Predicting New COVID-19 Cases in South Africa using Long Short Term Memory Recurrent Neural Networks

Jonathan Rossouw

1 Introduction

For over a year, the world as we knew it has been utterly upended by COVID-19. Millions of people around the world have gotten ill and millions have died. The whole world was brought to a standstill by the virus. South Africa in particular has been hard hit. Unemployment is at record highs while our hospitals are overflowing. Fortunately, scientists from around the world have worked together to develop vaccines that offer hope for an end to the pandemic. However, due to a lack of supply and poor planning, South Africa is not one of those countries that are able to quickly vaccinate their populations. Thus one of the only ways to fight the virus is to accurately predict the number of infections in the future and plan accordingly.

The number of COVID-19 infections, hereafter referred to as cases, is a time-series of data that can be modelled using machine learning techniques. In this paper, data from Google BigQuery's public datasets was used to model cases and then the models were used to predict 14 days of cases. Two classes of models were used. The first was a simple ARIMA model, used as a benchmark, and the second was a long short-term memory (LSTM) recurrent neural network (RNN). LSTM models have been used to model COVID-19 cases in Yudistira (2020) and Suárez-Cetrulo, Kumar & Miralles-Pechuán (2021) with high degrees of accuracy. LSTM models overcome issues faced by RNN models which allow the models to perform well on long sequences of data. LSTM models incorporate forget gates in each memory cell that allow hidden units to incorporate both new inputs and weighted combinations of past inputs. This effectively changes the weights of hidden units depending on the state of the inputs. Thus models are able to determine, from previous occurrences of similar states, the most appropriate weights for the current inputs based on the state of the inputs. This is particularly important for COVID-19 cases data as cases rise and fall in waves. The LSTM models are able to determine whether the inputs for part of a wave or a lull and change the weights in the hidden layers appropriately (Yudistira, 2020). Univariate and multivariate LSTM models were created and used to predict 14 days worth of cases. Model performances is based on the lowest test mean squared error (MSE). MSE was chosen as it is a standard measure of performance and it penalises predictions that are not able to identify sharp up-ticks in cases.

The remainder of the paper consists of a discussion of the data, the methodology of each of the models used, the results of the models and a conclusion.

2 Data

Data was sourced from Google Big Query public databases (Wahltinez & others, 2020). More specifically the covid_open_data was sourced using the following SQL query:

```sql
SELECT country_name, new_confirmed, date, location_key, cumulative_tested,
new_persons_vaccinated, new_deceased
FROM `bigquery-public-data.covid19_open_data.covid19_open_data`
WHERE country_name IN ('South Africa') AND location_key IN ('ZA')
ORDER BY date
```

Table 1: Description of data

Description	Date	Cases
First Cases	2020-03-03	5
Max Cases	2021-01-01	31328
Total Cases	-	1995556
Mean Cases	-	3648
Standard Deviation of Cases	-	4739

The new cases, cumulative tests, new vaccinated persons and newly deceased data was sourced. The new cases was used for both the ARIMA and univariate LSTM models and is discussed below where it is refered to as the data. The remaining three time series are used in the multivariate LSTM and are discussed in detail in the section 3.2.2.

The data consists of 546 observations spanning from 1 January 2020 to 30 June 2021. As seen in table 1, the first cases in South Africa appear on 3 March 2020, the maximum daily cases was 31328 cases on 1 January 2021. The total number of cases by 30 June is 1 995 556 with a mean of 3648 cases and a standard deviation of 4739 cases. From figure 1 it can be seen that South Africa has experienced three distinct waves of COVID-19. The fact that the data does not grow with a linear or exponential trend is part of the motivation for using a LSTM model. As described in the section 3.2, the model has a memory which allows it to identify which state the series is in and change the parameters appropriately. This allows the model to identify whether or not it is currently in a wave or a lull and predict cases accordingly.

The data is split into a training dataset from 13 June 2020 to 22 May 2021, a validation set which includes 19 May 2021 to 1 June 2021 for training and cases from 2 June 2021 to 15 June 2021 are to be predicted, and a test set where cases from 16 June 2021 to 30 June 2021 are to be predicted.

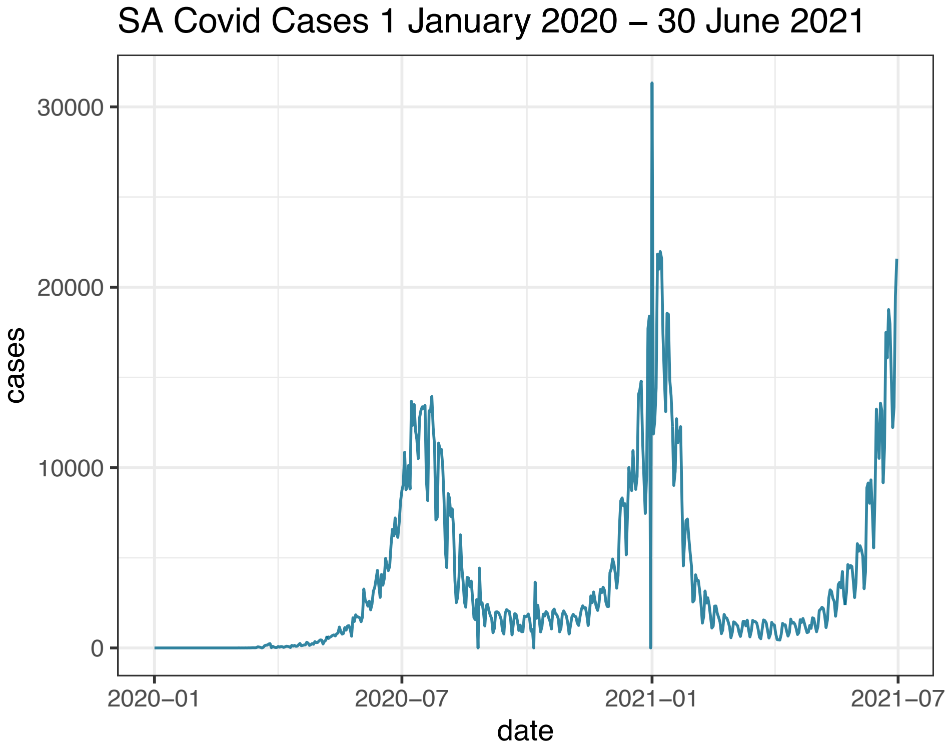

Figure 1: Cases from 1 January 2020 to 30 June 2021

3 Methodology

The methodology consisted of two distinct sections. The first being the ARIMA modelling where the data were wrangled to fit the requirements of the model, tests for stationarity using the Augmented Dickey Fuller (AFD) tests were carried out, autocorrelation functions (ACFs) and partial autocorrelation functions (PACFs) were plotted and a models were fit. The second section contains an overview of the LSTM models and four subsections. The first subsection described how the data were wrangled and data arrays created for the univariate LSTM models, the second describes how the data were wrangled and data arrays created for the multivariate LSTM, the third describes how the hyperparameter tuning was performed, and the fourth subsection describes how Google Cloud Compute was used for hyperparameter tuning. The performance of all models was determined by the MSE given by

$$\text{MSE} = \sum_{i=1}^{N}(P_i - A_i)^2 \tag{1}$$

where N is the number of observations predicted, which equals 14 in all cases, P_i is the predicted value for the observation, and A_i is the actual value for the observation.

3.1 ARIMA

ARIMA or Autoregressive Integrated Moving Average models are a class of time series models for modelling stationary inputs with stochastic shocks in the form of

$$X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} - \ldots - \phi_p X_{t-p} = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \ldots - \theta_q \varepsilon_{t-q},$$

where ϕ_i and θ_i are modelled on the data (Wei, 1989). The methodology of estimating the ARIMA order of the follows from Wei (1989). In order for the cases time series to be modelled using an ARIMA model, it was first centred by subtracting the mean and dividing by the standard deviation. Once the data is wrangled, the order of integration needs to be determined by performing the ADF test for stationarity. Figure 2 shows the ARIMA data in levels, first differences and second differences and table 2 shows the results of the ADF tests. From figure 2, in levels the ARIMRA data does not appear stationary, while both the first differenced and second differenced ARIMA data plots have portions of larger values that coincide with the first and second waves. From table 2 the null hypothesis of non-stationarity cannot be rejected for the ARIMA data in levels while it can be rejected for both the first and second differenced data.

Table 2: ADF test results

test	statistics	p_values
Levels	-1.614	0.74
First Difference	-5.635	0.01
Second Difference	-17.824	0.01

In order to to determine the correct ARIMA order, the ACFs and PACFs of the levels, first and second differenced data were inspected. Figure 3 shows the in levels the data was non-stationary and shows a high level of persistence. The first differenced ACF shows a high level of seasonality every seventh period, additionally, the majority of the points in the ACF and PACF plots are not between the blue dotted lines indicating they values are statistically significantly non-zero. This could indicate non-stationarity. The second differenced ACFs and PACFs show similar results with seasonality but with more ACF values between the dotted lines. Although the plots indicate the data may not be stationary, since this is the benchmark model, modelling will continue. From table 3, the x's, which indicate non-significant p-values for ARIMA order, are distributed across the table indicating that the model may be non-stationary. This follows from the ACFs and PACFs and an ARIMA model with order of integration of 1 is a misspecification. From table 4, ARIMA(2,2,2), ARIMA(3,2,3) and ARIMA(4,2,4) seem plausible and are compared in section 4.

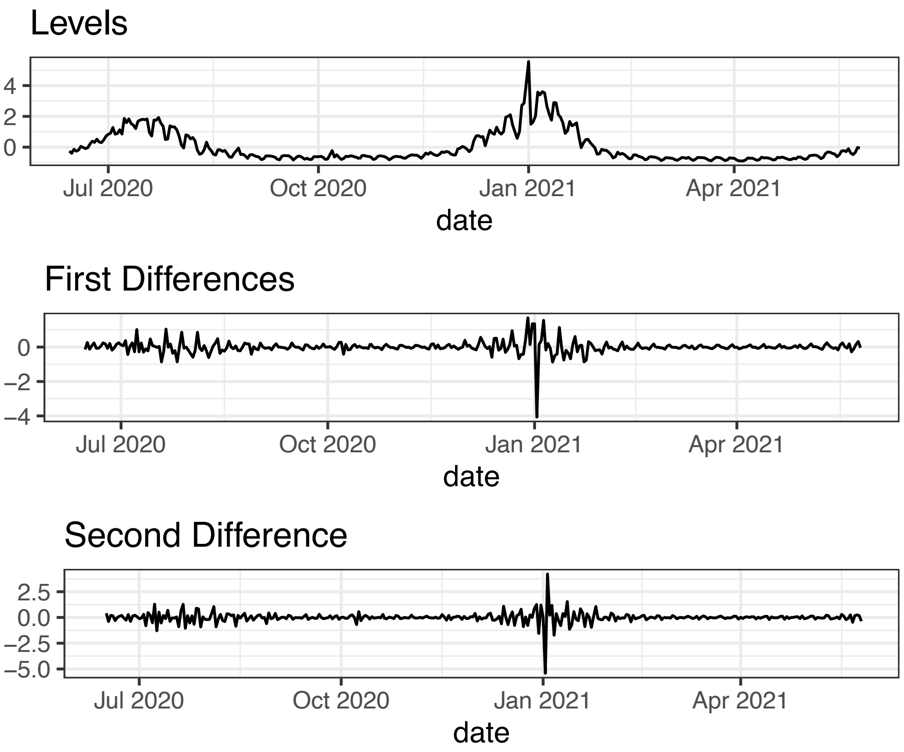

Figure 2: Plot of ARIMA data in levels, first differences and second differences

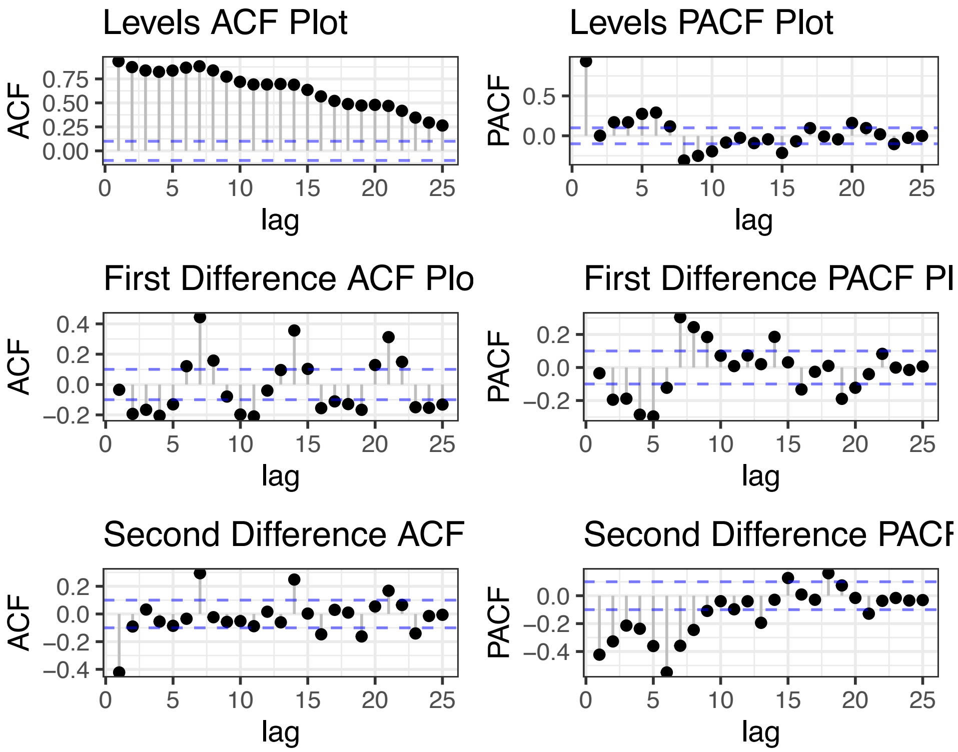

Figure 3: Plot of ACFs and PACFs for data in level, first differences and second differences

Table 3: First differenced extended ACF

	AR/MA					
	0	1	2	3	4	5
0	o	x	x	x	x	x
1	x	o	o	o	x	o
2	x	x	o	o	o	o
3	x	x	x	o	o	o
4	x	x	o	o	o	o
5	x	x	o	o	x	o

Table 4: Second differenced extended ACF

	AR/MA					
	0	1	2	3	4	5
0	x	o	o	o	o	o
1	x	x	o	o	o	o
2	x	x	o	o	o	o
3	x	x	x	o	o	o
4	x	x	o	x	o	o
5	x	o	o	x	o	x

3.2 LSTM

LSTM models are a form of RNN which perform well on predicting time series data. LSTM models overcome the issue of vanishing gradients that occur when RNN models deal with long input sequences. LSTM models overcome the issue by including a forget gate to the usual input and output gates of RNN models. Following Yudistira (2020), each LSTM memory cell consists of a memory cell c_t and working cell h_t. Memory cells are controlled by forgetting gates f_t which determines the retention of the sequence. Each memory cell has output as working memory h_t, and output gate o_t that controls the portion of c_t to be remembered. The input gate i_t controls the portion of previous working memory h_{t-1} and current input x_t to be remembered in the memory cell. The previous memory cell state h_{t-1} and the current input x_t are fed into the hyperbolic tangent, $tanh$, activation function. Figure 4 shows the structure of a LSTM memory cell where

$$\begin{aligned}
f_t &= \sigma(w_f \times [h_{t-1}, x_t] + b) \\
i_t &= \sigma(w_i \times [h_{t-1}, x_t] + b_i) \\
C_t &= tanh(w_c \times [h_{t-1}, x_t] + b_c) \\
c_t &= f_t \times c_{t-1} + i_t \times c_t \\
o_t &= \sigma(w_o \times [h_{t-1}, x_t] + b) \\
h_t &= o_t \times tanh(c_t)
\end{aligned}$$

The neural network architecture follows from Alice (2020), with both the univariate and multivariate LSTM models contain two LSTM hidden layers which each with 50 hidden units and a linear output layer. The hard-sigmoid recurrent activation function is used and a dropout rate of 0.1 is used to avoid overfitting. There are many different featuers and configurations to consider when creating LSTM models. In order to ensure that the best model is chosen for the specific structure of the data, hyperparameter tuning is required. Of the many hyperparameters, the loss function, optimizer and the number of epochs were considered. The loss function determines the performance of the model and thus how the parameters are fit to the data. Here MSE as described in equation 1 and mean absolute error (MAE) given by

$$\text{MAE} = \sum_{i=1}^{N} |P_i - A_i| \tag{2}$$

where N is the number of observations, P_i is the predicted value and A_i is the actual value of the observation were considered. The epochs refers to how many times the model cycles through the data set while training. This is in order to create a better generalization of the data and improve the fit of the model. However, too many epochs can result in overfitting thus hyperparameter tuning is required. The optimizer refers to the method which is used to update the weights in the hidden layers of the model while training. There are many different optimizers that are based on gradient descent. Here the adam optimizer which is a form of stochastic gradient descent that includes first order and second order moments, and the SGD optimizer, which is a version of gradient descent with momentum, are considered.

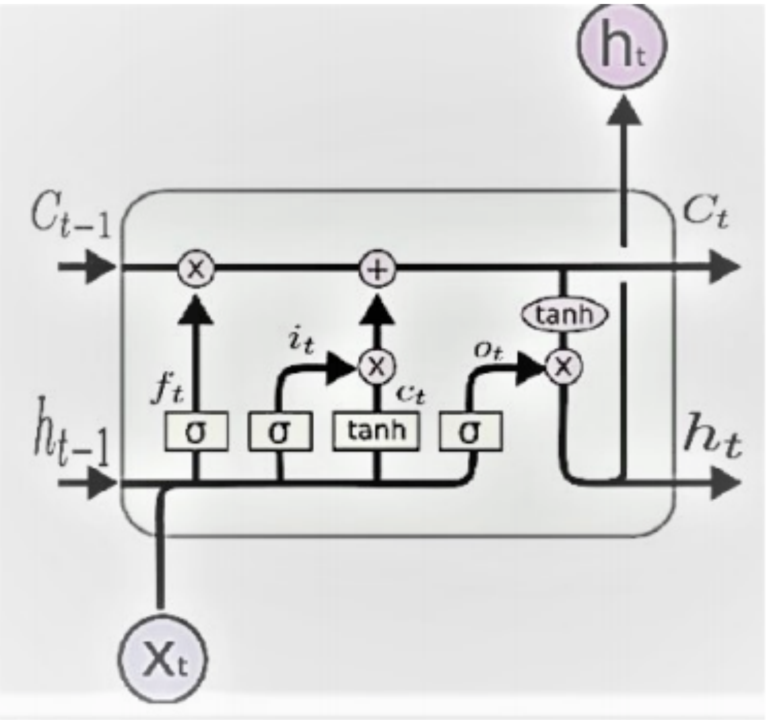

Figure 4: LSTM memory cell (Yudistira 2020)

3.2.1 Univariate LSTM

The univariate LSTM is fit on the single cases time series. Before the LSTM model can be fit, the data must be wrangled. Data is rescaled using min-max scaling following

$$Z_t = \frac{X_t - min(X)}{max(X) - min(X)}$$

where the minimum and maximum are determined from the entire time series. Figure 5 shows that the rescaled data increases the prominence of the peaks of the waves which are the most important aspect to model. As the aim of the model is to predict a 14 day period, the data is split into training variables and target variables. The training variables are 14 day periods and the target variables are the following 14 day periods. The training variables follow each other thus a target variable will be the next training variable. In total there are 22 training variable and target variable blocks. The training variables consist of data from 13 June 2020 to 8 May 2021 and the target variables consist of data from 28 June 2020 to 22 May 2021. The fitted model is used to predict the 14 day period 2 June 2021 to 15 June 2021 using 19 May 2021 to 1 June 2021 as the predictors. The MSE from the predictions and the validation set are used to determine the performance of the model.

3.2.2 Multivariate LSTM

The multivariate LSTM included, in addition to new cases, new persons vaccinated, the first difference of cumulative tests to create a new tests variable, and new deceased. Figure 6 shows the rescaled time series' of the new variables using min-max rescaling. The figure also shows the cross correlation function (CCF) plots of the new variables and new cases. New tests has its greatest cross correlation at zero which indicates the series are closely linked. Both new persons vaccinated and new deceased have the largest cross correlations at lags less than negative 10.

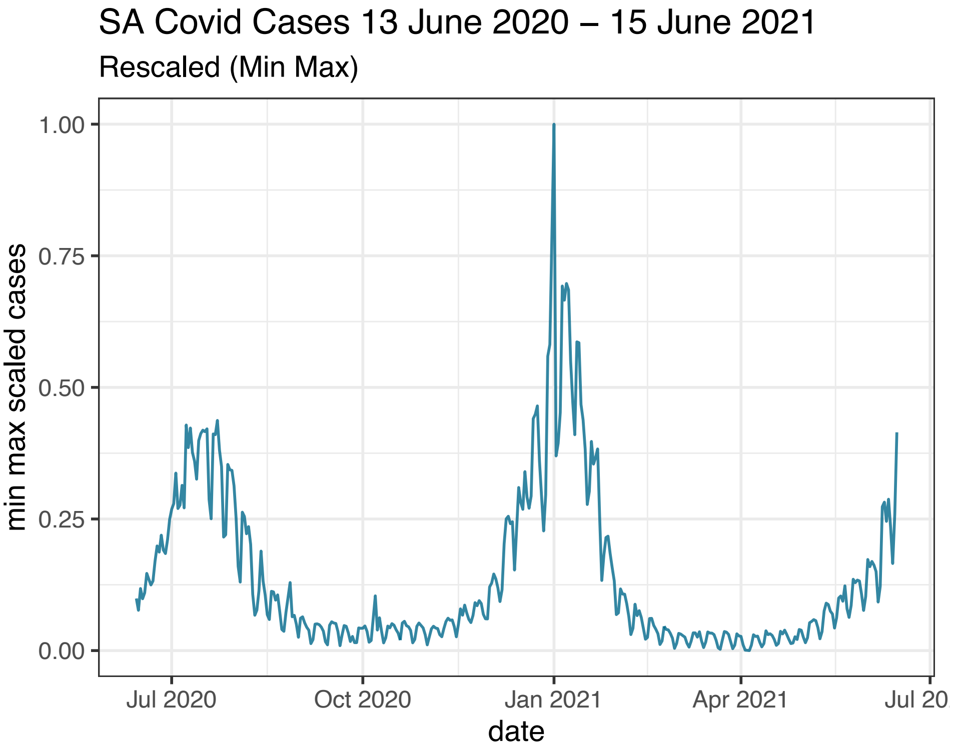

Figure 5: LSTM data with min-max rescaled cases

The training variables were created similarly to the univariate training variable. The new tests data was wrangled by interpolating an NA values from the cumulative tests time series and then taking the first difference of the series. The new vaccinated persons time series was wrangled by replacing NA values with 0 values. The new deceased time series was not wrangled. The wrangled times series followed the same process as applied to the cases time series in the univariate training set to create new 14 day long blocks of predictors. The predictors were then combined into an array with 4 levels with each level containing 22 blocks of training variables and 22 blocks of target variables.

3.2.3 Hyperparameter Tuning

For both the univariate and multivariate LSTM models, hyperparameter tuning was used to determine the best models. The hyperparameters that were tuned were number of epochs which was tuned over a range from 50 to 1050, optimizer which was tuned between the adam and SGD optimizers, and the loss function was tuned between MSE and MAE as described in equation 2. This resulted in a hyperparameter grid with 84 rows. The model with the lowest MSE from the predictions of the validation set were selected as the best models. To ensure reproducibility, the tensorflow model's seeds were set to 2021 before each model was fit.

3.2.4 Google Cloud Compute

In order to improve computational speed and allow for more extensive hyperparameter tuning, especially for the multivariate LSTM, a Google Cloud Engine Virtual Machine was used for hyperparameter tuning. Following McDermott (2021), a n1-standard-16 virtual machine with 16 vCPUs was spun up in zone europe-west2-a using the code attached in the appendix under Google Cloud Compute. On the machine, R and rstudio were installed, a python virtual environment was created and keras and tensorflow were installed and configured. While, the GPU version of tensorflow was possible to be configured with the correct CUDA

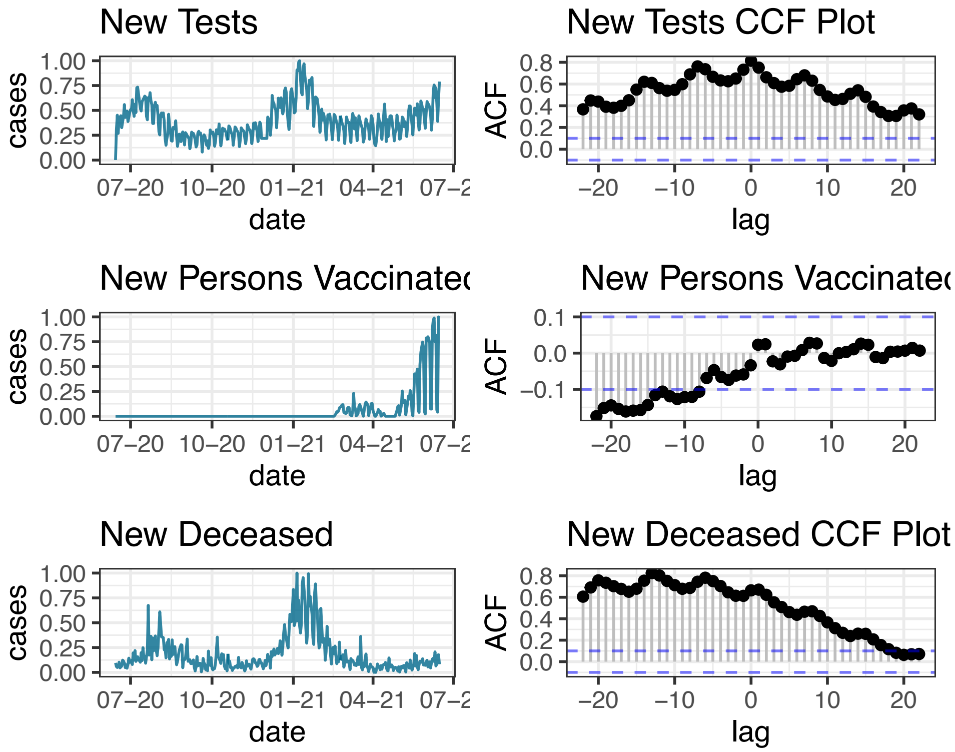

Figure 6: Levels and cross correlation plots of new tests, new persons vaccinated and new deceased

drivers, it was unable to be operated through rstudio on the virtual machine. As a result the hyperparameter tuning process took roughly two hours to complete for each of the univariate and multivariate LSTM models.

4 Results

4.1 ARIMA

Following section 3, three ARIMA orders were identified as possible fits. In order to determine the best fitting model, the three orders are fit to the data and used to predict 14 days worth of cases. The MSE of the predicted and actual cases were compared. Table 5 shows that the ARIMA(4,2,4) model had the lowest MSE of 5 223 898. Figure 2 shows that the ARIMA(4,2,4) model was able to fit the cyclical nature of the cases but is unable to predict the sharp uptick in cases following the 13th of June. Thus the model is unable to predict the third wave.

Table 5: Results of ARIMA ordering

ARIMA_Order	MSE
(2,2,2)	12187160
(3,2,3)	6008036
(4,2,4)	5223898

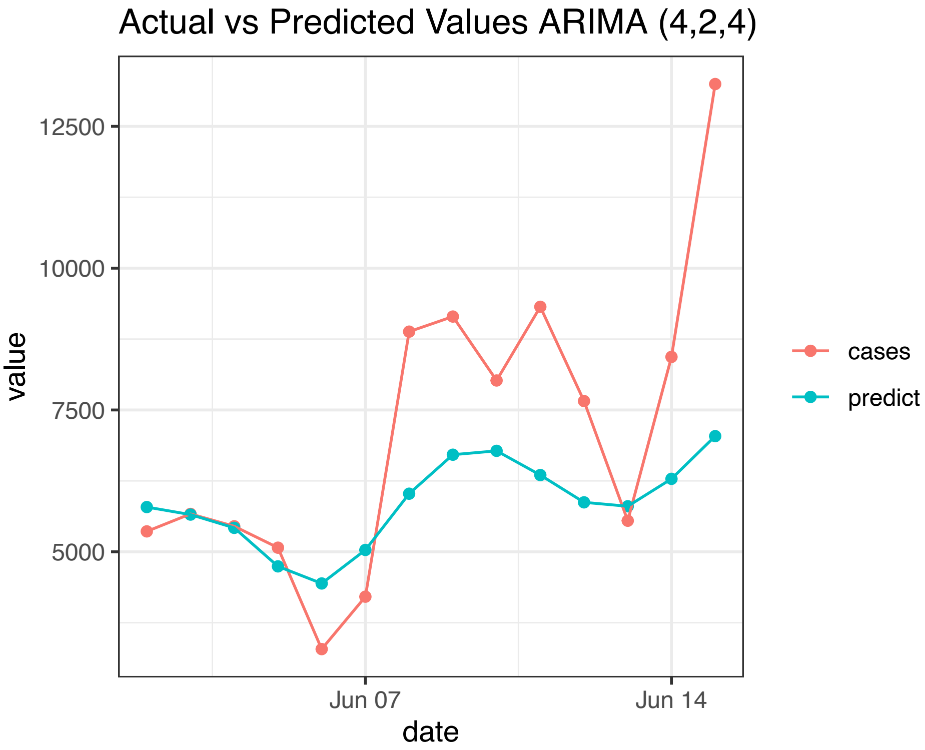

Figure 7: Actual cases and ARIMA(4,2,4) predicted cases

4.2 Univariate LSTM

The univariate LSTM models were trained on the univariate training data as described in section @ref{Methodology}, where the predictors are 14 day blocks of new cases and the target variables are the following 14 days. Hyperparameter tuning was used to determine the best model as in the model with the lowest MSE on the validation set as described in section @ref{Hyperparameter}. A grid search was conducted, with the hyperparameter grid consisting of 84 combinations of hyperparameters. Tuning was performed on a Google Cloud Compute Virtual Machine as described in section @ref{Google}. As seen in table 6, the best model produced a MSE of 1 008 802 when using MSE loss function, the adam optimizer, and 450 epochs. The best model was closely followed by a model using the MAE loss function, adam optimizer, and 550 epochs. Due to the variability in starting weights, in order to perform exhaustive hyperparameter tuning many different seeds and more combinations of hyperparameters would be necessary. Figure 8 shows the 14 day prediction from the validation set of the best univariate LSTM model when compared to the actual cases. The plot shows that the model was able to predict both the cyclical shape of the cases as well as the sudden uptick in cases associated with the thrid wave.

Table 6: Top 5 univariate LSTM models with lowest MSE

X	loss	optimizer	epochs	mse
34	mse	adam	450	1008802
41	mae	adam	550	1039631
33	mae	adam	450	1345770
30	mse	adam	400	1393256
61	mae	adam	800	1493986
49	mae	adam	650	1552784

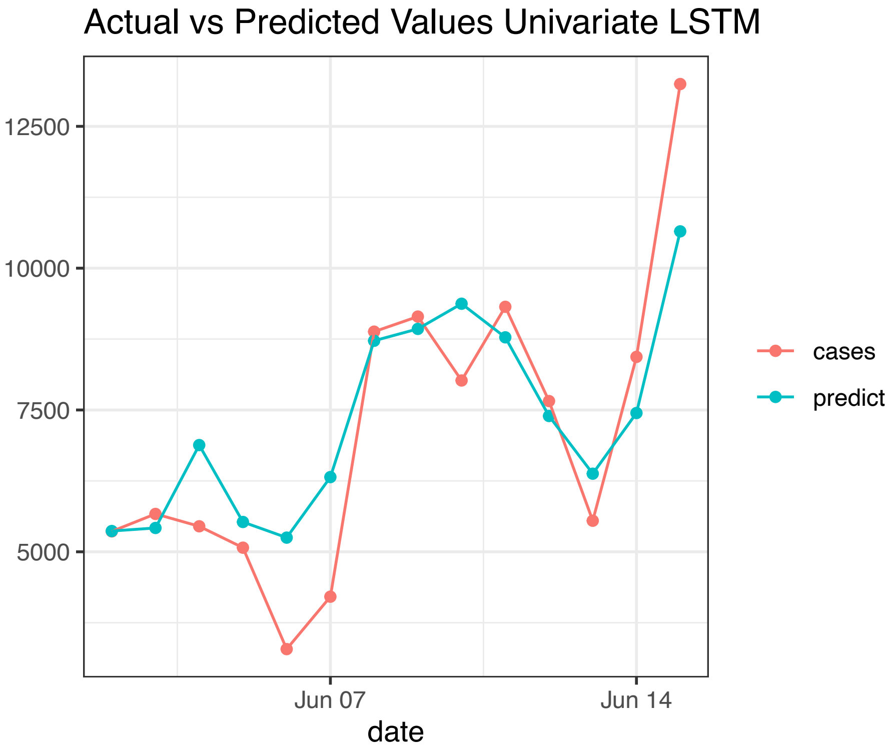

Figure 8: Best performing univariate LSTM model predictions

4.3 Multivariate LSTM

The multivariate LSTM models were trained on the multivariate training data as described in section @ref{Multivariate}, where the predictors are 14 day blocks of new cases, new tests, new persons vaccinated, and new deceased, and the target variables being the 14 day block new cases following the predictor block. Hyperparameter tuning was used to determine the best model as in the model with the lowest MSE on the validation set. A grid search was conducted as described in section @ref{Hyperparameter}, with the hyperparameter grid consisting of 84 combinations of hyperparameters. Tuning was performed on a Google Cloud Compute Virtual Machine as described in section @ref{Google}. As seen in table 7, the best model produced a MSE of 1 565 288 when using MSE loss function, the adam optimizer, and 550 epochs. This result is substantially worse than the best univariate model. The worse performance of the multivariate LSTM could be due to the increased noise of adding three more variables, the variables not being highly correlated with the target (as seen in the CCF plots), or the model being unable to identify the relationship between the four variables and the target variable. Figure 9 shows the 14 day prediction from the validation set of the best multivariate LSTM model when compared to the actual cases. The plot shows that the model was not able to closely match both the cyclical nature of the cases nor the sudden uptick in cases. This explains the higher MSE from the multivariate model when compared to the univariate model.

Table 7: Top 5 multiivariate LSTM models with lowest MSE

X	loss	optimizer	epochs	mse
21	mse	adam	550	1565288
26	mae	adam	650	1695926
42	mae	adam	1050	2933335
34	mae	adam	850	3131851
38	mae	adam	950	3337986
25	mse	adam	650	3782085

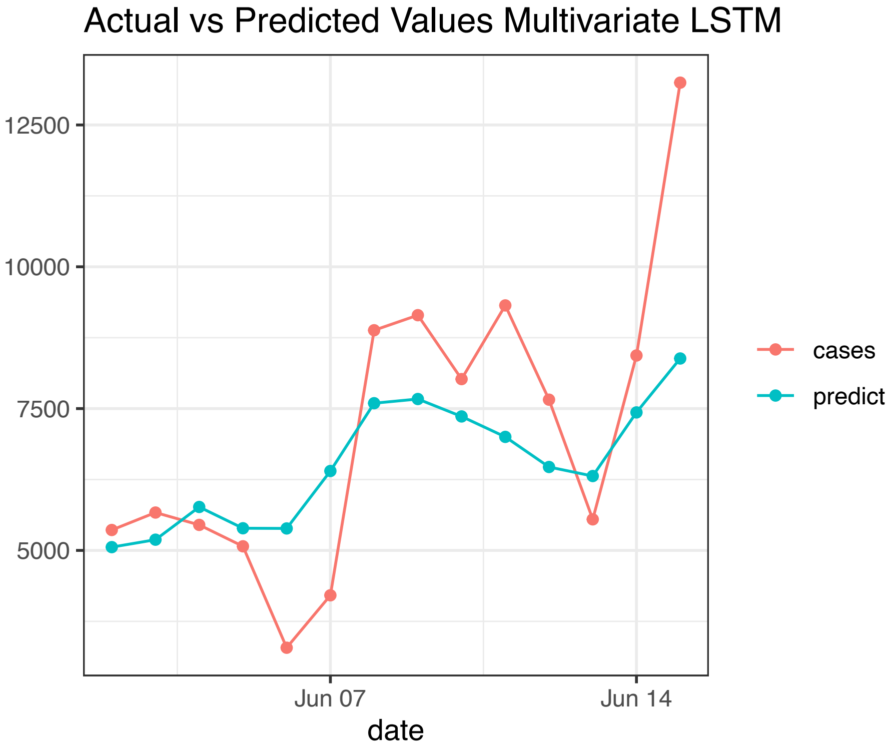

Figure 9: Best performing multivariate LSTM model predictions

4.4 Best Model and Test Set Prediction

Table 8 shows that results from the best performing model of each type. The worst performing model was the ARIMA(4,2,4) model with a MSE of 5 223 898. This is expected as the benchmark model and with ARIMA models generally not being suited to model this type of data. The second best model is the multivariate LSTM model with a MSE of 1 565 288. The model performed well and although was able to model the cyclical nature of the cases, it could not model the sudden uptick in the thrid wave. The best performing model is the univariate LSTM with MSE of 1 008 802. The model was able to both model the cyclical nature of cases as well as the sudden uptick in the third wave.

The best model, the univariate LSTM model, was fit to the test data set. The performance is shown in table 9 with a MSE of 19 025 366 which is substantially greater than the validation set performance. Figure 10 shows that the model is able to predict the cyclical nature of the case but predicts a much larger uptick in cases towards the end of the 14 day period when compared to the actual cases. Another reason for the poor performance is the magnitude of the predictions and actual cases is far higher with between 7500 and 20 000 cases compared to 2 500 and 13 000 cases.

Table 8: Table of results from best model of each type of model

	Model	MSE
1	Univariate LSTM	1008802
2	Multivariate LSTM	1565288
3	ARIMA(4,2,4)	5223898

Table 9: Performance of best model on test data

Model	MSE
Univariate LSTM	19025366

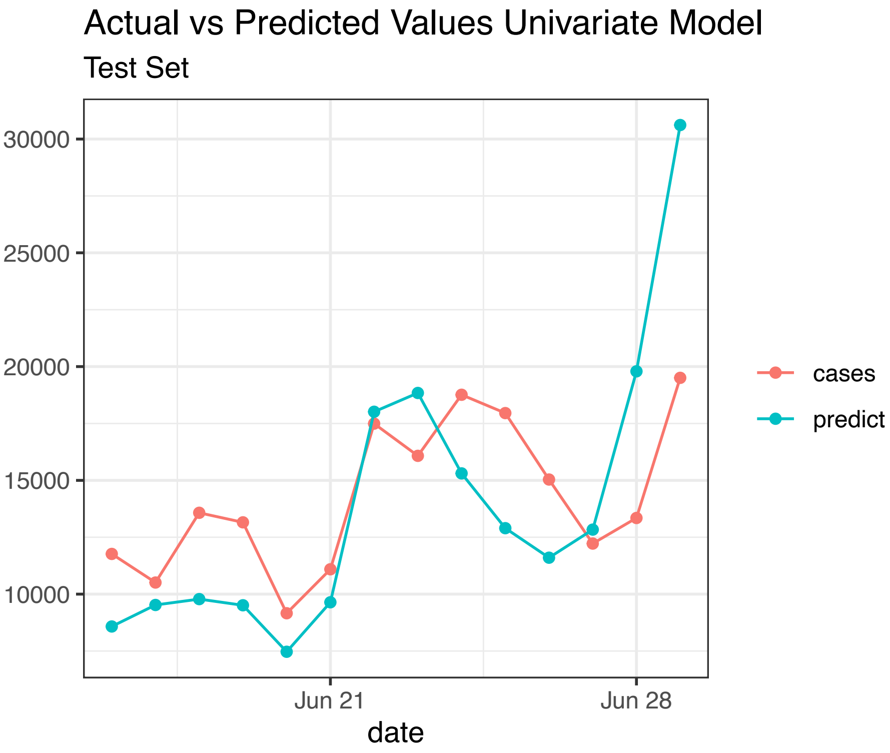

Figure 10: Predictions and cases for test data using best model

5 Conclusion

COVID-19 infections show a pattern of waves and lulls. This pattern results creates difficulties for traditional time-series prediction techniques. This is shown in the poor performance of the ARIMA(4,2,4) model. LSTM models are able to change predictions dependant on the state of the inputs. This allows for the models to differentiate between waves and lulls. This is shown in the improved performance of the LSTM models compared to the ARIMA models. The univariate models produced the best predictions compared to the

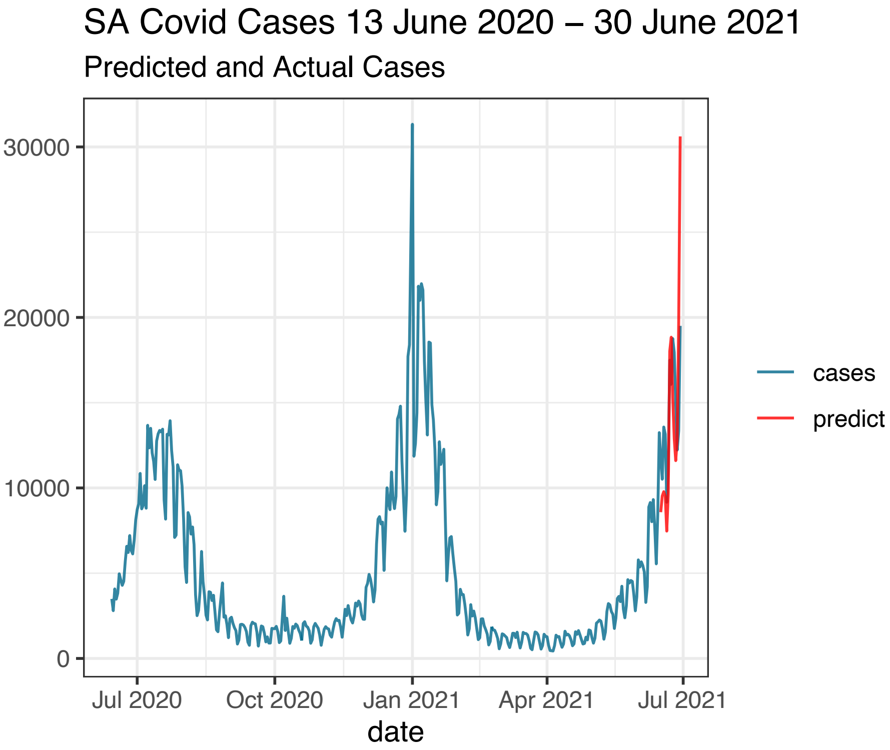

Figure 11: Test set predictions and actual cases for entire time series

multivariate models. This could be due to increased noise, low correlation between predictors and target variables, or incorrect architectures or hyperparameters for the multivariate models. The univariate models were able to predict both the cyclical nature of the cases and the increasing trend of the waves. The best univariate model produced a test mse of 19 025 366 which results in roughly 312 cases per day being incorrectly predicted.

The impact of lockdowns and other restrictions by the South Africam government were not included in the above models as the impact of them should be included in the cases data. In future, hyperparameter tuning over a larger grid, using Bayesian Optimization or changing the LSTM architectures by adding hidden layers or hidden units could improve results. Furthermore, creating a distribution of predictions based on many different starting weights would be necessary to create more robust predictions. Both of these changes would require many more models being fit which would increase the number of computations. The computational load could be reduced using the GPU version of tensorflow and further Google Cloud Compute Virtual Machines or Docker.

6 Bibliography

Alice. 2020. LSTM time series predictions in r. *Data Side of Life.* (January). [Online], Available: http://datasideoflife.com/?p=1171.

McDermott, G. 2021. Lecture 14: Google compute engine (part i), data science of economists. *University of Oregon.* [Online], Available: https://github.com/uo-ec607/lectures/tree/master/14-gce-i.

Suárez-Cetrulo, A.L., Kumar, A. & Miralles-Pechuán, L. 2021. Modelling the COVID-19 virus evolution with incremental machine learning. *CoRR.* abs/2104.09325. [Online], Available: https://arxiv.org/abs/2104.09325.

Wahltinez, O. & others. 2020. COVID-19 open-data: Curating a fine-grained, global-scale data repository for SARS-CoV-2. [Online], Available: https://goo.gle/covid-19-open-data.

Wei, W. 1989. *Time series analysis: Univariate and multivariate methods*. Vol. 33.

Yudistira, N. 2020. COVID-19 growth prediction using multivariate long short term memory. *arXiv e-prints*. (May):arXiv:2005.04809. [Online], Available: http://arxiv.org/abs/2005.04809.