Predicting Confirmed Cases of COVID-19

Cain Farnam

Abstract

Since its discovery, the novel coronavirus (COVID-19) has become the headline of every news outlet nearly everyday. With the number of confirmed cases now excedding 3 million globally, many wonder if the pandemic will ever end. This paper aims to predict the total number of confirmed cases of COVID-19 using time series models as well as neural networks. Five different models are trained and evaluated. The models are compared and the optimal model is chosen for application.

---

1 Introduction

1.1 COVID-19

In December of 2019, Chinese health officials identified a new virus (COVID-19) that originated from the Wuhan Province. Classified as a corona virus, COVID-19 is a respiratory virus that can cause fever, cough, and shortness of breath. The virus can be more severe, especially in cases of patients having a pre-existing condition. On March 11, 2020, the World Health Organization declared the outbreak of COVID-19 a pandemic. The number of total confirmed cases has grown to over 3 million globally, with over 1 million in the US alone.

1.2 COVID-19 Data

The data set contains numerous features about each individual patient that has been tested for COVID-19. The data used will be a subset of the data set, containing only the cumulative number of confirmed cases, deaths, and recoveries of COVID-19 since January 22, 2020 for each nation. The data set is provided by John Hopkins University.

2 Analysis

2.1 Exploratory Data Analysis

2.1.1 Global Totals

The graph in Figure 1 displays the total number of confirmed cases, recoveries, and deaths from COVID-19 globally.

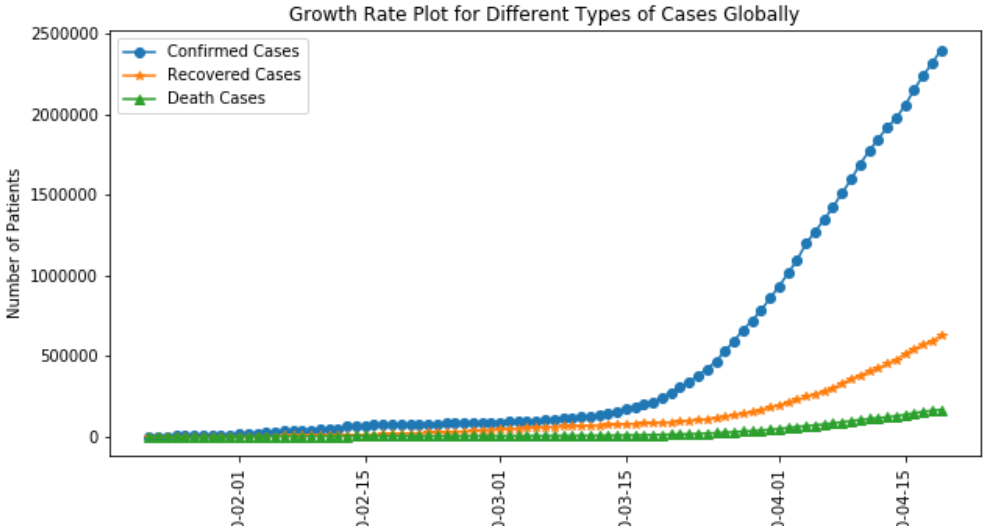

Figure 1: Global totals

2.1.2 US Totals

Figure 2 provides the graph for totals in just the US. Note that trend in the US seems to reflect that of the global graph.

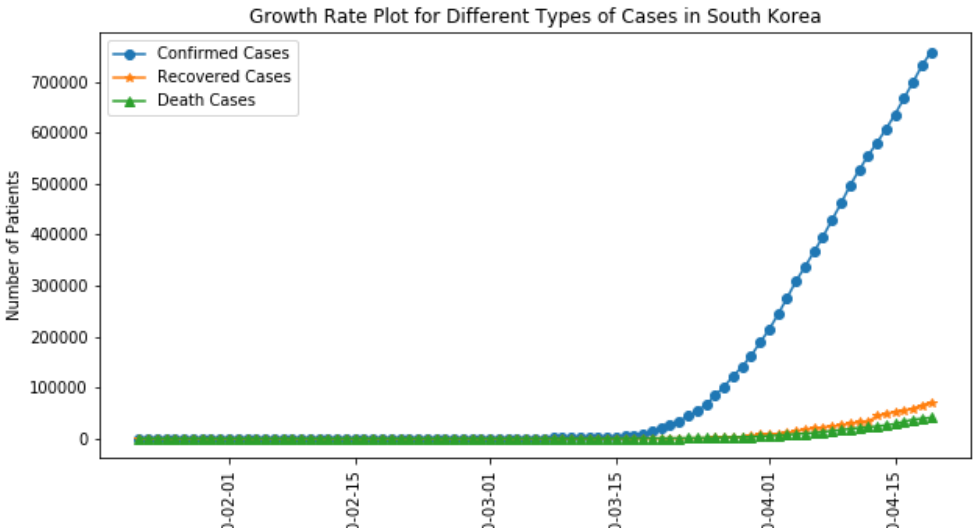

Figure 2: US totals

2.1.3 South Korea Totals

South Korea's COVID-19 totals are displayed in Figure 3. The number of confirmed cases has drastically reduced in speed. South Korea has been widely praised for their successful

efforts of slowing down the growth of the virus, or "flattening the curve" as it is expressed in the media.

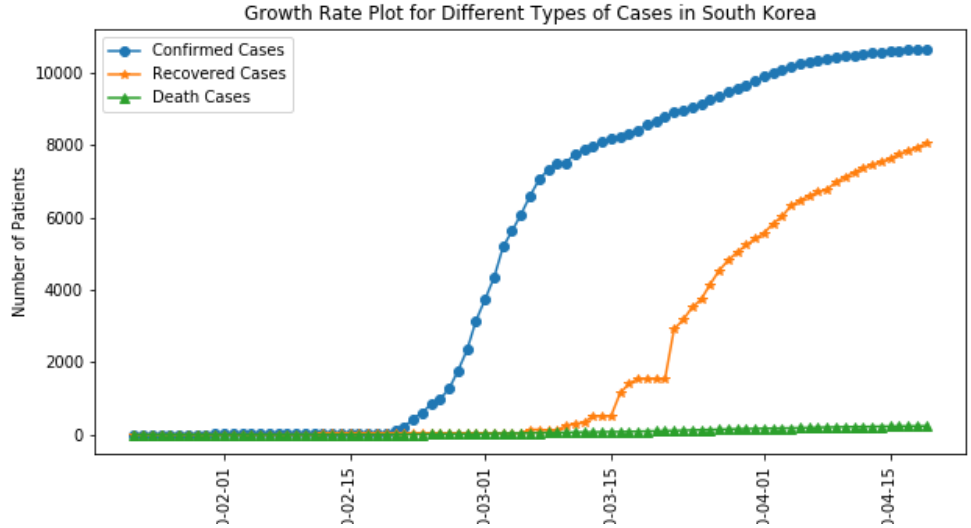

Figure 3: South Korea totals

2.1.4 Country Comparison

Figure 4 is a comparison of some of the most affected countries. It will be important to watch the totals in India as time goes on. There have been suspicions that several factors in India may lead to a larger outbreak than other countries.

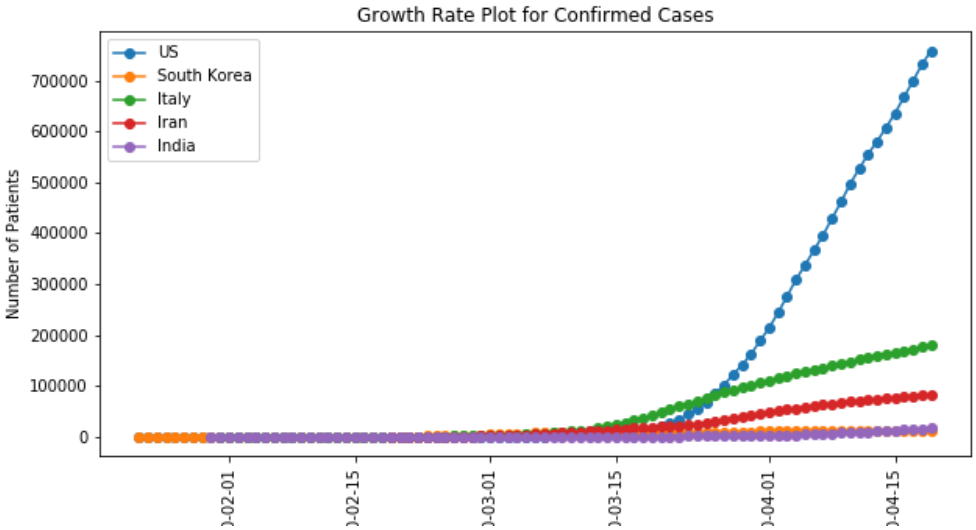

Figure 4: Comparison of confirmed totals

2.2 Objectives

The primary objective is to predict the spread of COVID-19, both globally and nationally. Due to the small amount of data, it may be difficult to accurately forecast the totals. Though it may be too early to tell. The models will be evaluated by both the mean absolute error (MAE) and root mean squared error (RMSE) performance measures.

2.3 Models

To forecast the number of confirmed cases of COVID-19, we will use the following models:

- Holt's Linear
- Holt-Winters
- Autoregression (AR)
- Autoregressive Integrated Moving Average (ARIMA)
- Recurrent Neural Network (RNN)

2.3.1 Neural Networks

A neural network is a type of deep learning model that is created with layers of nodes that perform operations on the data. Each layer receives input and parameterizes said input using "weights". After each of the layers undergo parameterization, the final layer predicts the output. A "loss function" (also called objective function) is then used to control the success of the algorithm by comparing the predicted outcomes to the expected outcomes. After the loss function calculates a loss score, an "optimizer" will slightly adjust the weights of all the layers by using a backpropagation algorithm. These steps are then repeated for multiple "training loops", or epochs, until the predicted values are very close to the expected values. Figure 5 below provides a very basic structure of a neural network with an input layer, two "hidden" layers, and an output layer that contains the predictions.

Neural networks have a large number of hyperparameters (parameters that are set before training). These hyperparameters may include:

- Types of hidden layers
- Number of hidden layers
- Amount of nodes in each layer
- Number of epochs
- Batch size
- Loss function
- Optimizer

This is not an exhaustive list of all hyperparameters, but all of those listed must be included in the model, whereas others don't have to be.

A recurrent neural network (RNN) is a special type of neural network that works really well with sequential data, like time series data. RNN's make use of long short term memory (LSTM) hidden layers that allow the model to retain pieces of information from previous

iterations. These LSTM layers are what make RNN's powerful when it comes to forecasting time series.

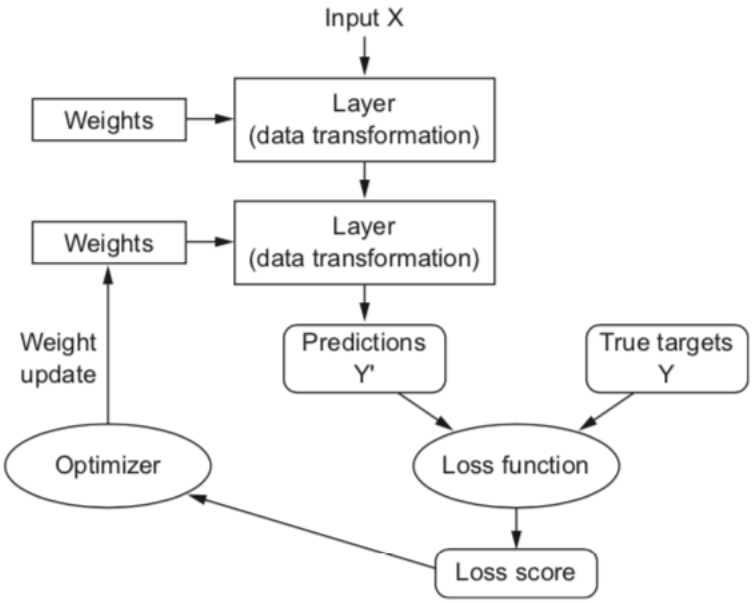

Figure 5: Neural network structure

3 Results

3.1 Holt's Linear Model

The Holt's linear model was tuned to find the optimal hyperparameters. The hyperparameters considered were the smoothing level and the smoothing slope. After testing various combinations of hyperparameters, the optimal Holt's linear model was found the have the following hyperparameters:

- Smoothing level: .3
- Smoothing slope: 1.1

The model performance measures from this model were as follows:

- MAE: 25750
- RMSE: 29245

Figure 6 shows the predicted values from this model compared to the actual values for April 15 through April 19th.

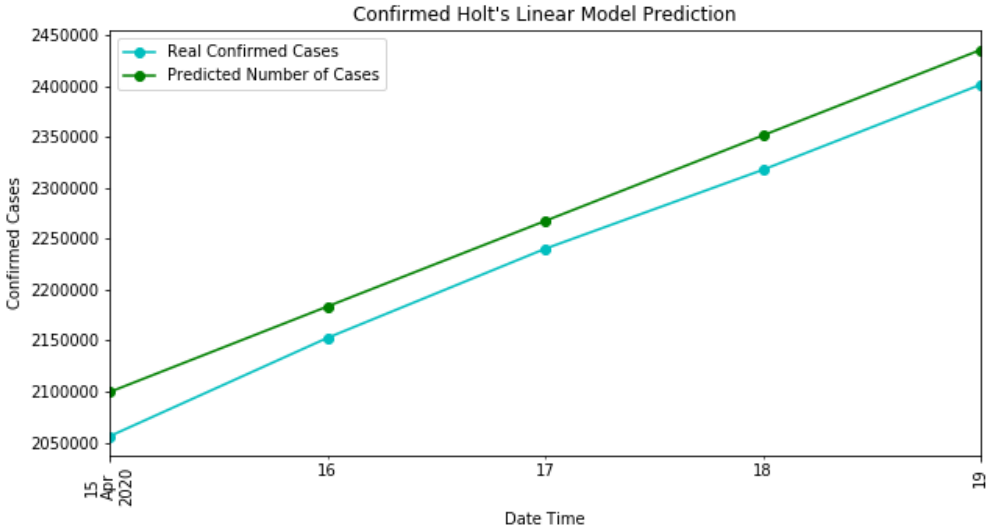

Figure 6: Holts linear model predictions vs. true values

3.2 Holt-Winters Model

The Holt-Winters model was trained using an additive trend and a multiplicative seasonality.

The model performance measures from this model were as follows:

- MAE: 23117
- RMSE: 26037

Figure 7 shows the predicted values from this model compared to the actual values for April 15 through April 19th.

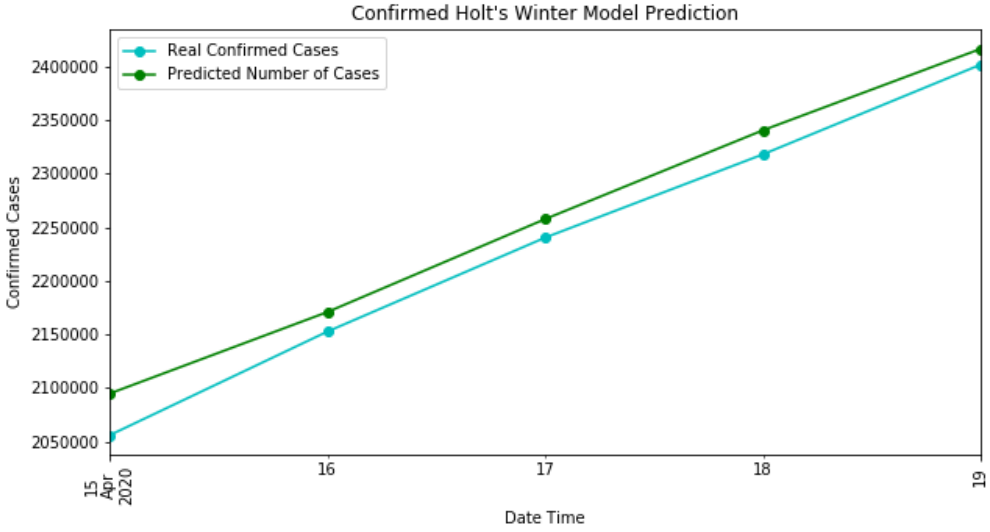

Figure 7: Holt-Winters model predictions vs. true values

3.3 Transformation

Before the data can be used to train the rest of the models, the data must first be stationary. As the Dicky-Fuller tests provides a p-value near 1, it's clear that the data is not stationary (it's clearly exponential).

After a log transformation, the Dickey-Fuller test returns a p-value of 0.00. Now that the data has been transformed, it can be safely used to train the AR, ARIMA, and RNN models. Figure 8 shows the decomposition of the data after the log transformation.

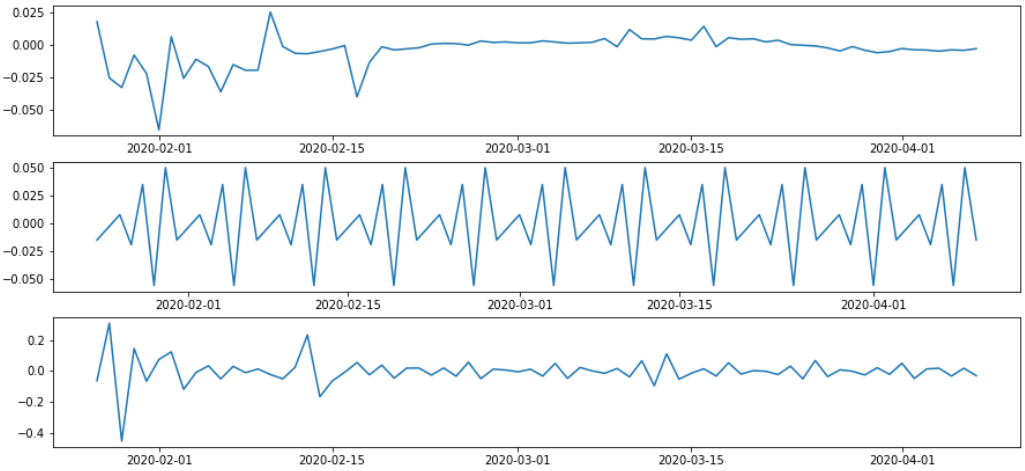

Figure 8: Decomposition of data after log transformation

3.4 AR

The AR(p) model was tuned by using various values for p. After inspecting the autocorrelation plot, the optimal value for p was found to be 4.

The model performance measures from this model were as follows:

- MAE: 29746
- RMSE: 35884

Figure 9 shows the predicted values from this model compared to the actual values for April 15 through April 19th.

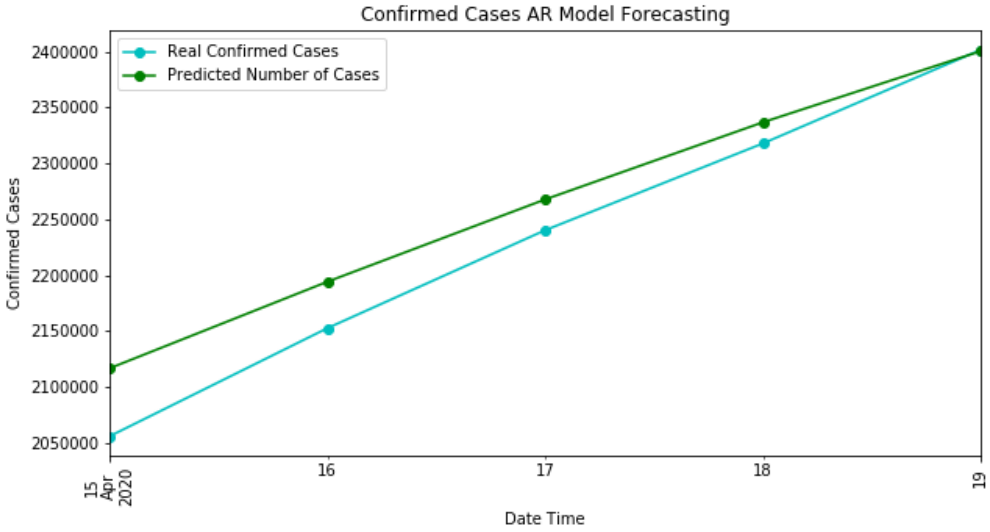

Figure 9: AR model predictions vs. true values

3.5 ARIMA

The ARIMA(p,d,q) model was tuned to find optimal values for p, d, and q. After testing various combinations of parameters, the optimal ARIMA model was found to have values p = 2, d = 2, q = 1.

The model performance measures from this model were as follows:

- MAE: 24290
- RMSE: 29326

Figure 10 shows the predicted values from this model compared to the actual values for April 15 through April 19th.

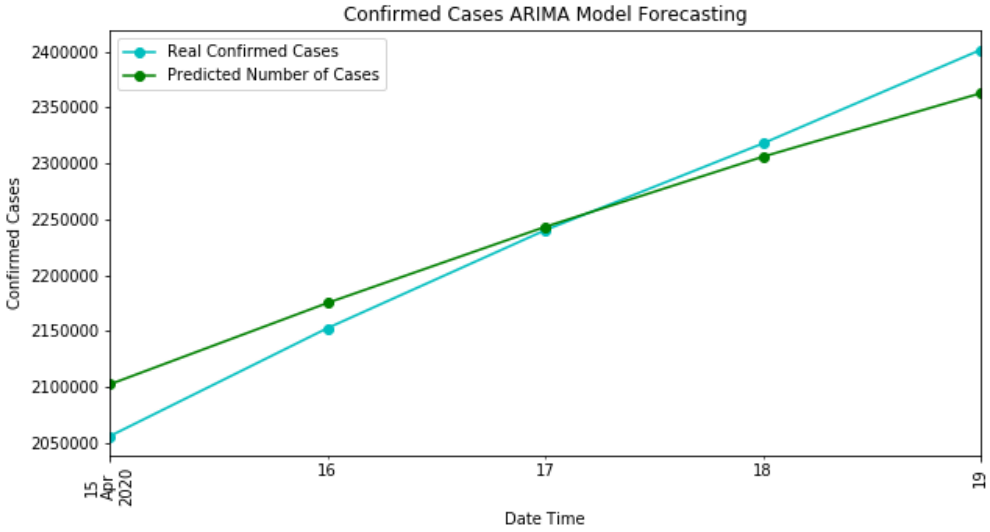

Figure 10: ARIMA model predictions vs. true values

3.6 RNN

The RNN model was tuned using 5-fold cross validation to find optimal hyperparameters. The hyperparameters considered were type of layers, number of layers, size of each layer, number of epochs, and batch size. The loss function (RMSE) and optimizer (ADAM) stayed constant over all models. The optimal RNN model was found to have the following hyperparameters:

- Layers:
  - LSTM: 2
  - Dropout: 2
- Nodes: 200
- Epochs: 30
- Batch size: 4

The model performance measures from this model were as follows:

- MAE: 11601
- RMSE: 14796

3.7 Comparison of Models

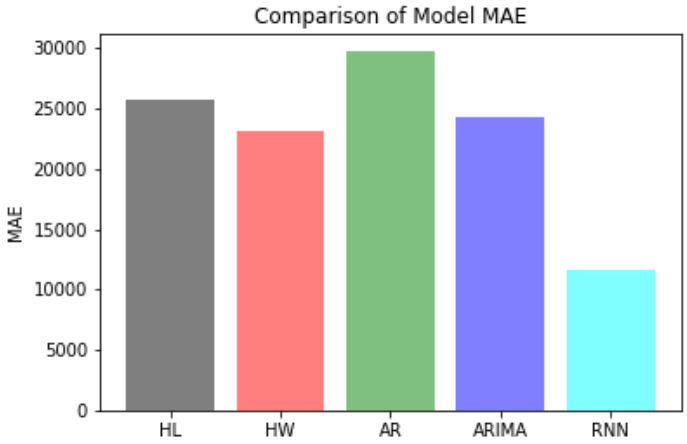

Figure 11: Comparison of Model Performance (MAE)

Figure 11 contains a barplot comparing all of the models' MAE performance while Figure 12 compares RMSE. It is clear to see that for both measures the RNN model is outperforming the other models by a great deal.

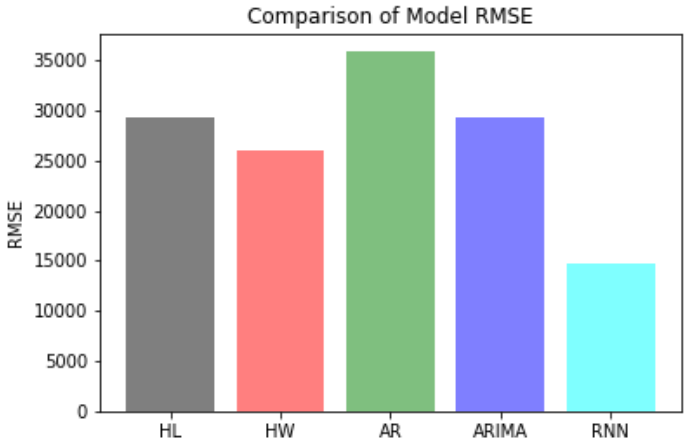

Figure 12: Comparison of Model Performance (RMSE)

4 Discussion

The RNN outperforms the traditional forecasting methods by far. It was able to correctly predict the number of COVID-19 confirmed cases globally with a MAE score less than half the next best model (Holt's Winters).

5 Future Work

Due to the nature of this data set, there will be an additional observation each day. It is likely that given the severity of COVID-19, research will continue on the data far after the pandemic is over. It would always be best to retrain the models with as much new data as possible. It may be work looking into other types of time series models or machine learning models.

References

Chollet, Francois. 2018. *Deep Learing with Python*. Manning.

Robert H. Shumway, David S. Stoffer. 2011. *Time Series Analysis and Its Applications*. Springer.

n.d. https://coronavirus.jhu.edu.