CS551 Deep Learning
Statistical vs. Neural Time Series Modelling: COVID-19 Pandemic

Shikhar Jaiswal
1601CS44
Department of Computer Science and Engineering
Indian Institute of Technology Patna
shikhar.cs16@iitp.ac.in

Abstract

Time series analysis is one of the most prolific domains of study in the world of machine learning, due to its wide area of application and inference. We employ the use of two class of models, namely **A**uto-**R**egressive **I**ntegrated **M**oving **A**verage (**ARIMA**), and **G**ated **R**ecurrent **U**nits (**GRU**) for analysing the time series (confirmed cases, deaths and recovered cases) of the ongoing **CO**rona**VI**rus **D**isease 2019 (**COVID-19**). These different models are compared using **R**oot-**M**ean-**S**quare-**E**rror (**RMSE**) criterion, to evaluate their robustness compared to a naive forecasting approach.

1 Introduction

In the statistical analysis of time series, **A**uto-**R**egressive **M**oving **A**verage (**ARMA**) models provide a parsimonious description of a (weakly) stationary stochastic process in terms of two polynomials, one for the Auto-Regression (AR) and the second for the Moving Average (MR). An ARIMA model is a generalization of the ARMA model [1]. Both of these models are fitted to time series data to predict future points in the series (forecasting).

GRUs are a powerful set of gating mechanisms inspired from **R**ecurrent **N**eural **N**etworks (**RNN**). GRUs try to solve the vanishing gradient problem that can come with standard recurrent neural networks.

In this work, we aim to contrast the statistical accuracy of ARIMA with the modelling capacity of GRUS, in forecasting the spread of the ongoing COVID-19 outbreak, which has recently been declared a pandemic by the **W**orld **H**ealth **O**rganization (**WHO**) [2].

2 The Dataset

Since the pandemic is ongoing at the time of submission of this project, we only rely on the official figures compiled by trusted institutions. As such, we train our models on the **2019-nCoV Dataset** [3] provided by **J**ohns **H**opkins **U**niversity **C**enter for **S**ystems **S**cience and **E**ngineering (**JHU CSSE**) [4]. This contains credible information sourced from various accredited national and international organizations and governmental institutions since January 21st 2020. Further, we make use of the latest 10 days of data for testing, while we use the rest as the training data for our models.

3 The Problem Statement

A **time series** is a sequence of observations ordered in time. The observations are collected over a fixed time interval, and the time dimension adds structure and constraint to the data. Hence we define our time series as:

$$X_t; t = 0, 1, 2, ...$$

where X_t is the observation at timestep t.
Forecasting is the art of predicting the future observations from the past. Given an observed value X, we aim to predict X_{t+1} using $X_1, ..., X_t$. In other words, learn f such that:

$$X_{t+1} = f(X_1, ..., X_t)$$

Thus our work is to accurately predict the future observations in the test set based on the training set of existing observations. Assume the predicted value at any time step $k \in N$ is P_k, where $N = \{k | t + 1 \leq k \leq t + n\}$ is a window of evaluation

for a suitable value of n. The widely used RMSE criterion is applied to evaluate performance:

$$RMSE = \sqrt{\frac{\Sigma_{k \in N}(X_k - P_k)^2}{|N|}}$$

4 Challenges

4.1 Curse of Configuration

Perhaps the biggest issue facing ARIMA models is that some of the traditional model identification techniques for identifying the correct model configuration from the class of possible models are difficult to understand and usually computationally expensive. This process is also subjective and the reliability of the chosen model can depend on the skill and experience of the forecaster.

4.2 Complex Relationships

The underlying theoretical model and structural relationships are not distinct as some simple forecasts models such as simple exponential smoothing. As such, even after choosing a promising ARIMA configuration subject to a suitable evaluation metric, it is challenging to provide a reasonable intuition behind the effectiveness of the model chosen. Moreover, the ARIMA models, as all forecasting methods, are essentially "backward-looking". As such, the long term forecast eventually goes to be straight line and poor at predicting series with extreme turning points.

5 The Work Approach

5.1 Pre-Processing

For evaluating our techniques, we make use of the **Walk Forward Validation** scheme, also known as expanding window multiple sets. Essentially we order our data in the following manner:

Data	
Train Size	Test Size
10 timesteps $(X_1 - X_{10})$	X_{11}
11 timesteps $(X_1 - X_{11})$	X_{12}
12 timesteps $(X_1 - X_{12})$	X_{13}
13 timesteps $(X_1 - X_{13})$	X_{14}

We set a base window size of 3 days and grow our window from there. Moreover, due to the scarcity of data (the pandemic being only 90 days old), we make use of the last 15 days as the test set observations and the remaining as the training set data.

5.2 Baseline Naive Forecasting / Persistence

We make use of **Naive Forecasting** or **Persistence** as a baseline approach for our problem. Persistence is a dumb forecasting approach, where we forecast the next value to be the same as the latest value available to us. That is, we predict:

$$P_{t+1} = X_t$$

This approach gives us a base score to beat and evaluate how well our model performs compared to small scale changes.

5.3 ARIMA Modelling

Auto Regressive (AR) Models

If the series is not entirely white noise, then the forecasting can be modeled as:

$$P_t = f(X_1, ..., X_{t-1}, e_t)$$

where e_t is the white noise. Practically, it is not feasible to consider all time steps in the above computation, so we approximate as:

$$P_t = \beta_0 + \beta_1 X_{t-1} + \beta_2 X_{t-2} + ... + \beta_p X_{t-p} + s_t$$

We denote this model as $AR(p)$.

Moving Average (MA) Models

In moving average models, we model upon the white noise observations:

$$P_t = f(e_1, ..., e_{t-1}, e_t)$$

Using the previous analogy, an $MA(q)$ model learns:

$$P_t = \gamma_0 e_t + \gamma_1 e_{t-1} + \gamma_2 e_{t-2} + ... + \gamma_q e_{t-q}$$

We denote this model as $MA(q)$.

Differencing: Converting Non-Stationary to Stationary

A **stationary** time series is one whose statistical properties such as mean, variance, etc. are all constant over time. In algebraic terms, the series tends to follow a linear curve. A time series which is non-stationary can be converted to a stationary one by differencing.

$$X_t^{'} = X_t - X_{t-1}$$

If the obtained time series is still non-stationary, we can do second-order differencing:

$$X_t^{''} = X_t^{'} - X_{t-1}^{'} = X_t - 2X_{t-1} + X_{t-2}$$

If differencing is done once, it is called $I(1)$. Extending the differencing to dth order, we get $I(d)$.

Auto Regressive Integrated Moving Average (ARIMA) Models

In ARIMA, the $AR(p)$ and $MA(q)$ are same as above.

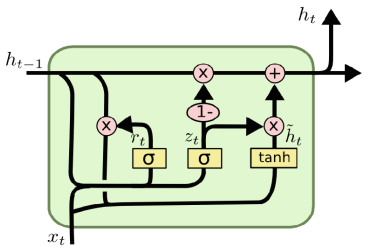
$$P_t = \beta_0 + \beta_1 X_{t-1} + \beta_2 X_{t-2} + ... + \beta_p X_{t-p} + s_t + \gamma_0 e_t + \gamma_1 e_{t-1} + \gamma_2 e_{t-2} + ... + \gamma_q e_{t-q}$$

However, I indicates the amount of difference done. Thus an $ARIMA(p, d, q)$ model is a combination of $AR(p)$ and $MA(q)$ with $I(d)$. We make use of a grid-based search strategy for finding the optimal values of p, d and q.

5.4 GRU Modelling

Introduced by Cho et al. [5], a **G**ated **R**ecurrent **U**nit (**GRU**) aims to solve the vanishing gradient problem which plagues a standard **R**ecurrent **N**eural **N**etwork (**RNN**).

A GRU instead of having a simple neural network perceptron, has a cell containing multiple operations. GRU uses the so called, **update gate** and **reset gate**. The σ notation above represent those gates, which allows a GRU to carry forward information over many time periods in order to influence a future time period. The value is stored in memory for a certain amount of time and at a critical point pulling that value out and using it with the current state to update at a future date.



The above cell symbolizes the following:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$
$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$
$$\tilde{h}_t = tanh(W \cdot [r_t * h_{t-1}, x_t])$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

where, z_t and r_t are the update and reset gates at timestep t, $\tilde{h}_t$ is the current memory content and h_t is the final memory content at current timestep. Further $\cdot$ and $*$ represent matrix multiplication and element-wise product respectively.

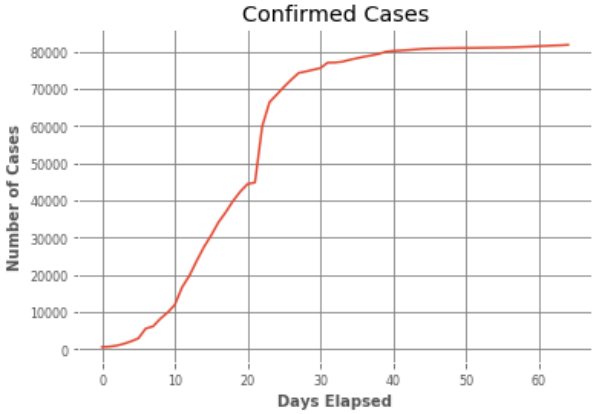
6 Benchmarks (as of April 05, 2020)

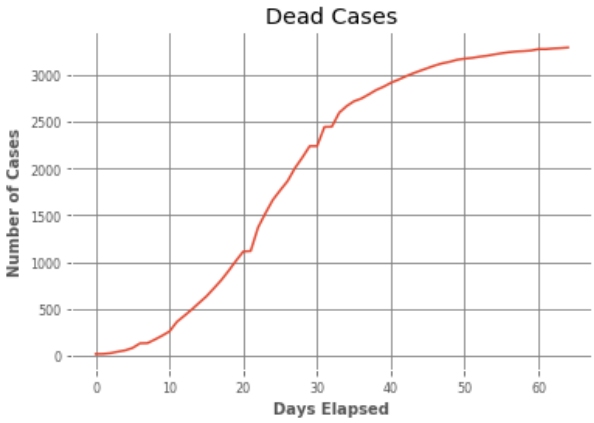
We run our model on the following three countries of interest:

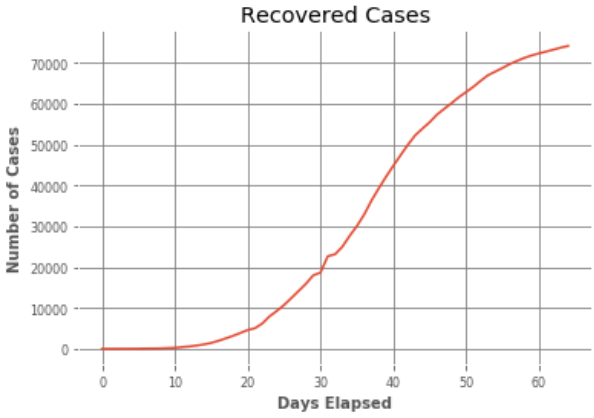
China

At the time of writing of this report, the spread of the pandemic in China has seemingly come to a standstill with over 80,000 infected patients, over 3,000 deaths and 70,000 further recoveries over a period of first 65 days.

Summary Statistics

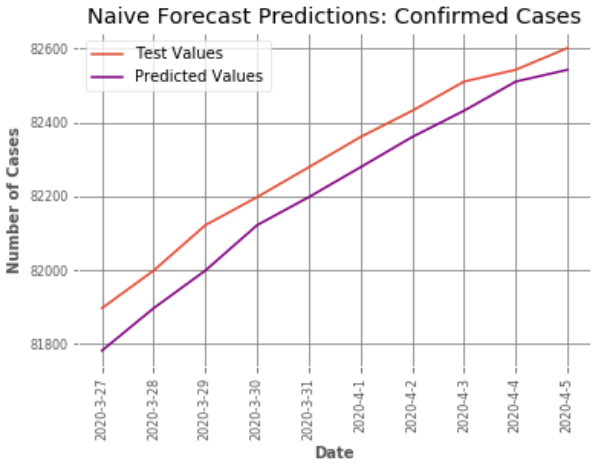




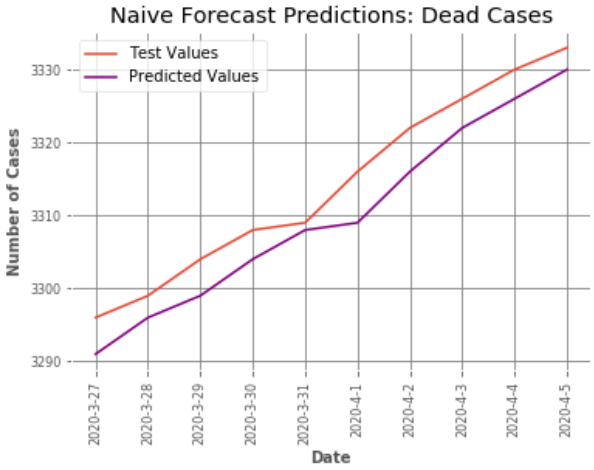


As the number of infections have stagnated, over the last 10 days, the naive forecasting method provides a robust baseline score for this sequence, as observed from the graphs below.

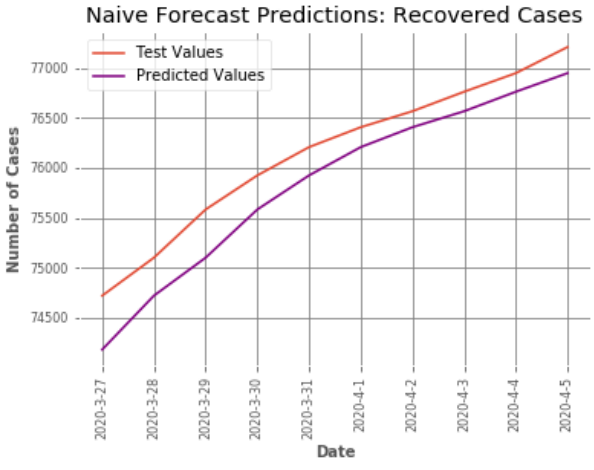
Naive Forecasting / Persistence



RMSE: 85.7939



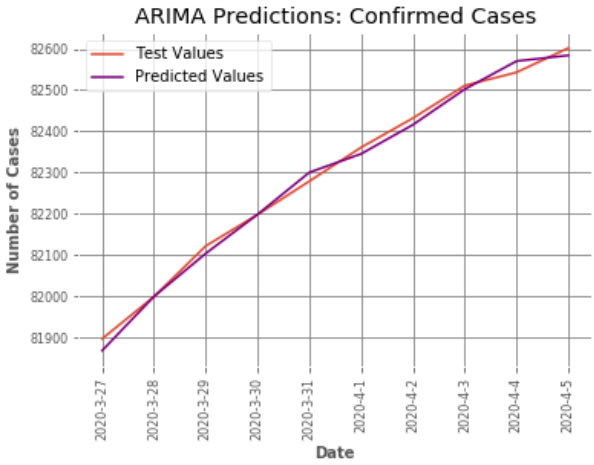
RMSE: 4.4944



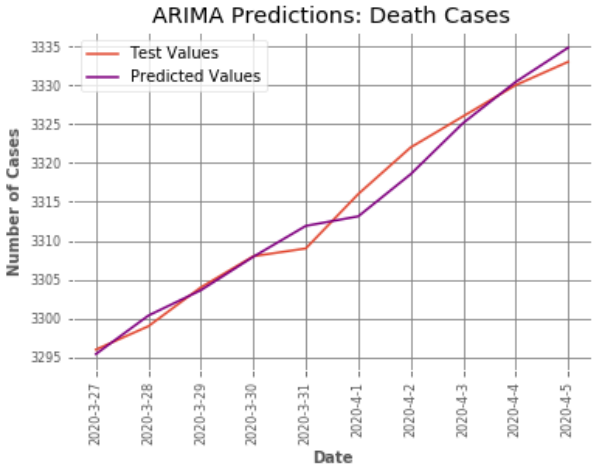
RMSE: 327.0409

For ARIMA modelling, we do a grid search over 512 value combinations and find that the values of $(5, 1, 4)$, $(2, 1, 2)$ and $(6, 2, 4)$ serve as the best approximation for the values of (p, d, q) for confirmed, dead and recovered time series.

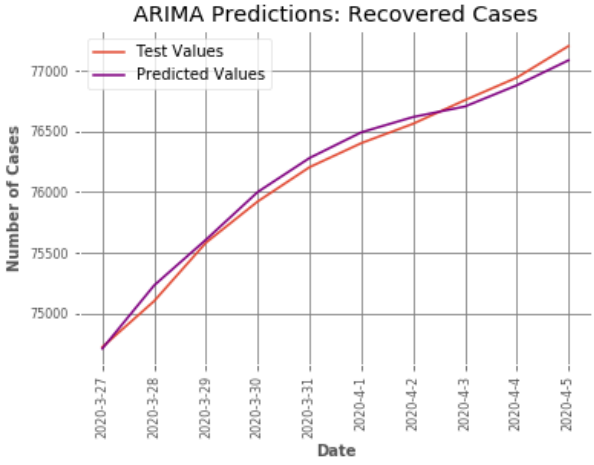
ARIMA Modelling



RMSE = 18.059

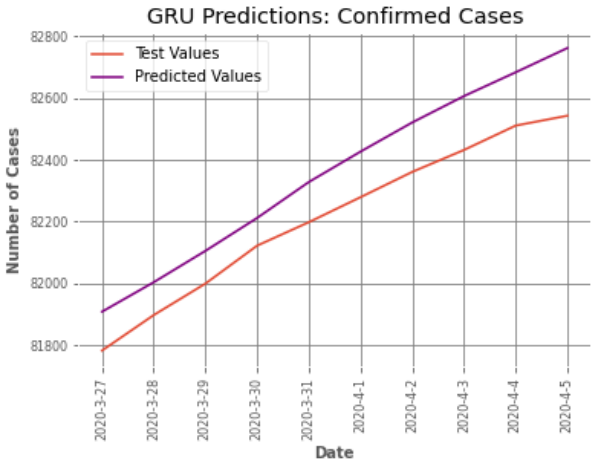


RMSE = 1.874

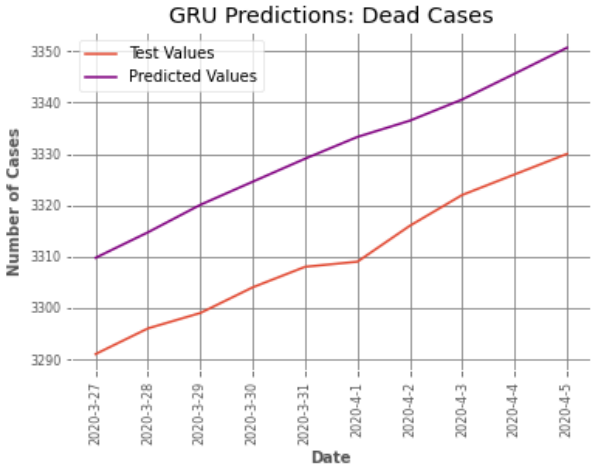


RMSE = 78.286

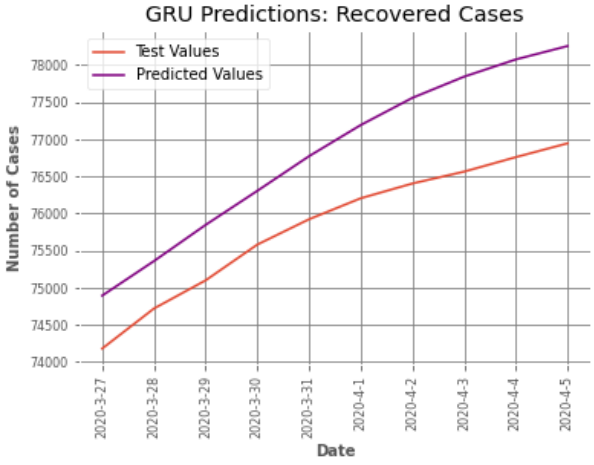
GRU Modelling



RMSE = 143.059



RMSE = 15.3298

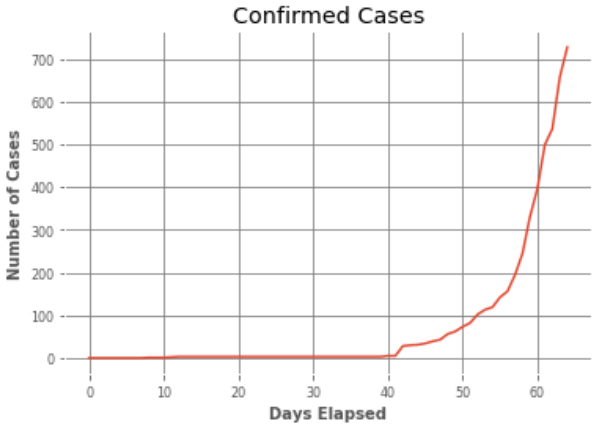


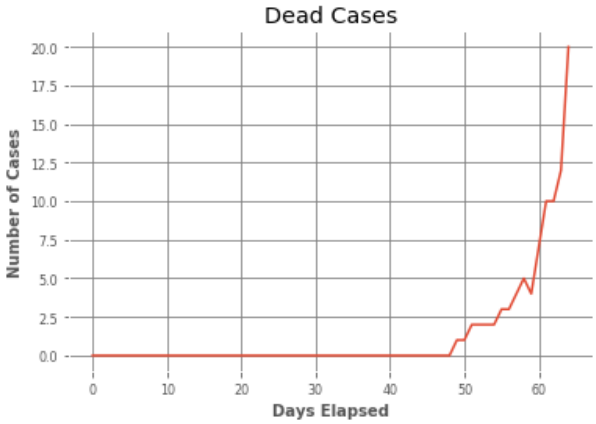
RMSE = 972.1164

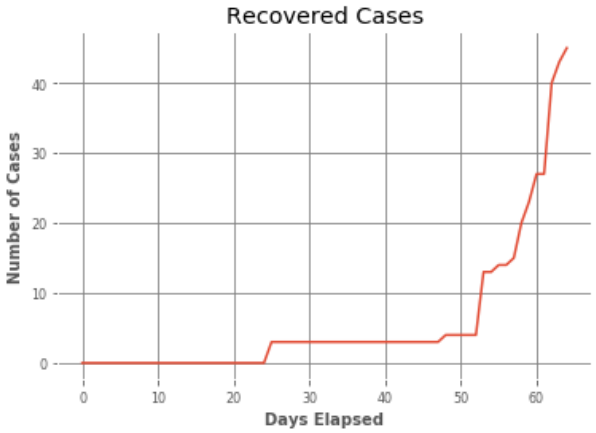
India

At the time of writing of this report, the spread of the pandemic in India has merely begun with over 700 infected patients, over 20 deaths and 45 recoveries over a period of 65 days. Since the entry date of the virus was quite late for India, hence for the most part of the above duration, our time series is pretty stagnant at zero.

Summary Statistics



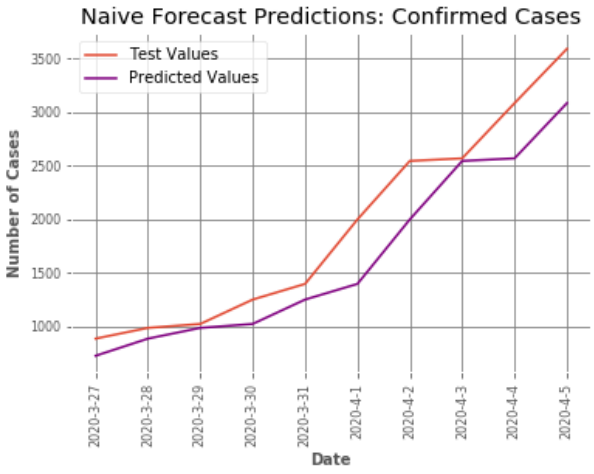




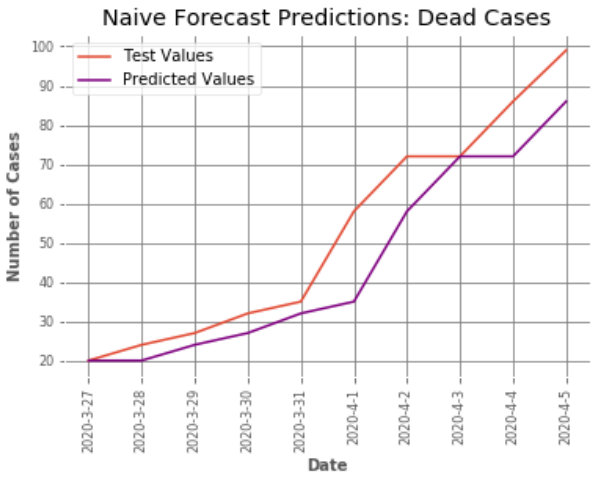
As the number of infections have only risen, over the last 10 days, the naive forecasting method

does not provide a robust baseline score for this sequence, as observed from the graphs below.

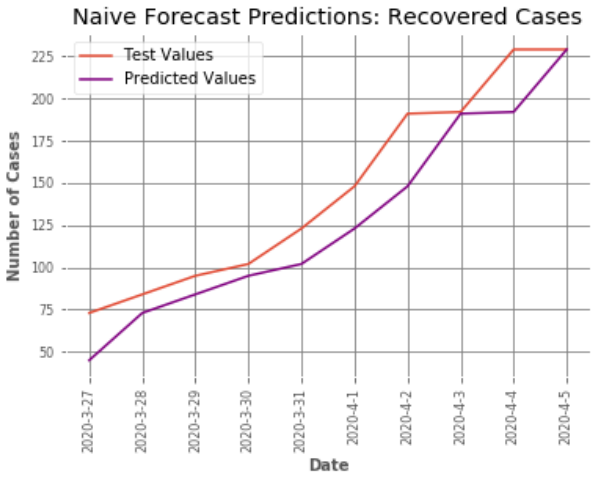
Naive Forecasting / Persistence



RMSE: 359.1485



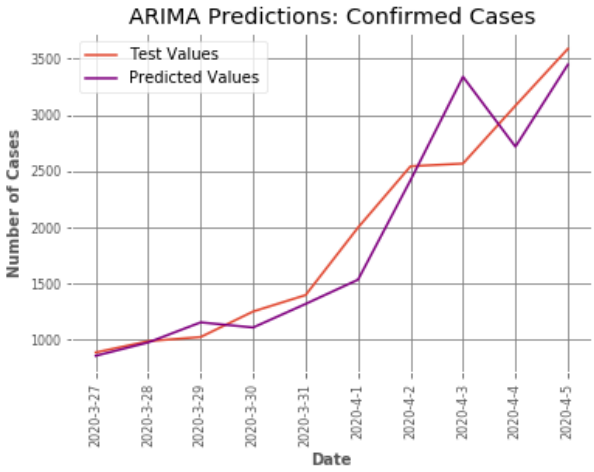
RMSE: 10.7191



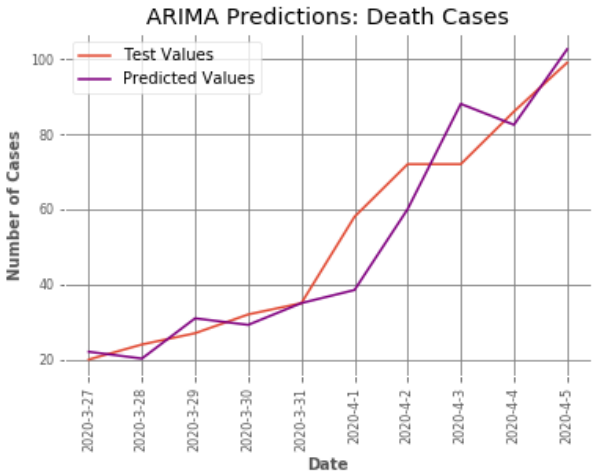
RMSE: 23.1516

For ARIMA modelling, we do a grid search over 512 value combinations and find that the values of $(1, 2, 2)$, $(2, 2, 1)$ and $(1, 2, 1)$ serve as the best approximation for the values of (p, d, q).

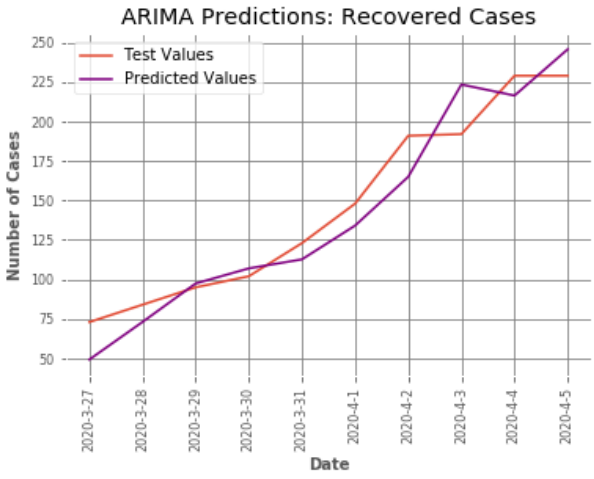
ARIMA Modelling



RMSE = 320.014

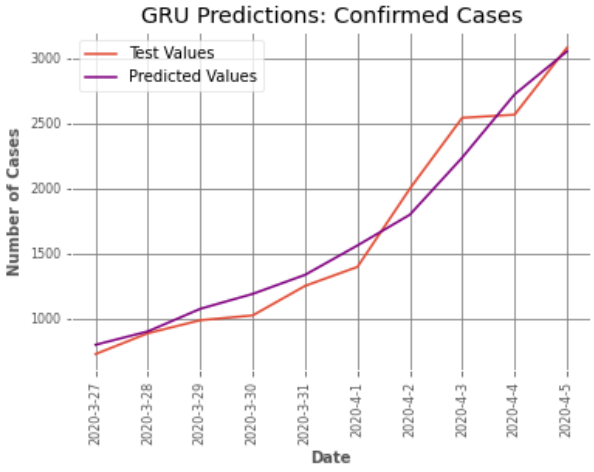


RMSE = 9.207

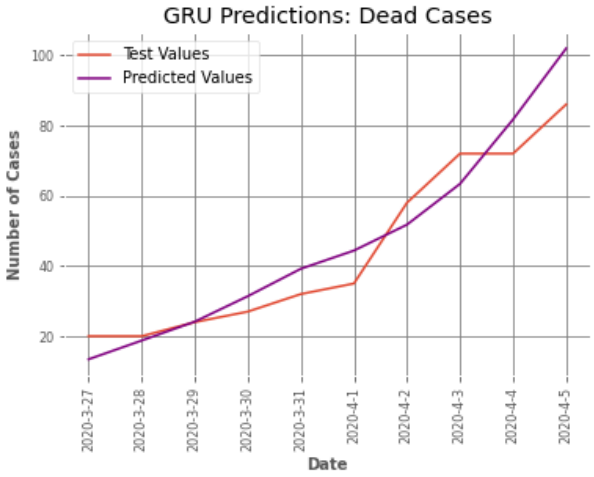


RMSE = 17.641

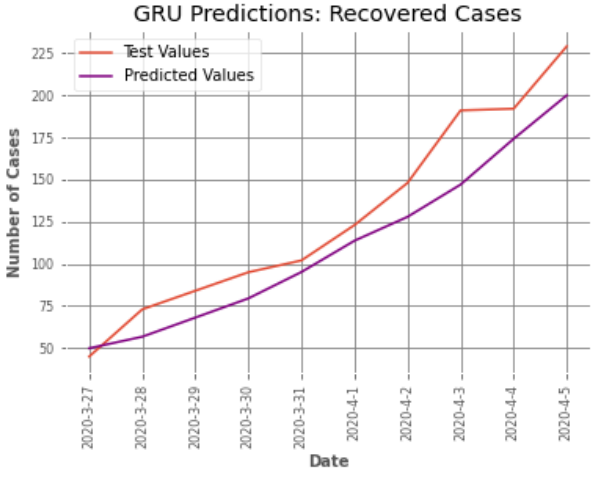
GRU Modelling



RMSE = 127.8480



RMSE = 6.9430

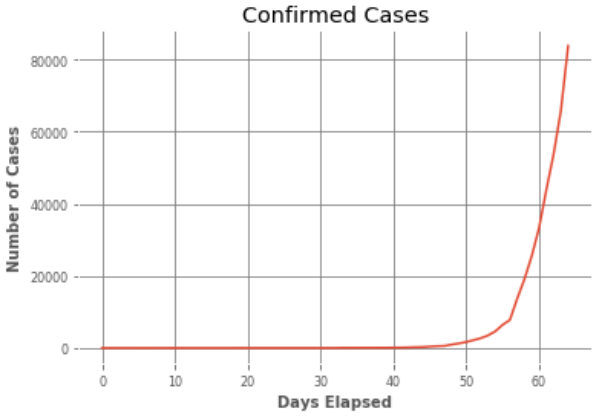


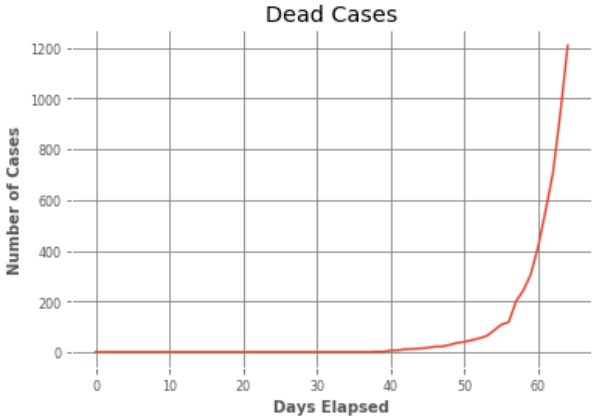
RMSE = 17.9782

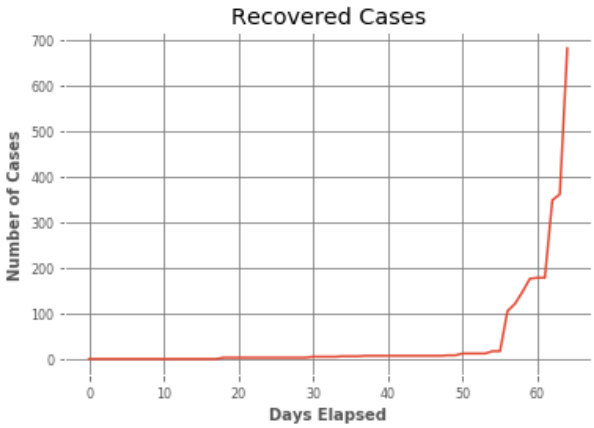
United States of America

At the time of writing of this report, the spread of the pandemic in the US has seemingly blown up considerably with over 80,000 infected patients, over 1,200 deaths and 700 recoveries over a period of 65 days.

Summary Statistics

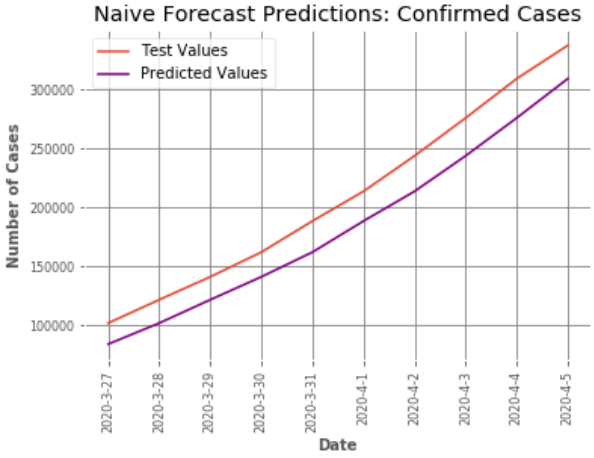




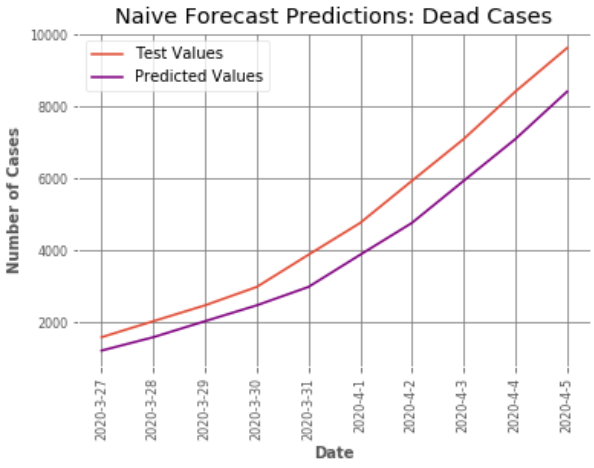


As the number of infections have considerably gone up, over the last 10 days, the naive forecasting method does not provide a robust baseline score for this sequence, as observed on the graphs below.

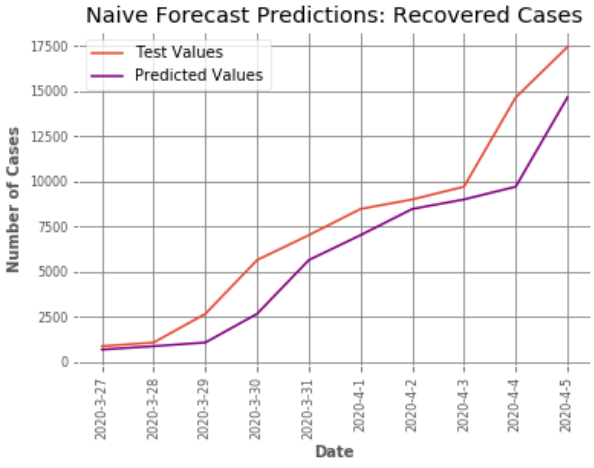
Naive Forecasting / Persistence



RMSE: 25875.3613



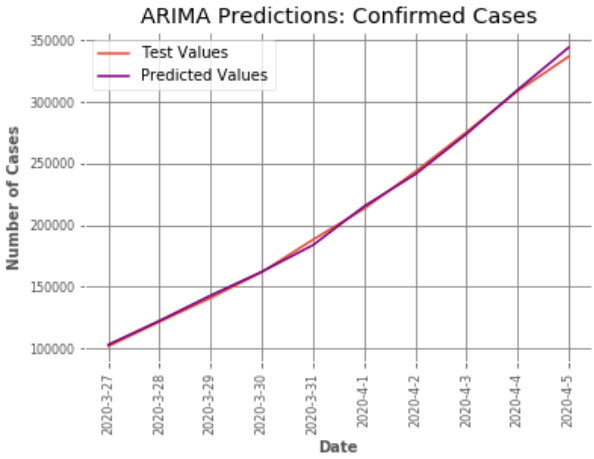
RMSE: 911.0608



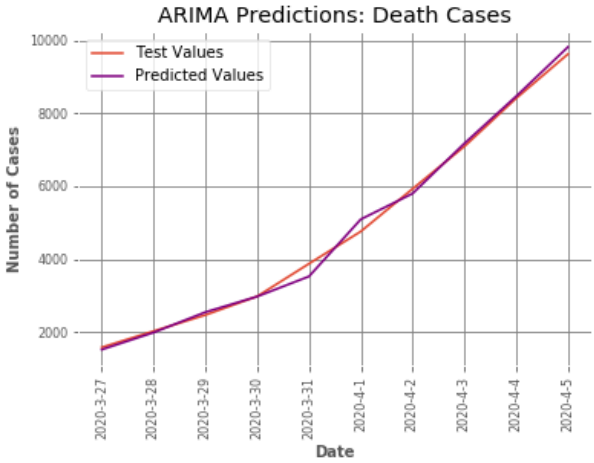
RMSE: 2203.2328

For ARIMA modelling, we do a grid search over 512 value combinations and find that the value $(2, 2, 1)$ serve as the best approximation for the values of (p, d, q) for all three time series.

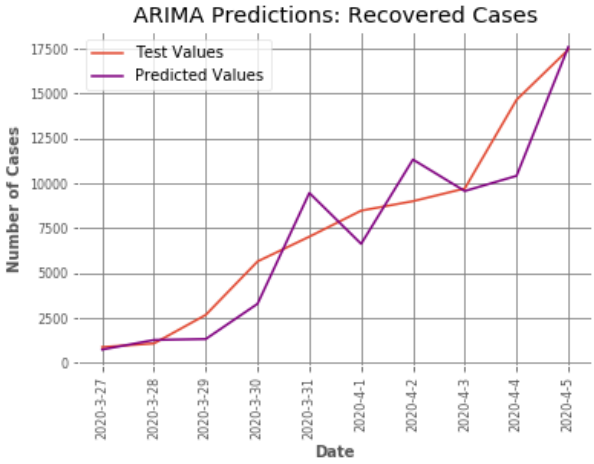
ARIMA Modelling



RMSE = 3034.169

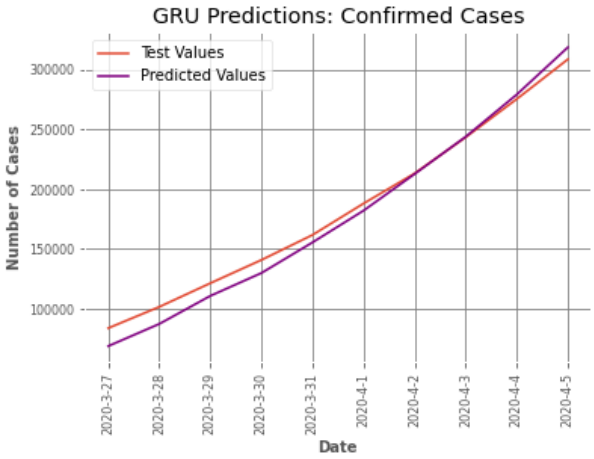


RMSE = 175.435

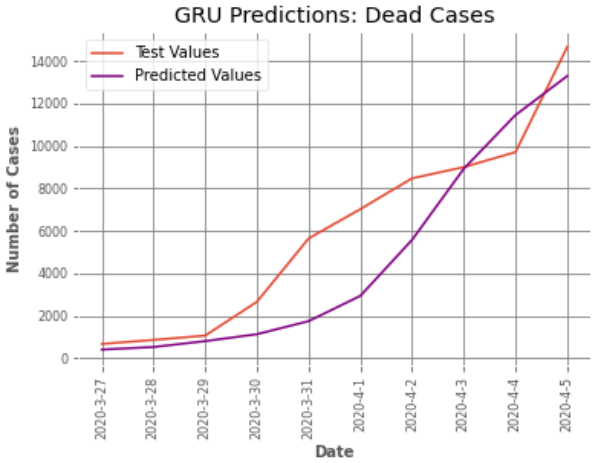


RMSE = 2004.952

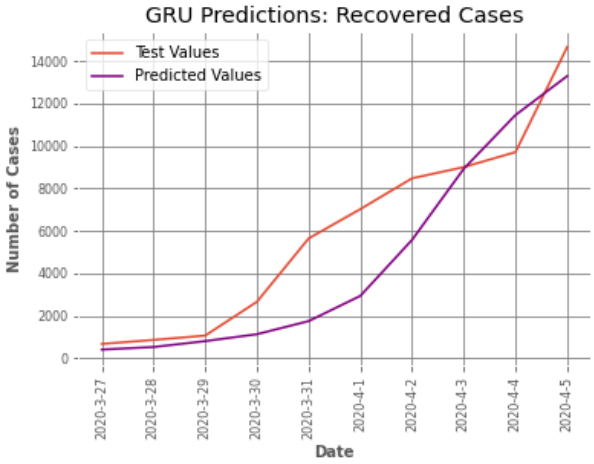
GRU Modelling



RMSE = 7009.9414



RMSE = 166.4410



RMSE = 1644.3438

7 Conclusion and Future Work

From our varied experiments, we can draw the following conclusions:

• For countries where the pandemic has already reached its peak, such as China, and the time-series are mostly stagnant, **ARIMA** outperforms every other technique.
• For countries where the pandemic has begun to spread, but is far away from imploding, **GRU** outperforms every other technique. In such countries, like India, the time series is zero for the most duration and rises sharply thereon.
• For countries where the pandemic is rapidly spreading, like the United States of America, and the time-series is the most volatile, **ARIMA** and **GRU** perform equally well.

7.1 Additional Features

In this project, we simply made use of the univariate data made available to us. Further, we can incorporate more crucial information such as weather and climatic conditions, population demographics, mobility, and the healthcare infrastructure in individual countries. However, we leave these extensions for future work.

7.2 Disease Specific Modelling

In recent times, disease spread modelling has gained a lot of traction. Models like **S**usceptible **I**nfectious and **R**emoved (**SIR**) are specifically tuned towards gaining insight into how pandemics spread on a global scale. The model consists of three components: S for the number of susceptible, I for the number of infectious, and R for the number of recovered or deceased (or immune) individuals. This model is reasonably predictive for infectious diseases which are transmitted from human to human, such as measles, mumps and rubella.

In conclusion, we would like to thank our professor, Dr. Joydeep Chandra for providing motivation to pursue this project as a part of the Deep Learning course. Further, we would like to thank Google Colaboratory team for their robust and easy-to-use online notebook resources for running various scripts and deploying the models on GPUs.

References

[1] G. E. Box, G. M. Jenkins, and G. C. Reinsel. "Time Series Analysis" *New Jersey: Prentice Hall (1994)*

[2] World Health Organization Director-General. "WHO Director-General's Opening Remarks At The Media Briefing On COVID-19" *11 March 2020*

[3] Johns Hopkins CSSE-GIS Data. "2019 Novel Coronavirus COVID-19 (2019-nCoV) Data Repository by Johns Hopkins CSSE" *GitHub*

[4] Johns Hopkins CSSE. "CSSE - Center For Systems Science and Engineering at JHU" *Johns Hopkins University, Baltimore, Maryland*

[5] K. Cho, B. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation" *University de Montreal, Canada (2014)*