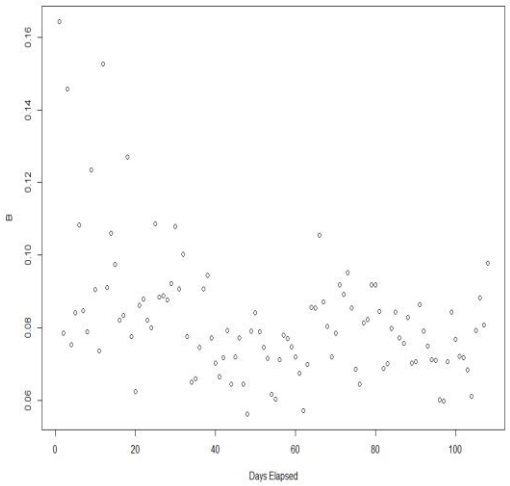

Figure 10: β Values for Maharashtra. *Shown above are the implied β values for Maharashtra. One notable characteristic of the data for this state includes the quick downward trend in values, implying measures being taken to control the outbreak Additionally, there is a large spread in data that gradually decreases as time passes, implying improvements in data collection.*

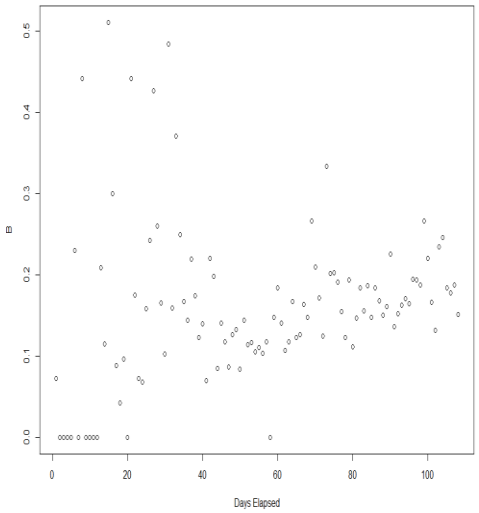

Figure 13: β Values for Assam. *Shown above are the implied β values for the state of Assam. Notice a large amount of variation during the initial stages of the pandemic and a smaller variation as more data is collected. It is thought that the large variation in β values, especially in the earlier data points is due to poor data collection. Additionally, a gradual upward trend can be noticed.*

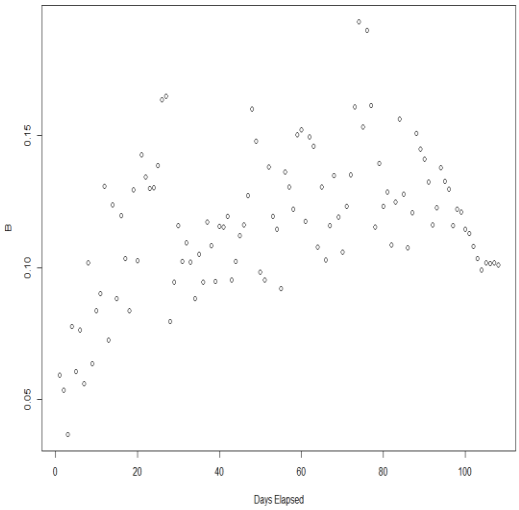

Figure 14: β Values for Rajasthan. *The β values calculated for Rajasthan display a gradual upward trend followed by a sharp decrease. Additionally, the variation in the data stays relatively constant throughout the observations and sharply decreases in the most recent data points. Overall, the β values appear to be approaching a plateau.*

The Prophet model found that holidays had a minor effect on accelerating the spread of COVID-19, likely due to widespread lockdowns in India. The Prophet changepoint tool found trend accelerations that were noted throughout the course of the model were largely uniformly spaced throughout the time frame observed during the study (Figures 15-19). Areas with large quantities of changepoints are areas where case counts are increase or decreasing in a non-linear manner. Areas devoid of changepoints signal constant trends: either constant growth, decline, or no change in case counts. Furthermore, it is unlikely that large events had an impact on the trend due to the lockdown status of the country as noted through official government records and disclosures; the changepoints were found to not correspond to any specific dates or holidays.

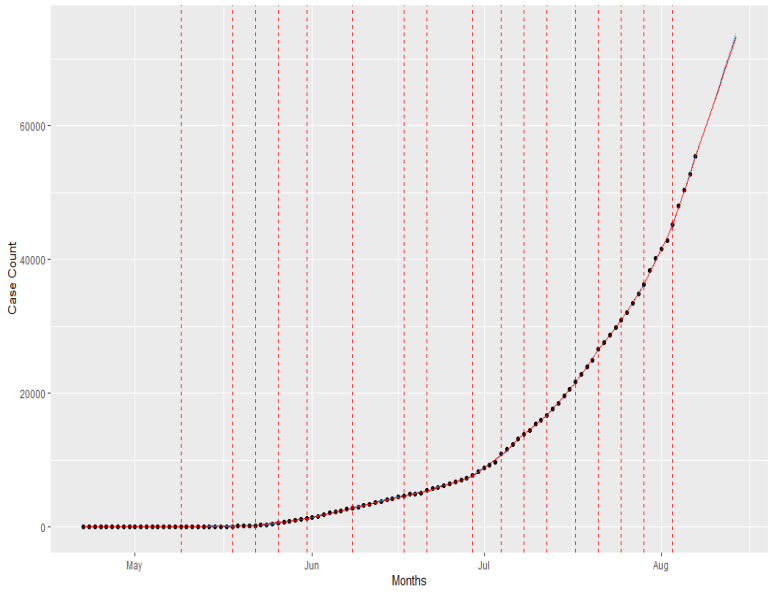

Figure 15: Changepoints and General Trend for Assam. *Changepoints (red) are shown superimposed on top of the trendline (black). The changepoints correspond to timepoints where the model detects a trendshift. The changepoints here are slightly more frequent in the recent data points. Observing the trend, the increased changepoints signal a slight acceleration in COVID-19 spread.*

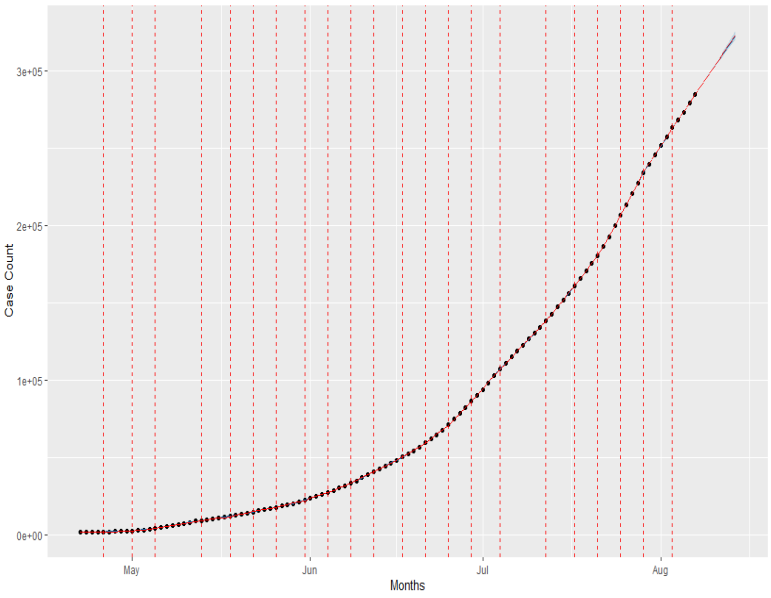

Figure 16: Changepoints and General Trend for Tamil Nadu. *In the case of Tamil Nadu, the changepoints are largely evenly spread (with some gaps). Due to the nature of exponential growth in cases seen during the early stages of a pandemic, the constant distribution of changepoints shows a more or less unchanging situation in the population adjusted rate of spread of COVID-19.*

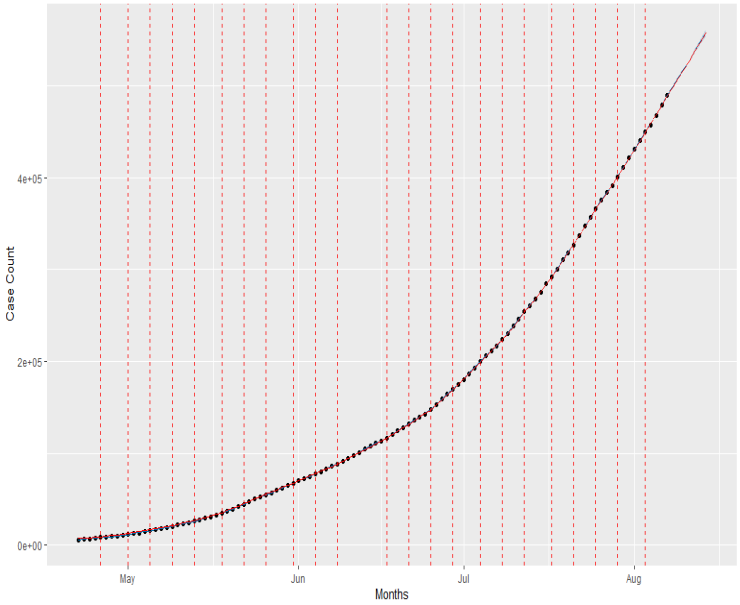

Figure 17: Changepoints and General Trend for Maharashtra. *The changepoints noted for data from Maharashtra are similar in nature to those from Tamil Nadu. The changepoints are largely evenly distributed throughout the timepoints with occasional gaps.*

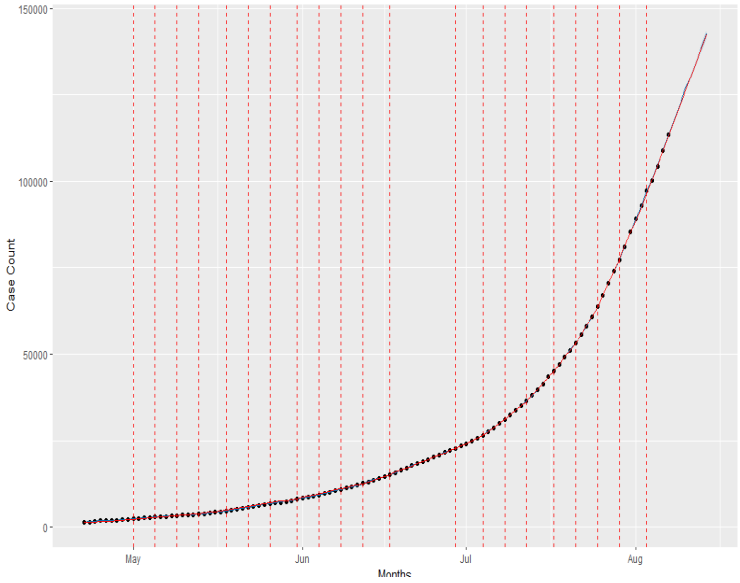

Figure 18: Changepoints and General Trend for Uttar Pradesh. *The changepoint chart is similar to those of Tamil Nadu and Maharashtra. Additionally, the data from Uttar Pradesh reflects a similarly timed 'changepoint void' in the month of June.*

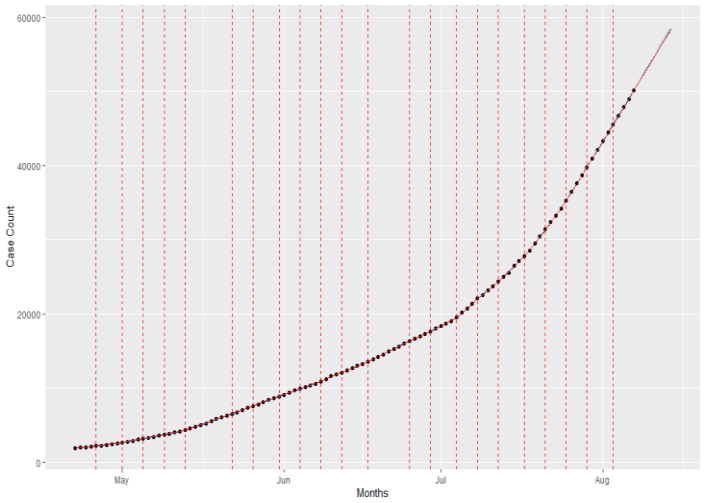

Figure 19:Changepoints and General Trend for Rajasthan. *Changepoints here, like in previous states, are distributed evenly throughout most of the timepoints. The cumulative case count trend for this state lacks the large changepoint graph in June and is similar to the data from Tamil Nadu.*

All 5 of the models were then run on the 5 states' data to test their predictive capabilities over 7 days. These forecasts were stored for one week until the next data set update and compared to newly included values as well as values reported through public media outlets [1,25]. The best performing models were the SIR and Prophet model. These models averaged 1%-2% error (MAPE) with the Prophet model edging out the SIR model by 0.1%-0.2% error each time. These models also did not display routine over or under prediction, calculating predicted values above and below actual values. Additionally, the SIR and Prophet models possessed the tightest prediction intervals when compared to other models (Figures 20,21). The worst performing models were the Arima, ES, and LSTM models. The ES and ARIMA models both averaged 5%-10% error when attempting a 7-day prediction. Additionally, both models had routine trend underprediction and had trouble adjusting to trend-shifts. The LSTM model was the worst performer and highly inconsistent. Training the model multiple times results in the model having wildly varying performance, sometimes underpredicting by 25%-30%. The LSTM model usually underpredicts but on the rare occasion, produces a model with ~5% error. All three of the lowest performing models suffered from large prediction intervals, pointing to a low confidence in prediction (Figures 22-24). The prediction intervals displayed in the figures below are of the state of Assam although the stated conclusions and results apply to data from all states.

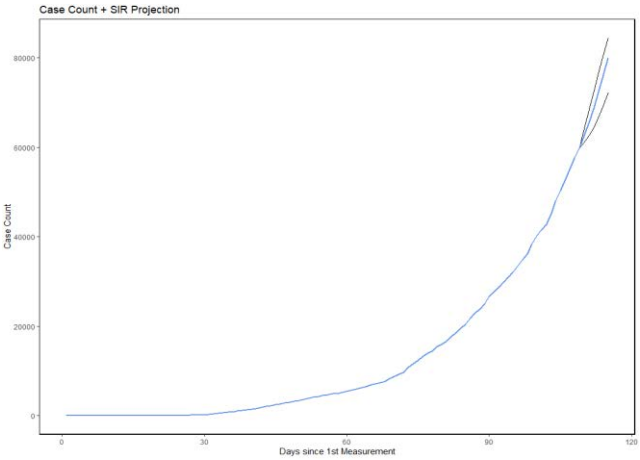

Figure 20: Assam, Case Count Projection, SIR Model. *The figure depicts the 107 days of trend data the model was trained on with a 7-day prediction appended on it. The trendline is displayed in blue with the prediction starting where the black lines start. The black lines depict the 95% confidence interval of the prediction. Unlike the Arima and ES models, this model depicts continued exponential growth and does not plateau.*

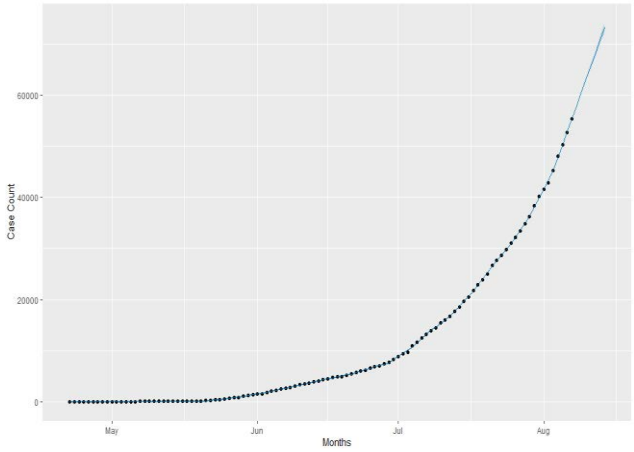

Figure 21: Assam, Case Count Projection, Prophet Model. *The figure depicts the 107 days of trend data and an appended 7-day prediction. The trend is depicted in blue with data points in black. The 95% confidence interval is shown as a dark gray space around the trend line. The model predicts similar values into the future as the SIR model but does so with a tighter confidence interval.*

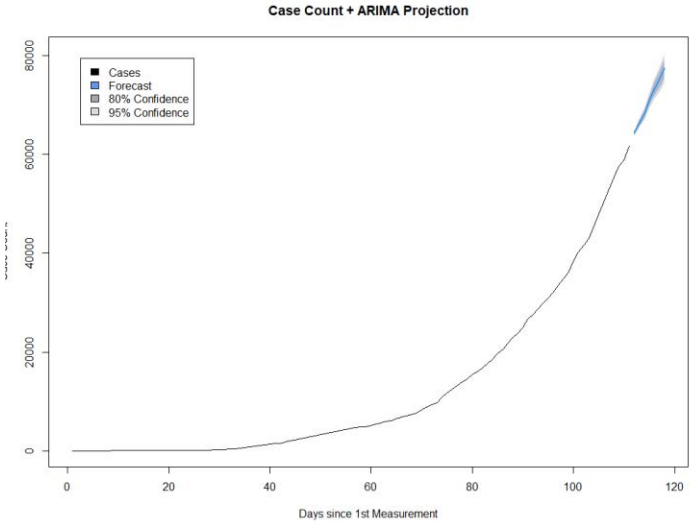

Figure 22: Assam, Case Count Projection, ARIMA Model. *The figure shows the output of the forecast package ARIMA model. The original trend is depicted as a black line with the prediction showing as a blue line. The 80%/95% confidence interval are shown in various shades of gray. The model, although more confident than the SIR model, suffers from routine underprediction with the model predicting a trend closer to a plateau.*

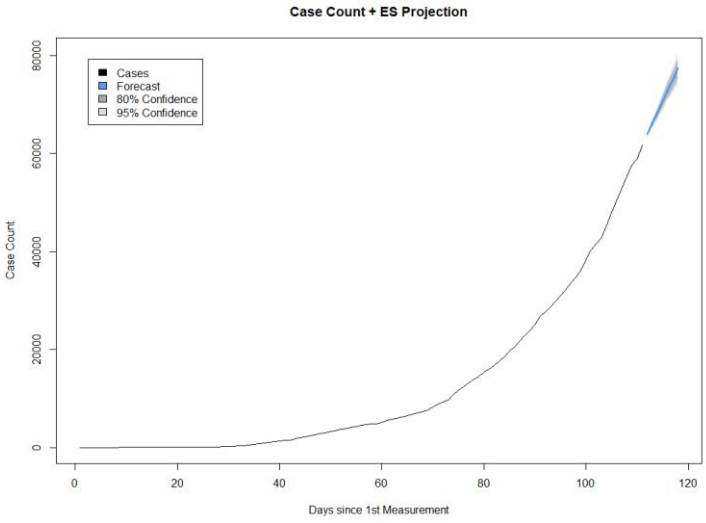

Figure 23: Assam, Case Count Projection, ES Model. *The figure shows the output of the stats package ES model. This model performs similarly to the ARIMA model and shows results in the same manner. Additionally, the model suffers from the same routine underprediction and plateau like prediction.*

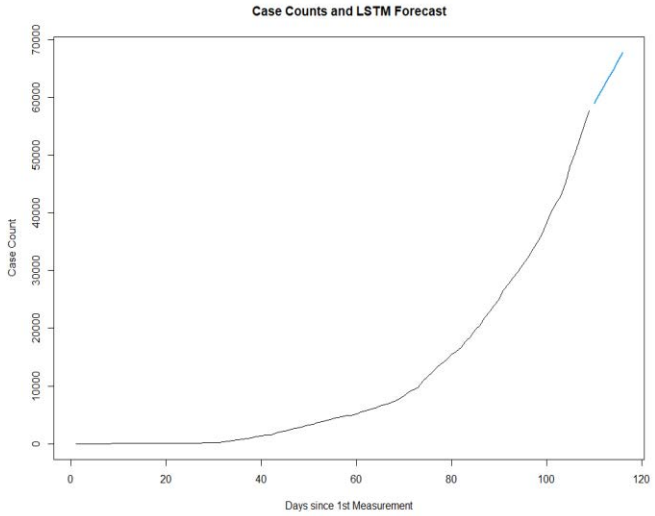

Figure 24:Assam, Case Count Projection, LSTM. *The LSTM model output is displayed. This model was the weakest of the five for this use case, both in its features and predictive ability. The model suffered from overfitting and underfitting, a consequence of the number of data points available to it. Due to the non-deterministic nature of the model, each time the model is trained/ran, a different result is obtained. The model suffers from routine underprediction and fails to deliver a confidence interval.*

5. Discussion and Conclusions

COVID-19 pandemic trends around the developed world have all followed similar patterns with regards to geography. In the United States, S.J. Goetz et. al. found that rural regions lagged behind urban regions in COVID-19 cases per capita, noting that once a rural region was compromised however, the virus spread through those communities much quicker [26]. Findings in Europe echo these patterns with I. Kashnitsky and J.M. Aburto finding that COVID-19, although transmitted more rapidly in urban regions, is felt more strongly in aging rural areas [27]. They contend that the next high-risk locations in Europe, with regards to COVID-19 transmission, are peripheral regions with aging populations and lower healthcare availability.

The findings of this study echo a similar conclusion, that although urban centers are usually hit first, rural regions risk bearing an unequal portion of COVID-19 mortality and infection in proportion to their population. Rural states like Assam had a delayed onset of pandemic conditions but ended up with higher transmission rates when compared to more urban states like Maharashtra. Urban center access in India is linked to healthcare access and positive health outcomes; these regions usually possess more government health centers due to their dense population but also possess numerous private healthcare institutions, a luxury rural regions do not have [24,28,29]. All states evaluated in this study followed the trend where rurality was associated with worse COVID-19 transmission scenarios, except Uttar Pradesh. However, Uttar Pradesh exists as somewhat of an outlier, as despite its relatively large rural population fraction (when compared to Maharashtra), Uttar Pradesh possesses an excess of healthcare centers which serve to equalize health access disparities and lessen the effect of rural living conditions [4,28]. Additionally, health access (measured through hospital bed count) was also correlated with better outcomes, as states with more extensive healthcare networks had better lower transmission rates per capita. Uttar Pradesh especially seems to reflect this conclusion as its rural population fraction suggests its transmission rates should be much higher and its hospital bed count suggests its transmission rates should be lower, yet the net effect on transmission rates lies somewhere in between.

These findings suggest that rural regions in India are more likely to have heightened COVID-19 transmission rates and bear worse per capita outcomes than their urban

counterparts. As data becomes more widely available, it is suggested that more granular district level data within the states in India are used to confirm this finding. This would allow state government policy and decision making to be ruled out as a confounding variable. Additionally, forecasts like the ones presented through the Prophet and SIR models should be used by state and local authorities to manage their resources and effectively control the outbreak. Having knowledge of where disease conditions in a region may be headed are crucial to mounting an effective response. In conjunction with vaccination efforts, the modeling strategies presented above could be used to locate high risk areas so that they may be targeted by public health agencies for vaccination and heightened vigilance. These modeling strategies can also be expanded out of India to assess COVID threats in other countries. Although specific conditions may differ between countries, the models' generality (ex. Condensation of transmission mechanics into a single β value, Prophet changepoint detector) allows them to be used in a wide range of geographic environments and still generate useful transmission insights.

The work done herein forms a basis for future expansion and exploration. The models covered within the study are only a select subset of those available and represent the most accessible and understandable. Further work should be done to bring more complex models-including newer machine learning models-in a user friendly and accessible manner to facilitate accurate forecasting of COVID-19 case counts. Advancements in modeling would also allow for a better comprehension of the underlying transmission mechanics of COVID-19 and would facilitate the understanding of why the disease spreads the way it does given a specific locale.

6. References

[1] Abbot, S., Sherrat, K., Bevan, J., Gibbs, H., Meakin, S., Hellewell, J., Munday, J., Barks, P., Campbell, P., Finger, F., & Boyes, R. (2020). *Subnational Data for the Covid-19 Outbreak*. Epiforecasts.io. https://epiforecasts.io/covidregionaldata/

[2] Google. (2020, August). *COVID-19 Community Mobility Reports*. https://www.google.com/covid19/mobility/

[3] Facebook. (2018a). *Seasonality, Holiday Effects, And Regressors*. Facebook-GitHub. https://facebook.github.io/prophet/docs/seasonality,_holiday_effects,_and_regressors.html

[4] Census 2011. (2011). *List of states with Population, Sex Ratio and Literacy Census 2011*. Population Census 2011. https://www.census2011.co.in/states.php

[5] Vasudevan, V., Gnanasekaran, A., Sankar, V., Vasudevan, S. A., & Zou, J. (2020). Disparity in the quality of COVID-19 data reporting across India. *medRxiv*, pp21–24. https://doi.org/10.1101/2020.07.19.20157248

[6] R-Core. (n.d.). *HoltWinters function | R Documentation*. R Documentation. Retrieved 2020, from https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/HoltWinters

[7] NIST SEMATECH. (2012). *NIST/SEMATECH e-Handbook of Statistical Methods*. Nist.Gov. https://www.itl.nist.gov/div898/handbook/

[8] Brownlee, J. (2020, April 12). *A Gentle Introduction to Exponential Smoothing for Time Series Forecasting in Python*. Machine Learning Mastery. https://machinelearningmastery.com/exponential-smoothing-for-time-series-forecasting-in-python/

[9] Wikipedia Contributors. (2010). *Exponential smoothing*. Wikipedia. https://en.wikipedia.org/wiki/Exponential_smoothing

[10] Coghlan, A. (2010). *Using R for Time Series Analysis — Time Series 0.2 documentation*. A Little Book of R for TimeSeries. https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html

[11] Hyndman, R. (n.d.). *auto.arima | R Documentation*. R Documentation. Retrieved 2020, from https://www.rdocumentation.org/packages/forecast/versions/8.13/topics/auto.arima

[12] Wanjohi, R. (2018, April 5). *Time Series Forecasting using LSTM in R*. Rbind.io. http://rwanjohi.rbind.io/2018/04/05/time-series-forecasting-using-lstm-in-r/

[13] Tolpygo, A. (2017, July 17). *Predicting Stock Volume with LSTM*. SFL Scientific | Data Science Consulting & AI Development Services. https://www.sflscientific.com/data-science-blog/2017/2/10/predicting-stock-volume-with-lstm

[14] Smith, D., & Moore, L. (2004, December). *The SIR Model for Spread of Disease - The Differential Equation Model | Mathematical Association of America*. Mathematical Association of America. https://www.maa.org/press/periodicals/loci/joma/the-sir-model-for-spread-of-disease-the-differential-equation-model

[15] Mahmud, A., & Lim, P. Y. (2020). *Applying the SEIR Model in Forecasting The COVID-19 Trend in Malaysia: A Preliminary Study*. *medRxiv*, pp8–16. https://doi.org/10.1101/2020.04.14.20065607

[16] Zhou, X., Ma, X., Hong, N., Su, L., Ma, Y., He, J., Jiang, H., Liu, C., Shan, G., Zhu, W., Zhang, S., & Long, Y. (2020). *Forecasting the Worldwide Spread of COVID-19 based on Logistic Model and SEIR Model*. *medRxiv*, pp5–9. https://doi.org/10.1101/2020.03.26.20044289

[17] Wikipedia Contributors. (n.d.). *Compartmental Models in Epidemiology*. Wikipedia. https://en.wikipedia.org/wiki/Compartmental_models_in_epidemiology

[18] Facebook. (2018b). *facebook/prophet*. Facebook-GitHub. https://github.com/facebook/prophet/

[19] Facebook. (2018c). *Trend Changepoints*. Facebook-GitHub. https://facebook.github.io/prophet/docs/trend_changepoints.html#automatic-changepoint-detection-in-prophet

[20] Choudhary, A. (2020, May 31). *Generate Quick and Accurate Time Series Forecasts using Facebook's Prophet (with Python & R codes)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2018/05/generate-accurate-forecasts-facebook-prophet-python-r/

[21] Fauchereau, N. (2018). *The impact of weather conditions on cycling counts in Auckland, New Zealand*. Jupyter Notebook. https://nbviewer.jupyter.org/github/nicolasfauchereau/Auckland_Cycling/blob/master/notebooks/Auckland_cycling_and_weather.ipynb

[22] Jason Lzp. (2020, December 6). *Time Series Forecasting With Prophet In R - Level Up Coding*. Gitconnected. https://levelup.gitconnected.com/time-series-forecasting-with-prophet-in-r-a9ee81dc82e1

[23] University of Virginia. (n.d.). *Predictive Analytics: Predicting and Forecasting Influenza*. Github. Retrieved August 5, 2020, from https://bioconnector.github.io/workshops/r-predictive-modeling.html#the_caret_package

[24] The Center for Disease Dynamics Economics & Policy. (2020, August 6). *COVID-19 in India: State-Wise Estimates of Current Hospital Beds, ICU Beds, and Ventilators*. Center for Disease Dynamics, Economics & Policy (CDDEP). https://cddep.org/publications/covid-19-in-india-state-wise-estimates-of-current-hospital-beds-icu-beds-and-ventilators

[25] The New York Times. (2020). *India Coronavirus Map and Case Count*. https://www.nytimes.com/interactive/2020/world/asia/india-coronavirus-cases.html

[26] Goetz, S. J., Tian, Z., Schmidt, C., & Meadowcroft, D. (2020, April 23). *Rural COVID-19 Cases Lag Urban Areas but Are Growing Much More Rapidly (Department of Agricultural Economics, Sociology, and Education)*. Penn State College of Agricultural Sciences. https://aese.psu.edu/nercrd/publications/covid-19-issues-briefs/rural-covid-19-cases-lag-urban-areas-but-are-growing-much-more-rapidly

[27] Kashnitsky, I., & Aburto, J. M. (2020). *COVID-19 in unequally ageing European regions. ResearchGate*, 1–2. https://doi.org/10.31219/osf.io/abx7s

[28] Barik, D., & Thorat, A. (2015). *Issues of Unequal Access to Public Health in India*. *Frontiers in Public Health*, *Vol.3*, pp1–3. https://doi.org/10.3389/fpubh.2015.00245

[29] Kumar, A., Rajasekharan Nayar, K., & Koya, S. F. (2020). COVID-19: *Challenges and its consequences for rural health care in India*. *Public Health in Practice*, *Vol.1*, pp1–2. https://doi.org/10.1016/j.puhip.2020.100009

7. Supplemental Material

1. Code (GitHub Repository):

https://github.com/AbhiBuab/COVID19-Modeling-Forecasting/

2. Raw Output Figure Library (Google Drive Folder):

https://drive.google.com/drive/folders/1SqdSVhtX97dK6WnGC8K6x6205fq9oLDD?

8. Acknowledgements

I would like to thank Dr. Veena Mendiratta (formerly of Nokia Bell Labs and currently Adjunct Professor, Northwestern University), for providing her mentorship on this project. She assisted me in learning R, aided me in conducting research, and provided feedback so that I could improve my analysis. She also aided in the editing and proofreading process. Without her insights and mentoring, this would have been a much rockier learning journey.