Predicting COVID-19 Infections in South Africa Using Deep Learning

CL Ngema

orcid.org 0000-0003-3812-049X

Dissertation accepted in fulfillment of the requirements for the degree *Master of Science in Computer Science* at the North-West University

Supervisor:	Dr RP Ntema
Co-supervisor:	Dr E Sonono
Co-supervisor:	Dr B Sidumo

Acknowledgements

Special thanks to my supervisor Dr Piet Ntema as well as my co-supervisors Dr Energy Sonono and Dr Bonelwa Sidumo, for their priceless advice, patience, mentorship, and unwavering support in bringing my research to fruition. Additionally, I am appreciative to my supervisors for their valuable insights and constructive feedback.

A special thanks also goes towards my family and associates for their constant encouragement and motivation. Their unwavering support and belief in me would not have made this possible.

Thank you all for your support and inspiration.

Abstract

In this work, the main aim was to employ deep learning techniques for predicting coronavirus disease 2019 (COVID-19) infections in South Africa, with a focus on enhancing the predictive capacity of COVID-19 virus new infections. Since the surge of COVID-19 infections, there have been concerns about the potential effects it might have in the healthcare sector, particularly in efficiently managing resource availability and allocation. Various researchers conducted different studies to predict the transmission of COVID-19 infections using techniques such as statistical predictive analysis, mathematical modeling, and machine learning approaches. Studies using mathematical and statistical modeling had limitations in predicting future infection waves as they were unable to predict future waves of infections, unlike those using machine learning. Hence, the main aim of undertaking this study was to explore the feasibility of employing a specific subset of machine learning, specifically deep learning, to predict COVID-19 infections in South Africa. The COVID-19 dataset used to build the deep learning models was sourced from an open data repository. To achieve the aim, the first step was to train the basic long short-term memory (LSTM), stacked LSTM, and bidirectional LSTM models. LSTM-based models have demonstrated remarkable capability in capturing temporal dependencies, while the stacked LSTM and the bidirectional LSTM architectures enhance this capability by incorporating additional layers and bidirectional information flow, respectively. The next step was to empirically test the trained models on a COVID-19 real dataset to check their performances based on the root mean square error (RMSE). A persistence model that was used as a baseline model to evaluate performances for the complex models was introduced. Comparing the predictions of the trained models with the baseline model, only the basic LSTM model had an RMSE lower than the persistence model. Lastly, identify gaps and challenges in this study and provide recommendations. The primary challenges and gaps identified include the fact that the predictions were made for South Africa overall and did not cater to subpopulations. This study concluded that the proposed deep learning models exhibited enhanced predictability capacity of COVID-19 new infections compared to previous studies that were reviewed. Additionally, the basic LSTM model was found to be the best model for predicting COVID-19 infections in South Africa as it had the lowest RMSE.

Keywords: COVID-19; Infection; Deep Learning; Prediction; Basic LSTM; Stacked LSTM; Bidirectional LSTM.

Contents

List of Figures

List of Tables

List of abbreviations

A list containing all abbreviations that will be used throughout text.

AI	Artificial intelligence
ANN	Artificial neural network
COVID-19	Coronavirus disease
CSV	Comma-separated values
ED-LSTM	Encoder decoder long short-term memory
GMM	Gaussian mixture models
LSTM	Long short-term memory
MAE	Mean absolute error
MSE	Mean squared error
NN	Neural network
PCA	Principal component analysis
ReLu	Rectified linear unit
RMSE	Root mean squared error
RNN	Recurrent neural network
SA	South Africa
SIR	Susceptible infectious removed
SEIR	Susceptible exposed infectious removed
SVM	Support vector machines
UK	United Kingdom
USA	United States of America
WHO	World health organisation

Chapter 1: Introduction and Background

1.1 Introduction

This study applied deep learning techniques to predict COVID-19 infections in South Africa, with the goal of enhancing predictive capacity of COVID-19 virus new infections. There were numerous concerns regarding the surge of infections and some researchers have done different studies in relation to the prediction of COVID-19 new infections. Some researchers have used mathematical modeling techniques, for example, Djilali and Ghanbari (2020), and others have proposed the use of statistical predictive analysis approaches, for example, Tátrai and Várallyay (2020). However, these methods have had some drawbacks as they have failed to correctly predict the second and third waves of infections. Some studies employed machine learning techniques, and the results were promising as the models were able to predict the infections, for example, Chandra et al. (2022).

Amongst some of the work done with regards to predicting COVID-19 infections is the one done by Djilali and Ghanbari (2020) where they used a mathematical modeling approach and proposed an SEIR system to model COVID-19 transmission. The geographical focus of the study was South Africa (SA), Turkey and Brazil. Based on their findings, COVID-19 was expected to disappear from the SA community in 100 days starting from 15 April 2020. However, the findings suggest that the proposed model was not effective in predicting the infections since it failed to capture the second, third, and other waves of infections as demonstrated in Figure 1.1.

Figure 1.1: Daily COVID-19 confirmed cases in South Africa (Coronavirus in South Africa, 2021)

Another study was performed by Tátrai and Várallyay (2020), where they used a statistical analysis approach to predict the COVID-19 epidemic growth in different countries and

regions. They managed to fit the logistic model in all the countries and regions of interest. However, it was not possible to make a realistic fit in some regions due to few registered infections and noise in the dataset. One of the regions had a second wave of infections which the proposed approach failed to capture.

A similar study was carried out by Tamang et al. (2020) to predict the number of rising cases and death cases in India, the United States of America (USA), France, and the United Kingdom (UK). The study used a machine learning approach. Artificial neural networks (ANN) was proposed due to its capability of learning from an experimental or real dataset to describe non-linearity and interaction effects with great success. When the forecasts were contrasted with an actual dataset, it became clear that ANNs could accurately predict future COVID-19 cases in any of the specified countries. The study had limitations in making the predictions, for instance, the predictions were made over one week and not for longer periods.

Another study by Chandra et al. (2022) employed a deep learning technique to predict COVID-19 infections in India over a time period of two months. The authors trained multiple different LSTM models to forecast infections in India and used the trained models to predict COVID-19 infections on the regions they identified as hotspot areas in India. The results from the study suggested that the proposed model, univariate random-split encoder decoder - long short-term memory (ED-LSTM) was effective in predicting COVID-19 infections as it was capable of predicting the two-months ahead wave of infections.

From the studies discussed above, it can be seen that predictive modeling techniques such as mathematical modeling and statistical modeling approaches have some shortcomings when predicting COVID-19 infections. The world was greatly affected by the COVID-19 pandemic, which made data-driven strategies for combating infectious diseases more crucial. With the availability of datasets on COVID-19, this study proposed applying machine learning models that can accurately predict and diagnose the disease. Importantly, these models can also be extended and adapted to other infectious diseases, providing a powerful tool for responding to future pandemics. By leveraging the insights gained from COVID-19 datasets, researchers and public health officials can develop more effective strategies for preventing and managing future outbreaks.

1.2 Background

In Wuhan City - China, reports of inexplicable pneumonia cases were recorded in December 2019. It was later announced that the cases were a result of a novel coronavirus which was named COVID-19 (coronavirus disease 2019, illustrated by Figure 1.2) by the World Health Organisation (WHO).

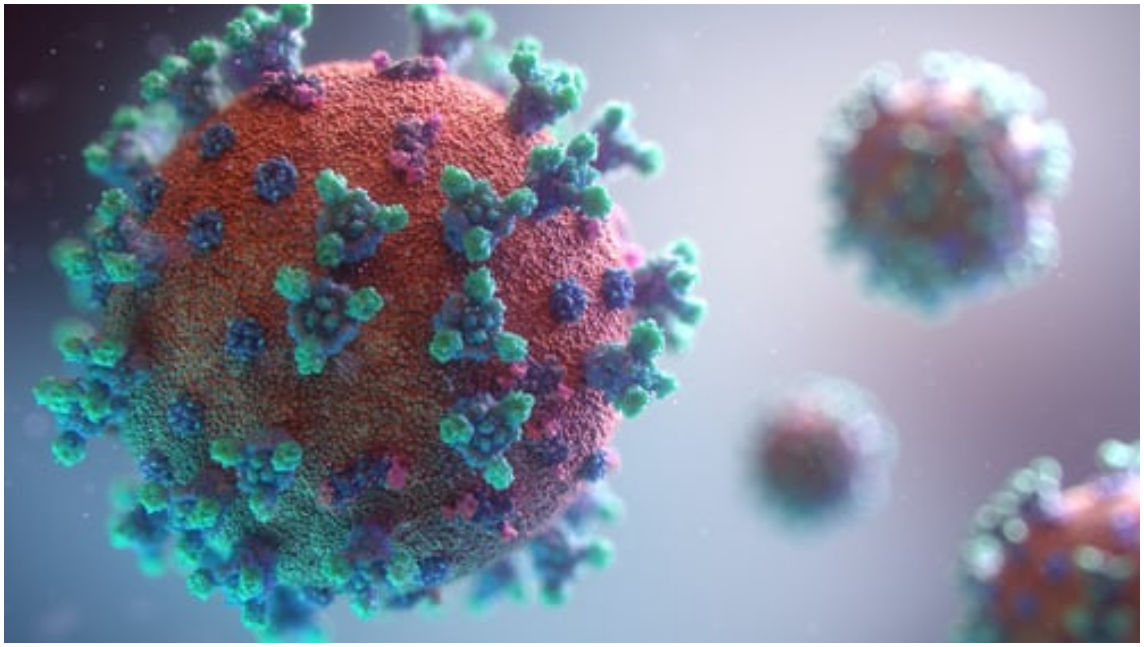

Figure 1.2: COVID-19 Image (Fusion Medical Animation, 2020)

Only a few of the hundreds of viruses that comprise the coronavirus family have been related to mild to severe respiratory tract infections in humans. Humans who have mild to severe respiratory tract infections have been connected to coronaviruses such as human coronavirus 229E, human coronavirus NL63, human coronavirus OC43, human coronavirus HKU1, severe acute respiratory syndrome coronavirus (SARS-CoV), and middle east respiratory syndrome coronavirus (MERSCoV) (Lone and Ahmad, 2020). First reports of SARS-CoV and MERS-CoV were made in November 2002 and September 2012, respectively (Lone and Ahmad, 2020). These two viruses caused severe respiratory illnesses and significant fatality rates after emerging in animal reservoirs.

COVID-19 was caused by a novel severe acute respiratory coronavirus-2 (SARS-CoV 2) and the WHO classified COVID-19 as a pandemic because of its rapid spread and the seriousness of the clinical outcomes it was associated with. On 5 March 2020, news of the virus' first case in South Africa was released (Mbuvha and Marwala, 2020).

Initially, there was no specific medical intervention as well as preventive vaccination options available known against this virus. After the first case was reported in South Africa, the government tried to minimise the increase in the number of COVID-19 cases by putting some measures in place. One of the measures was placing the country under complete lockdown for 21 days which was later extended by further 2 weeks. While the country was under complete lockdown, the government had time to improve the healthcare facilities in the country and prepare for the surge in COVID-19 cases.

As part of the lockdown, some measures were put in place and the measures played a crucial role in slowing down the rapid spread of COVID-19, although they had a devastating effect on the South African economy. According to data from the Labour Force Survey,

South Africa's employed population fell by 2.2 million in the second quarter of 2020 compared to the first, which raised the nation's already high unemployment rate (Bhorat et al., 2020). People lost jobs as some businesses retrenched staff due to the losses incurred during the lockdown when they were unable to operate. The small business sector was also negatively affected by the lockdown, for example, the vendors at train stations and taxi ranks lost the majority of their customers as fewer people commuted to work during the lockdown. Some workers like hairstylists were also not allowed to work as part of the restrictions that were placed and these workers were without an income for months (Odeku, 2021).

As time progressed, there was a breakthrough in vaccine development for COVID-19. Numerous vaccines were successfully developed by various studies in different institutes and approved by the WHO. The South African government managed to secure doses of the AstraZeneca vaccine. However, its rollout was put on hold after new information came to light. This information revealed that the AstraZeneca vaccine was found to be less effective against the coronavirus beta variant (501Y.V2 variant). Consequently, further research was deemed necessary in South Africa (Cohen, 2021). After that, the government managed to secure COVID-19 doses from the COVAX facility, Johnson and Johnson as well as Pfizer. Out of the 59 million people living in South Africa, around 7 million have received vaccinations to date. Of those vaccinated, 1.7 million received the Johnson and Johnson vaccination, whereas the remaining individuals received the Pfizer vaccine (BBC News, 2021).

Despite the fact that a certain percentage of South Africa's population received vaccinations, it was not enough to achieve herd immunity. The government proposed containment, which aimed to immunise just enough people so that COVID-19 patient admissions were no longer putting a strain on the country's health system. Studies by Madhi (2021) showed that this virus was constantly mutating and prior exposure did not provide immunity. Furthermore, vaccinating did not prevent future re-infection from the different mutations of the virus.

With time, all viruses undergo mutations, including COVID-19. The majority of modifications barely alter the virus' properties. Nevertheless, depending on the virus, certain modifications may have an impact on the virus' characteristics, such as its ease of transmission, related health effects like the severity of the illness, the effectiveness of vaccinations and medications, etc. COVID-19 also had a couple of different variants associated with it, some having severe health implications for those affected when compared to other variants. One of the COVID-19 variants that was of concern was the Omicron variant (Madhi, 2021).

1.3 Problem Statement

There was a worldwide public health emergency due to COVID-19. There were concerns relating to the continuing surge in the number of new infections, particularly in South Africa. The concerns are common in different sectors including the health sector, the business sector as well as small businesses. The major concerns around the health sector were due to limited hospital resources such as hospital beds and oxygen tanks. A rapid increase in the number of infections and people requiring hospitalisation would overwhelm the healthcare system, leading to a shortage of resources for those in need (Blumberg et al., 2020).

Predicting COVID-19 infections has become a topic of interest for researchers, and various techniques have been proposed to aid decision-makers in their decision-making processes. The study by Tamang et al. (2020), which utilised an artificial neural network curve fitting technique to make predictions for COVID-19 infections, assumed that the infections in all countries would follow a similar trend to that of China or South Korea. However, the study only made predictions for one week, and it did not extend the time frame to observe how that would affect the model's performance or evaluate the proposed methodology as a reliable technique for making COVID-19 predictions

Another study by Chandra et al. (2022) employed a deep learning technique to predict COVID-19 infections in India over a time period of two months. The main limitation of the study was that only states with COVID-19 hotspots in India were considered when training the model and making predictions. There was no sufficient information available on how the model performs when making predictions for infections in states that are not COVID-19 hotspots.

Although the topic at hand pertains to health-related aspects, it is crucial to note that the primary objective or main emphasis of this study was not focused on health. Instead, the health-related examples served as an incidental illustration within a broader context. The intention was to provide a specific case that could be easily understood and related to while recognising that the study encompassed a wider range of subjects and implications beyond the health domain. By incorporating this incidental health example, the study aimed to enhance comprehension and highlight the applicability of using deep learning techniques for forecasting.

To be better prepared, predictive models should be available to the decision-makers in South Africa. The predictive analysis as well as modelling for COVID-19 infections has been an area of interest for some researchers. Different researchers have proposed techniques that can be used to predict COVID-19 infections in South Africa and other parts of the world. Studies discussed in the introduction section above, it can be seen that other predictive modeling techniques, such as mathematical and statistical modeling approaches, have some shortcomings when predicting COVID-19 infections. Studies proposing the use of machine learning-based approaches in other geographical locations

have been successful, for example, the study by Tamang et al. (2020). Therefore, this study proposes the use of machine learning techniques, particularly deep learning, to predict COVID-19 infections in South Africa using a dataset similar to those applied by Chandra et al. (2022) in their study. The intention is to assist the healthcare sector in being better prepared, alerting the general public to potential surges, and providing insights to guide policy decisions, resource allocation, and risk management strategies across different sectors, fostering resilience and minimising disruptions in non-health domains.

1.4 Research Aim and Objectives

1.4.1 Research Aim

To employ deep learning techniques for predicting COVID-19 infections in South Africa, with a focus on enhancing predictive modeling.

1.4.2 Research Objectives

The specific research objectives to be fulfilled in order to achieve the research aim are as follows:

- To perform a predictive analysis using a traditional machine learning predicting model, the Long Short-Term Memory (LSTM).
- To empirically test the proposed LSTM models on a COVID-19 real dataset.
- To provide recommendations on the challenges and gaps identified in this study and how they could be addressed for future studies.

1.5 Research Methodology

As discussed in the previous sections, this study proposed to use deep learning techniques to predict COVID-19 infections in South Africa. This study focused on predicting the number of COVID-19 infections over time using historical data. For this study, a publicly available COVID-19 dataset was used to make the predictions. This study used deep learning techniques to predict COVID-19 infections in South Africa and all the modeling was done on Python.

1.5.1 Study Context

The study took place in South Africa and employed the use of deep learning techniques to predict COVID-19 infections in South Africa. The performance of the models was measured using metric root mean squared error to determine the accuracy of the models when making predictions on testing data.

1.5.2 Population and Sampling

- **Population**

  In principle, the COVID-19 data population consisted of all individuals who have come into contact with the virus. This included those who have been diagnosed with COVID-19, those who have been admitted to a hospital as a result of the virus, those who have recovered from the virus and those who have passed away as a result of the virus.

- **Sampling**

  No data sampling was done in this study. The study made use of a publicly available COVID-19 dataset for South Africa. The data was obtained from the Mediahack Coronavirus web page in the form of a CSV file.

1.6 Rigour and Validity

In order to assess the model's dependability and performance, the root mean squared error (RMSE) metric was used to assess and validate the model. Loss or a cost function was used as a measurement of determining how well a prediction model was performing in predicting the expected outcome. The preferred loss function for neural networks applied to regression models used was the mean squared error (MSE) (Pradhan and Kumar, 2010). The commonly used loss functions for neural networks (NN) for time series forecasting problems are the mean squared error (MSE), the root mean squared error (RMSE), and the mean absolute error (MAE).

1.7 Ethical Considerations

For this study, an open-source data source was used to do the modeling that will enable predictions to be made. This study went through the NWU ethics committee to be cleared.

1.7.1 Permission and informed consent

It was also essential to ensure that in the case where the data includes information that was collected with informed consent, this study ensured that the ethical guidelines established in the informed consent documents were followed. This included that the data was used for the intended purpose and the individuals who provided the data could not be identified in any way (Panigutti et al., 2022).

1.7.2 Fairness, transparency and social impact

Most deep learning models are subject to bias, which can result in unfair outcomes. To ensure fairness and prevent bias, the study can involve using diverse training data, carefully selecting features and parameters, and then testing the model on a variety of subpopulations (Safdar et al., 2020).

Another ethical consideration that was considered is **transparency**. Deep learning models can be difficult to understand, to ensure transparency, the data collection and modeling methods will be well documented and how the model makes predictions (Safdar et al., 2020).

Lastly, it was of importance to consider the **social impact** of predicting COVID-19 infections. The predictive models can have a significant impact if the decision-makers decide to utilise the models. The findings of this work will be clearly communicated in a responsible manner (Safdar et al., 2020).

1.8 Dissertation Layout

The outline of this dissertation is as follows:

Chapter 1 presents the introduction, background, problem statement, research aim and the objectives to the study.

Chapter 2 presents an extensive literature review of machine learning algorithms for predictive analysis as well as epidemiological modeling. It also outlines some advantages and drawbacks concerning these modeling techniques for prediction purposes on a COVID-19 dataset and identifies gaps in the studies.

Chapter 3 presents the research design, methodology and a detailed description of the nature and methodology applied in this study. Additionally, it presents the identified predictive modeling techniques for COVID-19 infections, along with the model performance measures utilised in this study.

Chapter 4 presents the results of this study. This chapter provides an analysis and dis-

cussion of the research undertaken.

Chapter 5 presents the conclusions and recommendations drawn from this study. This chapter concludes with remarks and recommendations for future studies, aiming to enhance areas identified for improvement within the findings.

Chapter 2: Literature Review

This chapter presents a literature review that introduces the concept of epidemiological modeling as well as machine learning. It also highlights some of the most commonly used models and algorithms that can be utilised to model the transmission of infectious diseases.

2.1 Epidemiological Modelling

The study of an outbreak's spreading behavior to forecast its trajectory and evaluate containment strategies is known as epidemiological modeling. It is a branch of mathematics and computer science. Stochastic and deterministic models are the two categories into which epidemic models can be divided. While deterministic models are mathematical representations of the spread of infectious diseases in a population without incorporating stochasticity. Stochastic models introduce randomness and are primarily used to assess the probabilistic distributions of likely outcomes by allocating random variation in multiple entries over time (Martcheva, 2015; Allen, 2008). In this section, some commonly used models under the two categories will be briefly introduced.

2.1.1 Deterministic Epidemic Models

To learn more about how an epidemic spreads throughout communities, one can simply apply any of the many classical deterministic epidemiological models that are available. The susceptible-infected-removed (SIR) model is one of the most widely used model (Cooper et al., 2020; Weiss, 2013; Gaeta, 2020). The model is a dynamical system which is given by three ordinary differential equations based upon the three following compartments:

- Susceptible $S(t)$: This compartment refers to all individuals who are not infected but could still contract the disease
- Infected $I(t)$: This compartment refers to all individuals who have been infected with the disease. They can also transmit it to those who are susceptible.
- Removed $R(t)$: This compartment refers to all who have recovered or died as a result of the virus and are assumed to be immune.

In general, the SIR model is usually based on the assumption that the population which is given by $N = S(t) + I(t) + R(t)$ is constant in time. However, Cooper et al. (2020) proposed the use of a modified SIR model that was taking the susceptible populations

as a variable which can be adjusted at various times to account for the increase in this population. Hence the SIR model is given by the following system of ordinary differential equations:

$$\begin{aligned} \frac{dS(t)}{dt} &= -\beta S(t)I(t), \\ \frac{dI(t)}{dt} &= \beta S(t)I(t) - \gamma I(t), \\ \frac{dR(t)}{dt} &= \gamma I(t), \end{aligned} \tag{2.1}$$

where β is the infection or transmission rate and γ is the recovery rate (Cooper et al., 2020). In Equation 2.1, $\frac{dS(t)}{dt}$ represents the rate of change of the number of susceptible individuals S over time t, $\frac{dI(t)}{dt}$ describes the rate of change of the number of infectious individuals I over time t and $\frac{dR(t)}{dt}$ represents the rate of change of the number of recovered individuals R over time t. In summary, the three equations of the SIR model describe how the numbers of susceptible, infectious, and recovered individuals change over time based on the transmission and recovery rates (Cooper et al., 2020).

The SIR model has several modifications depending on the assumptions of the population as well as the epidemic the model is based on. For example, in the case where there is an exposure period and the individual cannot infect others, then the transmission or spread of that epidemic can be modeled as a susceptible infected exposed removed (SEIR) model. This is basically an extension of the SIR model but introduces the exposed compartment. The SEIR model is given by the following system of differential equations:

$$\begin{aligned} \frac{dS(t)}{dt} &= B - \beta S(t)I(t) - \mu S(t), \\ \frac{dE(t)}{dt} &= \beta S(t)I(t) - (v+\mu)E(t) \\ \frac{dI(t)}{dt} &= vE(t) - (\gamma + \mu + \delta)I(t), \\ \frac{dR(t)}{dt} &= \beta S(t)I(t) - \gamma I(t), \end{aligned} \tag{2.2}$$

where β is the transmission rate, v is the progression rate which is the rate that individuals will move from exposed class to infected class, γ is the removal rate, μ is the natural death rate and δ is disease induced death rate (Shah and Gupta, 2013). In Equation 2.2, $\frac{dE(t)}{dt}$ describes the rate of change of the number of exposed individuals E over time t. The other equations in the system of equations defined by Equation 2.2 follow the same descriptions given under the SIR model. In summary, the four equations of the SEIR model provide a more detailed description of the disease transmission dynamics, accounting for an exposed compartment where individuals are in the latent period before becoming infectious (Shah and Gupta, 2013).

For instance, the study by Zisad et al. (2021) had the primary objective of creating a

predictive model for confirmed COVID-19 cases in Bangladesh. An integrated model was introduced to predict the confirmed cases in Bangladesh. The proposed integrated model combined the SEIR epidemiological model and neural networks. The integrated model was very effective in making the predictions as the accuracy of predictions ranged between 90% and 99% and the integrated model yielded better per performance than the SEIR model alone.

2.1.2 Stochastic Epidemic Models

Stochastic epidemic models are constructed based on random variables to assess probabilistic distributions describing the transmission of infectious diseases through individuals. The commonly used model SIR can be extended to give rise to a stochastic SIR model which will contain randomness. A general stochastic SIR model is defined by the following system of differential equations:

$$\begin{aligned} dS(t) &= (-\beta S(t)I(t) - \mu S(t) + \mu)dt - \sigma S(t)I(t)dW(t), \\ dI(t) &= (\beta S(t)I(t) - (\lambda + \mu)I(t))dt + \sigma S(t)I(t)dW(t), \\ dR(t) &= (\lambda I(t) - \mu R(t))dt, \end{aligned} \tag{2.3}$$

where σ is a nonnegative constant, λ represents the recovery rate of infected people and W is a real Wiener process. The other variables remain the same as defined under the section of deterministic epidemic models (Tornatore et al., 2005).

The study by (Ndiaye et al., 2020) is an example of the stochastic epidemic models being applied to predict the number of COVID-19 infections. The study explores using SIR models both deterministic and stochastic with numerical approximations as well as machine learning techniques to predict the number of COVID-19 cases both for the next 7 days and 3 weeks ahead. The stochastic model is considered mainly due to the lack of data as well as the random factors associated with the transmission of the disease, like the probability of touching infected objects. The key results from the study was that for some countries, like China, the pandemic was expected to end soon within few weeks after the study was conducted.

2.2 Machine Learning

This section presents literature on the machine learning techniques to be used in this study. The scope of this study will only be limited to supervised learning. The long short-term memory (LSTM) will be used to predict COVID-19 infections in South Africa.

Within the fields of computer science and artificial intelligence (AI), machine learning focuses on using data and algorithms to simulate human learning processes and gradually

improve their accuracy. Machine learning is an important component of data science. By employing statistical techniques, algorithms are trained to generate forecasts as well as classifications to make key insights from given datasets (Kubat and Kubat, 2017). According to Khanzode and Sarode (2020), machine learning is most popular due to its ability to produce models quickly, analyse big and more complex data and produce more accurate results.

There are three main machine learning methods namely: supervised learning, unsupervised learning, as well as reinforcement learning.

2.2.1 Supervised Learning

Supervised learning is based on training or learning a mapping function from input data to output labels based on a labeled dataset. During training, the algorithm is presented with labeled datasets, which instruct the algorithm on what output is related to each specific input value. When provided with new data that it hasn't seen before, the model is trained until it detects underlying patterns and relationships between the labels of the input and output data. This enables the model to produce accurate labeling results (Kubat and Kubat, 2017). This type of learning is good for classification and regression problems, such as determining if a patient is infected or not infected using X-ray images (diagnosis) or predicting the number of infections for a particular disease for a given future date. Following training, the model is provided with test data, which is labeled data that the algorithm has not yet been made aware of in order to assess how well the algorithm will perform with unlabeled data (Kubat and Kubat, 2017).

Supervised learning can be categorised into two methods namely; classification and regression. Under the two main methods for supervised learning, there are several popular machine learning algorithms that can be employed for various tasks and these algorithms include linear regression, logistic regression, decision trees, random forests, support vector machines (SVM) and naive Bayes, amongst others.

1. **Classification**

   This is a machine learning method that uses training data to determine the category of new observations. When classifying new observations, an algorithm first learns from the provided dataset or observations and groups them into various classes or groups, like yes or no, 0 or 1, spam or not spam, etc. The classes are referred to as labels, categories or targets.

   - **Decision Tree**
     Decision trees are structures that resemble flowcharts, with each internal node denoting a test on an attribute, each branch denoting an outcome of the test, and each leaf node representing a class label. Each tree is constructed using

a random subset of the features and the final prediction is made by aggregating the predictions of all trees, reducing overfitting and improving accuracy. Multiple trees can be combined to form random forests which are used to make predictions (Izza et al., 2020).

An example of an application of the decision tree algorithm in an incidental health study would be when a hospital wants to predict the severity of COVID-19 infection in patients. The goal would be to develop a model that classifies patients into different groups corresponding to the different COVID-19 case severities. The case severity can be classified into mild, moderate, and critical and the features that can be considered may be age, gender, symptoms, exposure history, any pre-existing health conditions as well as any other relevant health metrics. The trained model can then be used to predict the severity of COVID-19 cases for new patients based on their profiles. This can then assist the decision-makers in the hospital to better allocate resources and prioritise high-risk patients.

The study by Gupta et al. (2021) is an example of where the decision trees model alongside other models was applied in a COVID-19 dataset to predict or detect cases in India. There were three target classes namely confirmed cases, death cases and cured cases which the proposed models were trying to classify into. Random forests, linear model, support vector machine, decision tree as well as neural network were used to make classifications and the random forest yielded better results than the models in detecting COVID-19 cases in India.

- **Support Vector Machine**
  Support vector machines are effective for both classification and regression algorithms. They find a hyperplane that best separates different classes while maximising the margin between them. They are particularly useful when dealing with a high dimensional feature space. The support vector machine learning problem is a classical quadratic optimisation problem with constraints represented as:

$$\begin{aligned} \min \quad & \frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w} \\ \text{subject to} \quad & y_i\left(\mathbf{w}^{\mathsf{T}}x_i + b\right) \geq 1, \quad i = 1, 2, \ldots, l \end{aligned} \tag{2.4}$$

  where $\mathbf{w}$ is the weight vector subject to training, y_i is the target value, b is the bias and x_i are the input features (Kecman, 2005).

The study by Parbat and Chakraborty (2020) serves as an example where support vector regression was applied in a COVID-19 cases dataset from India. The support vector regression model predicted the total number of deaths, recovered cases, cumulative number of confirmed cases and number of daily.

The results suggested that the model was very effective in predicting cumulative cases and did not yield as good results in predicting the number of daily cases due to noise in the dataset for that feature as it had many spikes. Overall, the model had 97% accuracy in predicting deaths, recovered, cumulative number of confirmed cases and had 87% accuracy in predicting daily new cases.

- **Naive Bayes**
  The Bayes theorem and the assumption of feature independence form the foundation of naive Bayes classifiers. The naive Bayes model is based on constructing a Bayesian model that assigns a posterior class probability to an instance (Bayes, 1968). Let y denote the class of an observation $\mathbf{x}_i$, where $y \in \{y_1, y_2, \dots y_m\}$ and $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})$. To predict the class of the observation $\mathbf{x}_i$ by using the Bayes rule, the highest posterior probability of:

$$P(y_j|\mathbf{x}_i) = \frac{P(y_j)P(\mathbf{x}_i|y_j)}{P(\mathbf{x}_i)} \tag{2.5}$$

  should be obtained. Since the naive Bayes classifiers are based on the assumption of independence, this implies that features $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p$ are conditionally independent of each other given a certain class. Thus Equation 2.5 can be rewritten as:

$$P(y_j|\mathbf{x}) = \frac{\prod_{k=1}^{p} P(x_k|y_j)P(y_j)}{P(\mathbf{x})}. \tag{2.6}$$

  A simple naive Bayes classifier can be represented as:

$$\hat{y} = \arg\max_{y_j} \prod_{k=1}^{p} P(x_k|y_j)P(y_j), \tag{2.7}$$

  where $\hat{y}$ is the maximum a posteriori class (Berrar, 2018).

  The study by Romadhon and Kurniawan (2021) serves as an example where naive Bayes amongst other models (k-nearest neighbour and logistic regression) was applied in a COVID-19 cases dataset from Indonesia. The main aim of the study was to predict the patient's recovery/cure rate, with the age and gender being the main variables considered. Based on the findings of the study, it can be noted that k-nearest neighbour, logistic regression as well as naive Bayes can be used to predict the cure rate for COVID-19 patients in Indonesia with the K-nearest being the most effective in predicting as it had the highest accuracy.

2. **Regression**

Regression is a type of predictive modeling technique that looks into the relationship between one or more independent variables (predictor) and a dependent variable (target) (Sykes, 1993).

- **Linear Regression**
  Linear regression is a basic algorithm used for predicting a continuous output from input features. The algorithm exhibits a linear relationship assumption between the input features and the target variables. The general regression model can be represented as:

$$\begin{aligned} y &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \mathcal{E} \\ &= \beta_0 + \sum_{i=1}^{k} \beta_i X_i + \mathcal{E} \end{aligned} \tag{2.8}$$

  where y is the dependent variable, $X_1, X_2, \ldots, X_k$ are k independent variables, $\beta_0, \beta_1, \beta_2, \ldots, \beta_k$ are the regression coefficients representing the model parameters for a specific population, and $\mathcal{E}$ is the error term (Uyanık and Güler, 2013).

  An example of an application of the linear regression model in an incidental health study would be when a hospital wants to predict a patient's blood pressure based on certain patient characteristics. The goal may be to understand how factors like gender, weight, cholesterol levels and age contribute to blood pressure. Then, the model can be used to predict blood pressure for new which may assist the hospital to intervene early and provide preventative measures if a patient is identified to be at risk of having high blood pressure.

  The study by Rath et al. (2020) serves as an example where linear regression was applied in a COVID-19 dataset to predict the next coming days active cases in India and Odisha. The study explored using a simple linear as well multiple linear regression models to perform the predictions. The findings of the study suggests that the proposed regression models were found to be very effective in forecasting the next number daily active cases.

- **Logistic Regression**
  Logistic regression is an algorithm mainly used for classification tasks. It is another type of regression modeling that is used to explore the relationship between a dependent variable and a set of independent variables. A statistical model is built using logistic regression to describe the relationship between a set of independent variables and a binary outcome. It uses a logistic function to predict the likelihood that an instance will belong to a particular class. The logistic regression model can be represented as:

$$\begin{aligned} P(Y = \text{outcome of interest} | X_1, X_2, \ldots, X_k) &= \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k)}} \\ &= \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^{k} \beta_i X_i)}} \\ &= \frac{e^{\beta_0 + \sum_{i=1}^{k} \beta_i X_i}}{1 + e^{\beta_0 + \sum_{i=1}^{k} \beta_i X_i}} \end{aligned} \tag{2.9}$$

where Y denotes the binary variable, P is the probability of an event occurring, $X_1, X_2, \ldots, X_k$ are k independent variables, $\beta_0, \beta_1, \beta_2, \ldots, \beta_k$ are parameters corresponding to the independent variables (Palei and Das, 2009).

An example of an application of the logistic regression model in an incidental health study would be when a hospital wants to predict or assess the risk of COVID-19 infection in individuals having symptoms of COVID-19 or seeking testing if they are infected with the virus. The goal would be to develop a model that classifies patients into two groups, those likely to test positive for COVID-19 and those likely to test negative. The features that can be considered would be age, gender, symptoms, exposure history, a binary variable to indicate COVID-19 test results (positive and negative), and any pre-existing health conditions. The trained model can then be used to predict the probability of testing positive for COVID-19 for new patients and then a threshold would need to be set that will classify the patients into positive or negative categories. This study would potentially assist the healthcare facilities with resource allocation efficiently by identifying high risk individuals who may need more immediate attention or isolation.

The study by Wang et al. (2020) serves as an example where logistic regression was applied in a COVID-19 dataset to predict the trend of the epidermic. The study uses a two step approach to predict the trend. Firstly, the COVID-19 epidemiological data s integrated into the logistic model to fit the cap of the epidermic trend. Then, the cap is fed into the FbProphet model to derive the epidemic curve as well as to predict the trend of the epidemic. The results from the study suggests that the proposed two step approach has a valuable advantage in terms of forecasting the epidemic trend and in using the hybrid model can significantly improve estimates of the number of infections.

2.2.2 Unsupervised Learning

Unsupervised learning is a machine learning approach that does not require the user to supervise the model whilst it is training on unlabeled data. From the provided unlabeled data, the model itself extracts insights and hidden patterns. Regression and classification problems cannot be directly solved using unsupervised learning. Unlike supervised learning, where input data is paired with corresponding output data, unsupervised learning deals solely with input data, lacking matching output data (Hinton and Sejnowski, 1999).

Unsupervised learning can be categorised into two methods namely clustering and association. Under the two methods for unsupervised learning, there are several popular machine learning algorithms that can be employed for various tasks. These algorithms include k-means clustering, hierarchical clustering, principal component analysis (PCA),

and association rule learning just to mention a few.

1. **Clustering**

   This is a machine learning technique where the population or data points are divided into several groups so that the data points within each group are more similar to each other than they are to the data points outside of them. Essentially, this is a collection of objects according to their similarities and dissimilarities amongst each other (James et al., 2013). Some of the most commonly used clustering algorithms are provided below.

   - **K-means**
     K-means is a clustering algorithm that partitions data into K non-overlapping clusters. It minimises the intra-cluster variance, aiming to place each observation into the closest cluster. It is efficient and easy to implement but requires the number of clusters to be predefined. To define K-means more formally (James et al., 2013), let $C_1, \ldots, C_k$ denote sets corresponding to each cluster. The sets should satisfy two properties:

     (a) $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$. This implies that the sets are unique and each observation can only belong to one cluster.

     (b) $C_1 \cup C_2 \cup \cdots \cup C_k = \{1, 2, \ldots, n\}$, suggesting that each observation must be a member of at least one of the k clusters.

     The K-means clustering algorithm learning problem is a classical quadratic optimisation problem that can be defined as:

     $$\min_{C_1,\ldots,C_K} \quad \sum_{k=1}^{K} W(C_k) \tag{2.10}$$

     where $W(C_k)$ is the within-cluster variation which is a measure of the amount by which observations in a cluster differ and is defined by:

     $$W(C_k) = \frac{1}{|C_p|} \sum_{i,i' \in C_p} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2, \tag{2.11}$$

     with $|C_k|$ denoting the number of observations in the kth cluster. Thus, Equation 2.10 becomes:

     $$\min_{C_1,\ldots,C_K} \quad \sum_{k=1}^{K} \frac{1}{|C_p|} \sum_{i,i' \in C_p} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2. \tag{2.12}$$

     An example of an application of the K-means model in an incidental health study would be when a hospital wants to identify distinct patterns or clusters of COVID-19 patients based on certain healthcare indicators. The goal would be

to gain insights into different patient profiles containing features such as age, temperature, respiratory rate, blood test results, a binary variable indicating COVID-19 status, and any other relevant health metrics. The different clusters would enable us to analyse and understand the different patient profiles. This study would then enable the hospital to tailor resource allocation and care strategies based on the identified clusters. For example, more resources may be allocated to clusters with higher severity.

The study by Kurniawan et al. (2020) serves as an example where K-means was applied in a COVID-19 dataset to search hidden and unknown clusters of countries exhibiting a rapid COVID-19 infections and determine the relationships among a set of attributes. The major findings of this study was that the there was a strong positive correlation among the total deaths and patients categorised as critical.

- **Hierarchical Clustering**
  Hierarchical clustering builds a hierarchy of clusters by recursively merging or splitting them. It does not require specifying the number of clusters in advance as in the case of the K-means algorithm (Wang, 2003).

  The study by Rios et al. (2021) serves as an example where hierarchical clustering was used to predict COVID-19 next waves of infections. The study explores the feasibility of understanding and predicting the virus's progression in one country by considering historical reports from others. Employing an artificial intelligence approach based on unsupervised machine learning, the researchers organise time series data to facilitate cross-country comparisons and introduce a transition index. This index proves valuable in identifying potential trends and informing public policies. Despite challenges related to data collection variations among countries, the study's visualisation metaphors offer insights into historical infection paths and potential future waves, making it a significant contribution to both COVID-19 research and unsupervised machine learning methodologies. The study acknowledges limitations related to data collection challenges but emphasises their commonality in data-driven projects.

- **Gaussian Mixture Models**
  Gaussian mixture models (GMM) represent the data as a weighted sum of Gaussian distributions. It assumes that the data is generated from a mixture of these distributions. GMM can be used for clustering, density estimation, and generating synthetic data. It provides flexible modeling capabilities but may require specifying the number of components. To define the GMM mathematically, we start off by defining the Gaussian distribution (Normal distribution) which is given by:

$$\mathcal{N}(X|\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}\sqrt{|\Sigma|}} \exp\left\{-\frac{(X-\mu)^T\Sigma^{-1}(X-\mu)}{2}\right\} \tag{2.13}$$

where μ is a D-dimensional mean vector, Σ is a $D \times D$ covariance matrix, and $|\Sigma|$ is the determinant of the covariance matrix. As previously mentioned the GMM models each cluster as a Gaussian distribution, hence a GMM is represented as a linear combination of the basic Gaussian probability distribution is expressed as:

$$p(X) = \sum_{k=1}^{K} \pi_k \mathcal{N}(X|\mu_k,\Sigma_k) \tag{2.14}$$

where K is the number of components in the mixture model and π_k is the mixing coefficient, it gives an estimate of the density of each Gaussian component (Patel and Kushwaha, 2020).

An example of an application of the GMM model in an incidental health study would be when a hospital wants to understand the underlying subpopulations within COVID-19 patients based on a combination of certain healthcare indicators. The goal would be to apply the GMMs to identify clusters that might represent different severity levels among COVID-19 cases. The features that would be considered for such a study may be age, temperature, respiratory rate, blood test results, a binary variable indicating COVID-19 status, and any other relevant health metrics. The GMM would be applied to the selected features. The algorithm assumes that the data is generated from a mixture of several Gaussian distributions, each representing a different subpopulation. Then a cluster analysis would be performed on the resulting clusters to understand the different patient subgroups. Such a study can help the hospital in providing more personalised and targeted medical interventions based on the characteristics of each identified subpopulation.

The study by Singhal et al. (2020) serves as an example where GMM was used to capture the trend of a number of COVID-19 cases and predict the cases. The introduces two methods for modeling COVID-19 infections: a mathematical model estimating critical factors influencing the virus spread and forecasting active cases for the next 30 days, analysing infection containment measures in India, Italy, and the USA. The second method employs a data-driven approach using a Gaussian mixture model to separate trend and variability in daily infection cases, predicting peak values and corresponding dates. The study estimates total cases and end-dates of the pandemic with 95% confidence intervals, comparing results with a prior study.

2. **Association**

   This is another type of unsupervised learning technique that, in order to make mappings more profitable, checks for dependencies between data items. It looks for interesting relationships or connections between the variables in a dataset.

   - **Principal Component Analysis**
     Principal component analysis (PCA) is a dimensionality reduction technique that finds the most important features or components in the data. It transforms the data into a new coordinate system to maximise the explained variance. If a column has less variance, it has less information. PCA helps in visualising data, reducing dimensionality, and removing redundant information (Jolliffe, 1990).

     An example of an application of PCA in an incidental health study would be when we want to reduce the dimensionality of a dataset containing various health indicators for COVID-19 patients. The goal would be to identify the most influential factors and potential patterns that contribute to the severity of COVID-19 cases.

2.2.3 Reinforcement Learning

Reinforcement learning is another type of machine learning technique. In essence, it is the science of making decisions. It involves discovering the best way to behave in a given situation in order to maximise rewards. Through interactions with the environment and observations of its responses, this ideal behavior is acquired (Sutton and Barto, 2018).

An example of reinforcement learning in healthcare is the use of reinforcement algorithms to optimise treatment plans for chronic diseases. In this scenario, a reinforcement learning agent can be trained using patient data and medical guidelines to make decisions about the dosage and frequency of medication for a patient with a chronic disease. The agent learns from past patient outcomes and receives feedback in the form of rewards or penalties based on the clinical outcomes of its decisions.

Through repeated interactions with the patient's electronic health records, reinforcement learning can learn to penalise treatment plans that are tailored to individual patient characteristics and optimise the patient's health outcome over time. This can result in more effective and efficient treatment plans, reduced costs, and improved patient well-being.

The study by Khalilpourazari and Hashemi Doulabi (2021) serves as an example of how reinforcement can be used to predict the COVID-19 outbreak. In this study, the focus is on predicting the COVID-19 pandemic to aid policymakers in optimising healthcare system capacity and resource allocation, ultimately minimising fatality rates. The researchers propose a new hybrid reinforcement learning-based algorithm for complex optimisation

problems, demonstrating its superiority over state-of-the-art methods. Applying the algorithm to real-world data from Quebec, Canada, the study accurately forecasts the COVID-19 outbreak trends, considering various scenarios and identifying crucial factors influencing pandemic growth. The results emphasise the significant impact of the transmission rate on future trends, highlighting the importance of implementing social distancing measures and increasing asymptomatic case detection to curb the virus spread and halt the pandemic.

2.2.4 Deep Learning

In this section, the concepts of neural networks as well as recurrent neural networks are introduced in greater detail which will enable us to better understand the LSTM model. For the purpose of this study, two categories of deep learning namely neural networks and recurrent neural networks will be considered. It is essential to have a basic understanding of neural networks as well as recurrent neural networks as the LSTM is built on the foundations of the two.

Neural Network

Neural networks are artificial systems that are a class of machine learning models which are modeled under the inspiration of biological neural networks. They are a subset of artificial intelligence that was created to imitate how biological nervous systems and the human brain interpret and utilise data.

It is estimated that there are 100 billion neurons in the human brain. Figure 2.1 illustrates an example of a brain neuron. Neurons communicate via electrical signals that are short-lived impulses in the voltage of the membrane (cell wall). Synapses on dendrites act as mediators between interneuron connections (Gurney, 2018).

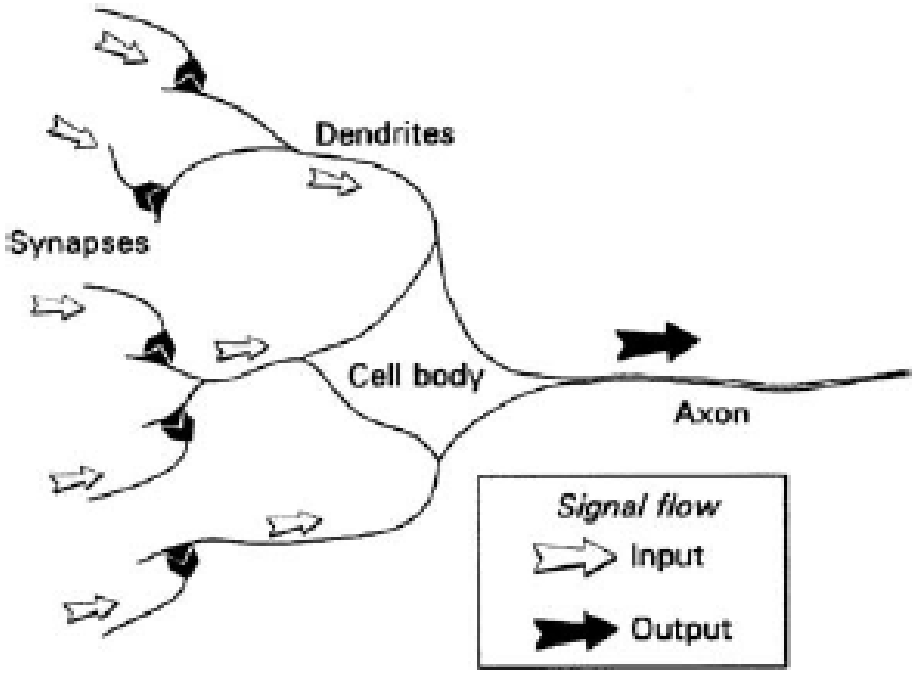

Figure 2.1: Structure of a brain neuron (Gurney, 2018)

Every neuron normally has thousands of connections with other neurons, which means that a constant stream of incoming signals are being received by it until they reach the cell body. In some manner, the signals are combined, and if the total signal is greater than a certain threshold, the neuron may produce a voltage impulse in response. This impulse is then sent through the axon to other neurons. The axon is a branching fibre (Gurney, 2018).

Certain incoming signals have an inhibitory effect and prevent the production of impulses, while other incoming signals are excitatory and help to promote the generation of impulses. Neural networks use this architect and processing style, which emphasises the significance of interneuron connections throughout (Gurney, 2018).

Artificial neurons serve as the fundamental building blocks of neural or artificial neural networks, which are composed of neurons, connections, weights, biases, propagation functions, and learning rules. An artificial neuron's fundamental architecture consists of an input layer that processes data from inputs and an output layer that generates outputs or predictions in response to the inputs. The artificial neuron's hidden layers in between process the input data in a non-linear way. Figure 2.2 is an illustration of the basic structure of an artificial neuron.

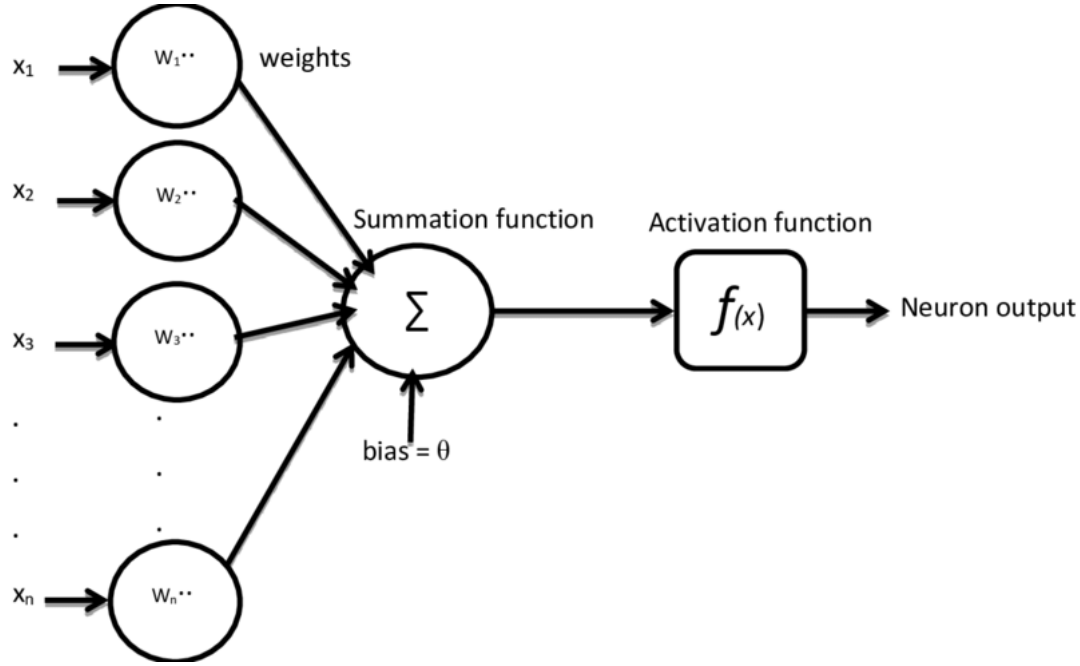

Figure 2.2: Basic structure of an artificial neuron (Yacim and Boshoff, 2018)

An artificial neuron has a finite number n of inputs. Each input x_i has a corresponding weight w_i. The inputs are the set of values that are used to predict the output value whereas weights are real values associated with each feature that specifies the importance of the value in the predictions of the output. The summation function serves to bind the inputs and weights together and get the resulting sum. From Figure 2.2, the summation function Σ of this particular artificial neuron is given by:

$$\Sigma = \sum_{i=1}^{n} x_i w_i + b, \tag{2.15}$$

where b is the bias term. The result obtained from Equation 2.15 are passed to the activation function, which gives the output $f(\Sigma)$ (Yacim and Boshoff, 2018). More discussion on the activation function is provided in the following subsection.

Activation Functions

Activation functions are mathematical equations that determine the output of a neural network model. They serve to introduce non-linearity characteristics to networks allowing the neural networks to learn powerful operations. They are present in every neuron in the network and help determine when to turn off neurons, based on the input used. They also help to normalise the output of any input in the range between -1 and 1 or 0 to 1.

Some of the most commonly used activation functions are discussed below.

- **ReLu**
  The rectified linear unit (ReLu) function is amongst the commonly used activation

functions in modern neural networks. This function is a piecewise linear function that returns 0 as an output if its input is negative, or otherwise directly outputs the input (Ramachandran et al., 2017).

Mathematically, the ReLu function can be defined as:

$$f(x) = \max(0, x). \tag{2.16}$$

The visualisation presented in Figure 2.3 is a graphical representation of the ReLu function:

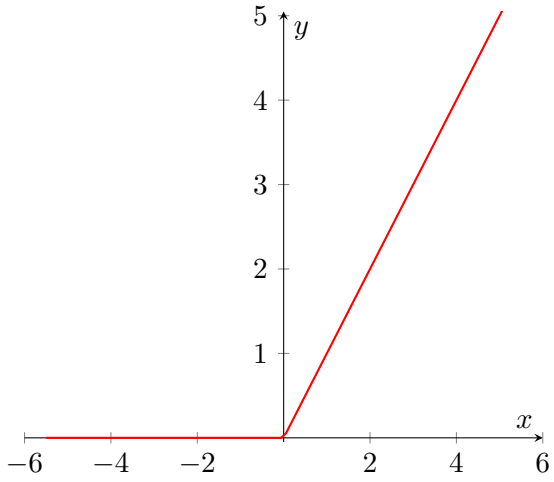

Figure 2.3: ReLu function

- **Sigmoid**

  Another common activation function is the sigmoid function. This is a non-linear activation function that takes input and converts it to a value between 0 and 1. This function has the useful property that its gradient is defined everywhere and it is mathematically easier to work with. However, the exponential functions make it computationally expensive to compute the sigmoid function in practice and at times leads to the ReLu function as the most preferred choice due to its simplicity (Ramachandran et al., 2021).

  Mathematically, the sigmoid function can be defined as:

$$\begin{aligned} S(x) &= \frac{1}{1+e^{-x}}, \\ &= \frac{e^x}{e^x+1}, \qquad \forall x \in \mathbb{R}. \end{aligned} \tag{2.17}$$

  The visualisation presented in Figure 2.4 is a graphical representation of the sigmoid function:

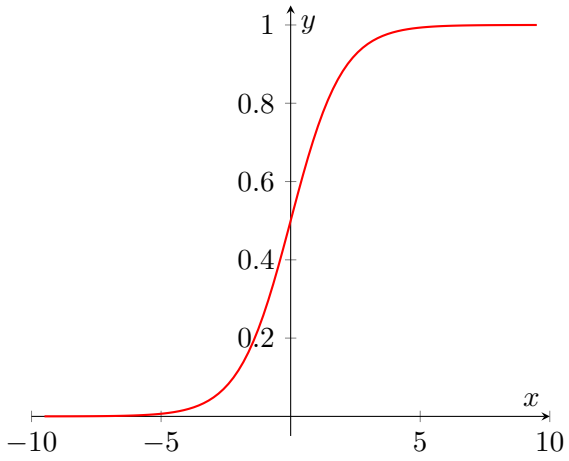

Figure 2.4: Sigmoid function

- **Hyperbolic Tangent**
  This is a continuous non-linear activation function that produces outputs between the values -1 and 1. This is the most commonly used activation for recurrent neural networks and the LSTM model (Farzad et al., 2019).

  Mathematically, this function can be defined as:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \qquad \forall x \in \mathbb{R}. \tag{2.18}$$

  The visualisation presented in Figure 2.5 is a graphical representation of the hyperbolic tangent function:

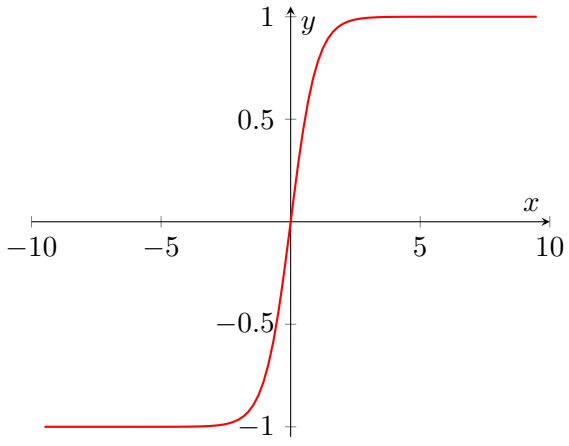

Figure 2.5: Hyperbolic tangent function

Hadamard Product

The Hadamard product or element wise multiplication ($\odot$) of two matrices or vectors is similar to matrix addition, corresponding elements in the same rows and columns on the matrices are multiplied together to form a new matrix or vector (Million, 2007). The order of matrices to be multiplied should be the same and the resulting matrix will also be of the same order and matrices being multiplied need not necessarily be square.

Suppose there are two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ defined as

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{bmatrix},$$

their Hadamard product will be

$$\boldsymbol{A} \odot \boldsymbol{B} = \begin{bmatrix} a_{11} \times b_{11} & a_{12} \times b_{12} & \cdots & a_{1n} \times b_{1n} \\ a_{21} \times b_{21} & a_{22} \times b_{22} & \cdots & a_{2n} \times b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} \times b_{n1} & a_{n2} \times b_{n2} & \cdots & a_{nn} \times b_{nn} \end{bmatrix}.$$

For example, consider the following matrices,

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix},$$

their Hadamard product will be given by

$$\boldsymbol{A} \odot \boldsymbol{B} = \begin{bmatrix} 1 \times 5 & 2 \times 6 \\ 3 \times 7 & 4 \times 8 \end{bmatrix} = \begin{bmatrix} 5 & 12 \\ 21 & 32 \end{bmatrix}.$$

Recurrent Neural Network

Recurrent neural networks (RNNs) are a type of neural network where the output from the preceding step is used as an input to the current step. All of the inputs and outputs in conventional neural networks are independent of one another and these kinds of networks do not remember things that they learn (Medsker and Jain, 2001). After every iteration, while training the network, it starts afresh without remembering anything it has learned during the previous iteration. This common property for traditional NN does not seem to work for data that is sequential like time series data. RNNs are useful when making predictions on data that is sequential since the state vector and the cell state allow us to carry information across a larger time window than simple NN (Medsker and Jain, 2001).

Chapter 3: Research Methodology

This chapter describes the nature and scope of the methodology applied in the study. It will present the identified predictive modeling techniques for COVID-19 infections as well as the proposed model performance measures or metrics used in the study.

The diagram in Figure 3.1 presents the research design applied in this study as well as the flow of how the study will take place. The following subsections will provide a detailed description of how each step will be executed as well as reasons for each step.

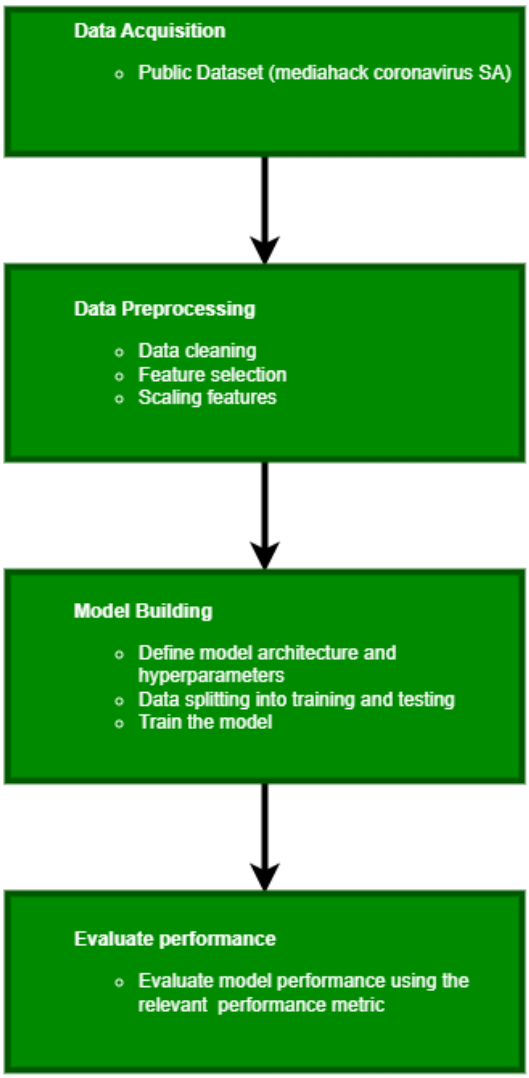

Figure 3.1: A flowchart of the research design.

3.1 Data Acquisition

For this particular study, the dataset used to model or predict COVID-19 infections in South Africa is a publicly available open dataset obtained from the Mediahack Coronavirus Dashboard web page (https://mediahack.co.za/datastories/coronavirus/data/). A more detailed descriptive analysis of this dataset will be provided in Chapter 4.

3.2 Data Preprocessing

Data preprocessing is a data mining step that is performed on raw data and transforms it into a format that can easily be understood by computers and machine learning algorithms. This is a very important step in the data science lifecycle. If this step is missed or not executed properly, then the results of the model will not be reliable. The dataset used in this study is a time series dataset, which will be utilized to predict COVID-19 infections at a later stage in South Africa.

The first step will be to understand the data to be used and that will be achieved by performing some data exploration. This step entails checking if there were any missing values in the dataset, and the replace any missing values with zeros. The values that were identified to be null were when there were not yet any observations for that particular variable, hence, they are equivalent to zero.

Another important step that will have to be performed is feature engineering. The process of feature engineering entails selecting, manipulating, or transforming features in a dataset to improve the model performance and interpretability of a machine learning model. For this particular work, we will perform feature selection which is a subset of feature engineering. Feature selection is the process of choosing a subset of significant or relevant features from a larger set in a dataset. This is a very critical step in the machine learning pipeline, particularly when dealing with a dataset that with many potentially irrelevant features that will necessarily not contribute to model performance. Once the relevant features have been obtained, the next step will be looking at the basic statistics (5-number summary) for each selected variable in the dataset which will be useful in identifying any outliers or anomalies in the data. This is to have a better understanding of the type of dataset being used for predictions and then create a certain visualisation for each feature with the purpose of assisting in understanding the data better and in identifying trends and relationships between the selected features if they exist.

The LSTM model functions by learning a function that maps a sequence of historical observations as input to an output observation. As such, the sequence of observations in our dataset is transformed into multiple examples from which the LSTM can learn. For the purpose of learning a one-step prediction, the data was split into various input/output patterns, with three-time steps serving as the input and one-time step serving as the

output.

Then, the data will be split into 80% and 20% for training and testing, respectively and then data will be scaled using the MinMax Scaler. The MinMax Scaler is a data preprocessing technique used to normalise features in a dataset, which standardises the observations into values between 0 and 1 to make training the models easier and use less memory.

3.3 Long Short-Term Memory Network

The challenge of learning long-term dependencies given vanishing gradients has been a significant drawback for RNNs. The LSTM network overcomes this constraint by employing memory cells in the hidden layer, which aid in the retention of long-term dependencies within the data. The LSTM network is an advanced RNN that allows information to persist and is capable of learning long term dependencies.

This section presents the different proposed models to be considered for this study as well as the hyperparameters that will be used to build the various models.

3.3.1 Persistence Model

To measure the performances of the proposed LSTM models, this study starts off by training a baseline model to predict new COVID-19 infections. Using observations from the current time step t, predictions for observations at the next time step $t+1$ are generated. Persistence implies that future values of the time series are calculated on the assumption that conditions remain unchanged between “current” time t and "future" time $t+1$. Persistence modeling is used to ensure that no time is being wasted by training models that are not predictive.

3.3.2 Basic Long Short-Term Memory Model

A typical LSTM network is made up of memory blocks called cells. The LSTM cells are responsible for remembering things and consist of three major mechanisms known as gates. The three different types of gates of an LSTM cell are the forget gate, the input gate as well and the output gate (Chandra et al., 2022). Figure 3.2 is an illustration of an LSTM cell, where X_t is the input vector, C_t and C_{t-1} are the current and previous cell memories, respectively. H_t and H_{t-1} are the current and previous cell outputs, respectively. $\tilde{C}_t$ is the intermediate cell state and f_t, I_t, O_t are the forget, input, and output gates, respectively.

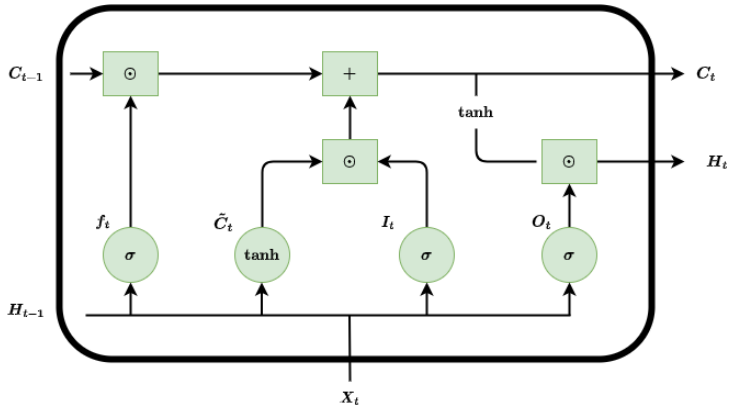

Figure 3.2: Basic LSTM Cell (Chandra et al., 2022)

The primary function of the **forget gate** is to decide which data from the cell state should be retained or discarded. The information that is no longer required or is of less importance for the LSTM to understand things is removed. The equation for the forget gate is given by:

$$f_t = \sigma(X_t U^f + H_{t-1} W^f), \tag{3.1}$$

where X_t is the input at the current timestamp, U^f is the weight associated with the input, H_{t-1} is the hidden state at the previous timestamp and W^f is the weight matrix associated with the hidden state. The sigmoid function presented by σ is applied to the equation which ensures that the value of f_t lies between 0 and 1. This f_t value will be later multiplied with the cell state of the previous timestamp.

The primary function of the **input gate** is to quantify the importance and significance of the new information brought by the input. The equation for the input gate is given by:

$$i_t = \sigma(X_t U^i + H_{t-1} W^i), \tag{3.2}$$

where U^i is the weight matrix of the input and W^i is the weight matrix of input associated with the hidden state.

As new data is coming through, the intermediate cell state should be updated to reflect the new information. The intermediate cell state $\tilde{C}_t$ is updated using the following equation,

$$\tilde{C}_t = \tanh\left(x_t U^C + h_{t-1} W^C\right). \tag{3.3}$$

Not all in the intermediate cell will be remembered by them, only the most important and relevant will be remembered. If there is relevant information, then the current cell state will be updated with the relevant values being added to it. The current cell state C_t will be updated using the following equation:

$$C_t = \sigma\left(f_t \odot C_{t-1} + i_t \odot \tilde{C}_t\right). \tag{3.4}$$

The equation to determine the value at the output gate is the following:

$$O_t = \sigma(x_t U^O + h_{t-1} W^O). \tag{3.5}$$

Finally, to determine the value for the hidden state, the following equation is used:

$$h_t = \tanh(C_t) \odot O_t. \tag{3.6}$$

Figure 3.3 depicts a representation of a basic LSTM network.

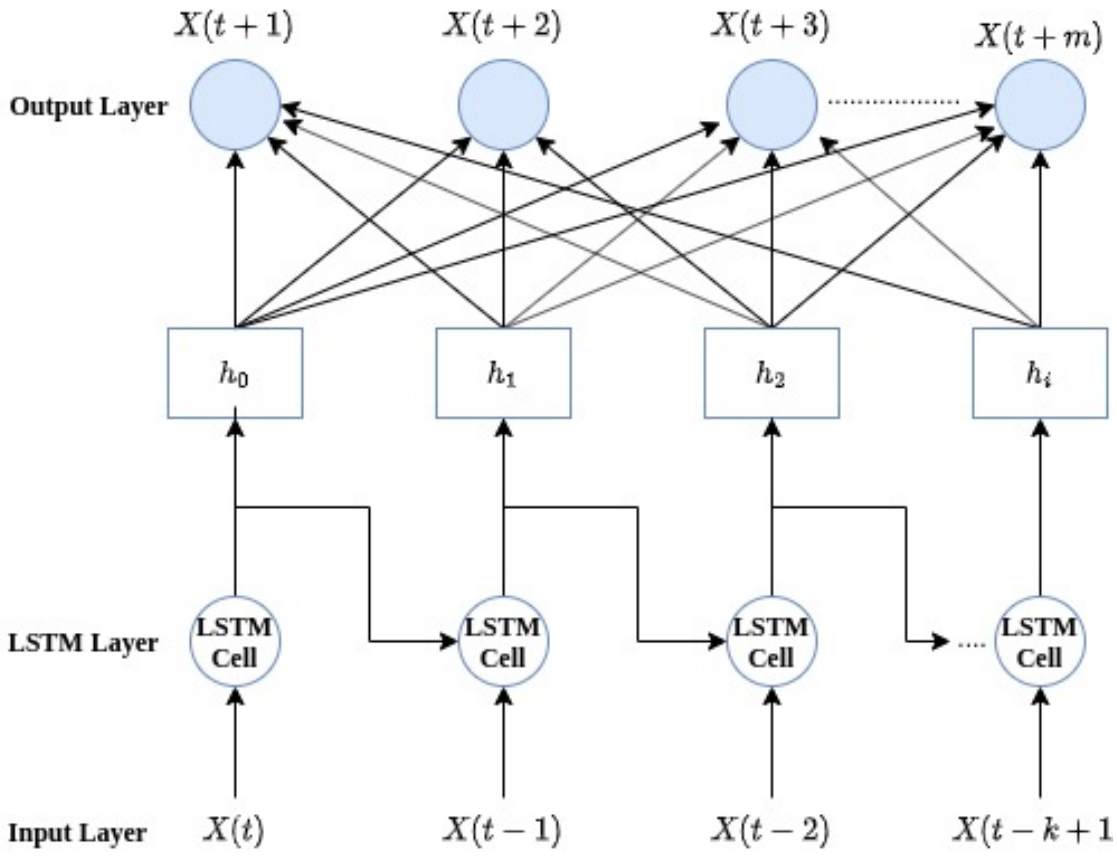

Figure 3.3: Basic LSTM Network (Chandra et al., 2022)

3.3.3 Stacked Long Short-Term Memory Model

The stacked LSTM model is an extension of a basic LSTM model. The architecture of this model consists of multiple layers of LSTM cells. In a stacked LSTM model, the output of each LSTM layer is fed as input to the next layer in the stack. This allows the network to learn more complex representations of the input data by building hierarchical abstractions across multiple layers.

3.3.4 Bidirectional Long Short-Term Memory Model

In a standard LSTM network, the data is processed in a temporal way and the information data that contains future features is neglected. The bidirectional LSTM addresses these shortcomings by having two states, the forward LSTM layer which makes use of past features, and the backward LSTM layer which utilises the future states for a specific time frame. The bidirectional LSTM has the powerful property that it can remember long sequences since the input is considered for both directions. Figure 3.4 illustrates a bidirectional LSTM network.

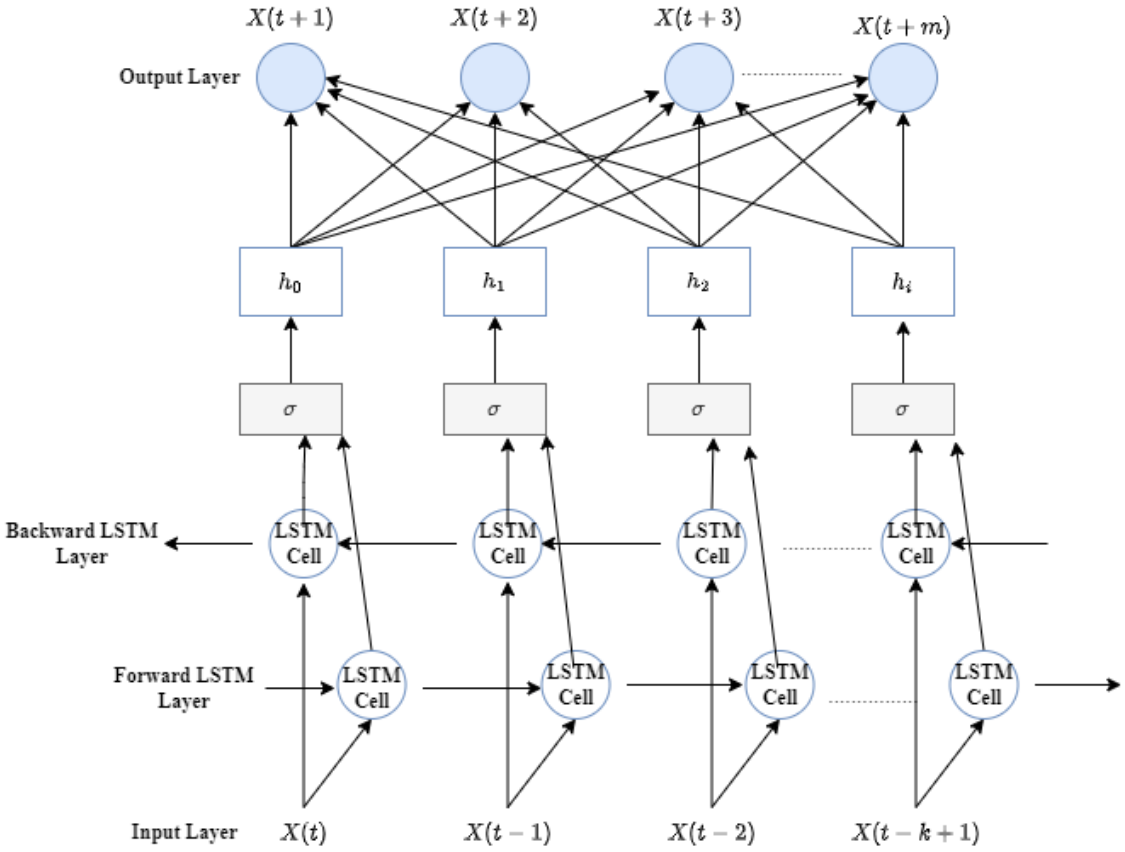

Figure 3.4: Bidirectional LSTM Network (Ihianle et al., 2020)

3.4 Hyperparameters

Hyperparameters are configurations that are set prior to training a machine learning model and are not learned from the data used for training. They play a critical role in determining the behavior and performance of a machine learning algorithm. Unlike model parameters, which are learned during training (such as weights in a neural network), hyperparameters are predefined by the researcher based on what the researcher observed when doing an exploratory analysis of the data that will be used for building the model and they remain constant throughout the training process. Below are some

common hyperparameters in machine learning

1. Learning Rate
   The learning rate controls the step size during the optimisation process, for example, gradient descent. It establishes the rate at which a model learns. While a larger learning rate may result in faster convergence at the risk of overshooting the optimal solution, a smaller learning rate may lead to more stable convergence but may require more training time (Goyal et al., 2017).

2. Number of epochs
   An epoch is a complete pass through the entire training dataset. The number of epochs indicates how many times the training data will be iterated over by the model. Too many epochs can cause overfitting, while too few can cause underfitting (Goyal et al., 2017).

3. Batch Size
   Batch size determines the number of data samples used in each iteration of training. It can affect training speed and memory usage. Smaller batch sizes may lead to noisier updates but can help the model generalise better (Goyal et al., 2017).

4. Number of layers and units
   These hyperparameters are specific to neural networks and deep learning models. The number of layers and the number of neurons (units) in each layer define the model's architecture (Phaisangittisagul, 2016).

5. Activation Functions
   Activation functions are used in neural networks to introduce non-linearity. The choice of activation function can impact how well the model can approximate complex functions. More discussion on activation functions to follow in the upcoming sections of this chapter (Ramachandran et al., 2017).

6. Dropout Rate
   Dropout rate is a regularisation technique employed in neural networks to circumvent overfitting. The dropout rate determines the probability that a neuron is temporarily "dropped out" during training, meaning it doesn't contribute to the forward or backward pass in that iteration (Ba and Frey, 2013).

7. Regularisation strength
   Regularisation strength involves introducing a penalty term into the loss function in order to prevent overfitting. The regularisation strength (e.g., L1 or L2 regularisation) controls the impact of this penalty on the loss (Phaisangittisagul, 2016).

8. Optimizer
   Optimizers like Adam or stochastic gradient descent determine how the model's weights are updated during training. Each optimizer has its own hyperparameters, such as learning rate and momentum (Li et al., 2018).

9. Loss function
   The loss function is used to measure how well the model is performing. Common loss functions include mean squared error for regression tasks and categorical cross-entropy for classification tasks. More discussion on loss functions to follow in the upcoming sections of this chapter (Li et al., 2018).

Table 3.1, Table 3.2, and Table 3.3 contains a summary of the different hyperparameters that will be used to build the basic LSTM model, the stacked LSTM model, and the bidirectional LSTM model, respectively.

Table 3.1: Hyperparameters for the basic LSTM model

Hyperparameter	**Configuration**
Batch size	72
Dropout rate	0.2
Number of neurons	32
Epochs	100
Loss	Mean squared error
Optimizer	Adam

Table 3.2: Hyperparameters for the stacked LSTM model

Hyperparameter	**Configuration**
Batch size	72
Dropout rate	0.2
Number of neurons	32
Epochs	100
Loss	Mean squared error
Optimizer	Adam

Table 3.3: Hyperparameters for the bidirectional LSTM model

Hyperparameter	**Configuration**
Batch size	72
Dropout rate	0.2
Number of neurons	32
Epochs	100
Loss	Mean squared error
Optimizer	Adam

3.5 Performance Evaluation Metrics

A metric is needed to determine how well the model is performing. Loss or a cost function is used as a measurement of determining how well a prediction model performs in predicting the expected outcome and if there is a change or difference between the true and predicted value. The preferred loss function for neural networks applied to regression models is the mean squared error (MSE) (McCaffrey, 2018).

The commonly used loss functions for NN are the following:

MSE
The mean squared error is given by:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{3.7}$$

where N represents the number of testing samples, y_i and $\hat{y}_i$ are the observed and predicted values, respectively.

RMSE
The root mean squared error (RMSE) is given by:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}. \tag{3.8}$$

MAE
The mean absolute error (MAE) is given by:

$$MAE = \frac{1}{N}\left|y_i - \hat{y}_i\right|. \tag{3.9}$$

The evaluation metric that will be used for this study is the RMSE. RMSE is sensitive to the magnitude of errors and penalises large errors more heavily than small errors. This aligns with the objective of this work by emphasising the importance of minimising prediction errors (Li et al., 2019). RMSE is also intuitive and easy to interpret. A lower RMSE indicates better model performance.

Chapter 4: Results and Discussions

This chapter mainly focuses on the results of this study. This entails the selection of features, the modeling aspect of the study, the performance of different models during training and their effectiveness in predicting unseen testing data. Finally, an extensive discussion comparing the results of this study with other historical studies.

4.1 Descriptive Analysis

This section provides a descriptive analysis of the dataset used in this study. This entails the data description where the different features found in the dataset are explained. The minimum date on the dataset is the 5th of March 2020 and the maximum date is the 22nd of August 2021. Table 4.1 provides a description of the features found in the dataset that will be used for modeling.

Table 4.1: COVID-19 dataset features and their definitions

Feature	**Definition**
cumulative_cases	running total of cases over time
recoveries	number of recoveries
deaths_daily	number of deaths per day
cumulative_deaths	running total of deaths over time
active_cases	number of active cases (reported cases which have not resulted in a death or recovery)
week_label	label of a week based on when the week started and when it ended
week_no	week number
cumulative_tests	running total of tests over time
tests_daily	number of tests per day
cases_daily	number of new cases per day
perc_positive	cases_daily divided by tests_daily
test_thousand	cases_daily divided by tests_daily, multiplied by 1000
weekly_tests	number of tests per week
weekly_cases	number of cases per week
tests_per_case	weekly_tests divided by weekly_cases
seven_day_positives	sum of past seven days in the cases_daily column
seven_day_average	seven_day_positives divided by seven_day_tests
case_fatality_rate	cumulative deaths divided by cumulative_cases
fatalities_seven_days	sum of the last seven days in the deaths_daily column

fatalities_seven_days_average	fatalities_seven_days divided by seven_day_positives
public_tests	tests done in public healthcare facilities
private_tests	tests done in private healthcare facilities
new_public_tests	new number of tests done in public healthcare facilities
new_private_tests	tests done in private healthcare facilities
7_day_weekly_average	seven_day_positives divided by 7
vaccinated_total	total number of vaccines administered
vaccinated_daily	number of vaccines administered per day
non_sisonke_vaccination_total	vaccinated_total minus vaccinated_sisonke
vaccinated_sisonke	vaccinations for healthcare workers
j_and_j	number of individuals who have taken the Johnson and Johnson vaccine
pfizer	number of individuals who have taken the Pfizer vaccine
vaccinations_7_day_average	sum of last 7 days from the vaccinated_daily column, divided by 7

4.2 Feature Selection

This step involved using a filter method to identify the highly correlated features to each other so as to identify redundant features that will be discarded and only keep relevant features. The goal of identifying and keeping the relevant features is that those features will contribute to model building and interpretability.

One thing to note about the dataset used for this study was that it was mainly built with the intention of creating dashboards that were to be used to track certain COVID-19 metrics. Some of the features are aggregations of other features, for example, there is a feature named weekly_tests and another feature titled tests_daily. Weeky_tests is already an aggregate tests_daily, hence keeping both features will not provide any meaningful insights since one of the fields is already contained in the other feature. Figure 4.1 contains a heatmap that is used to visualise a correlation of the features that are in the dataset.

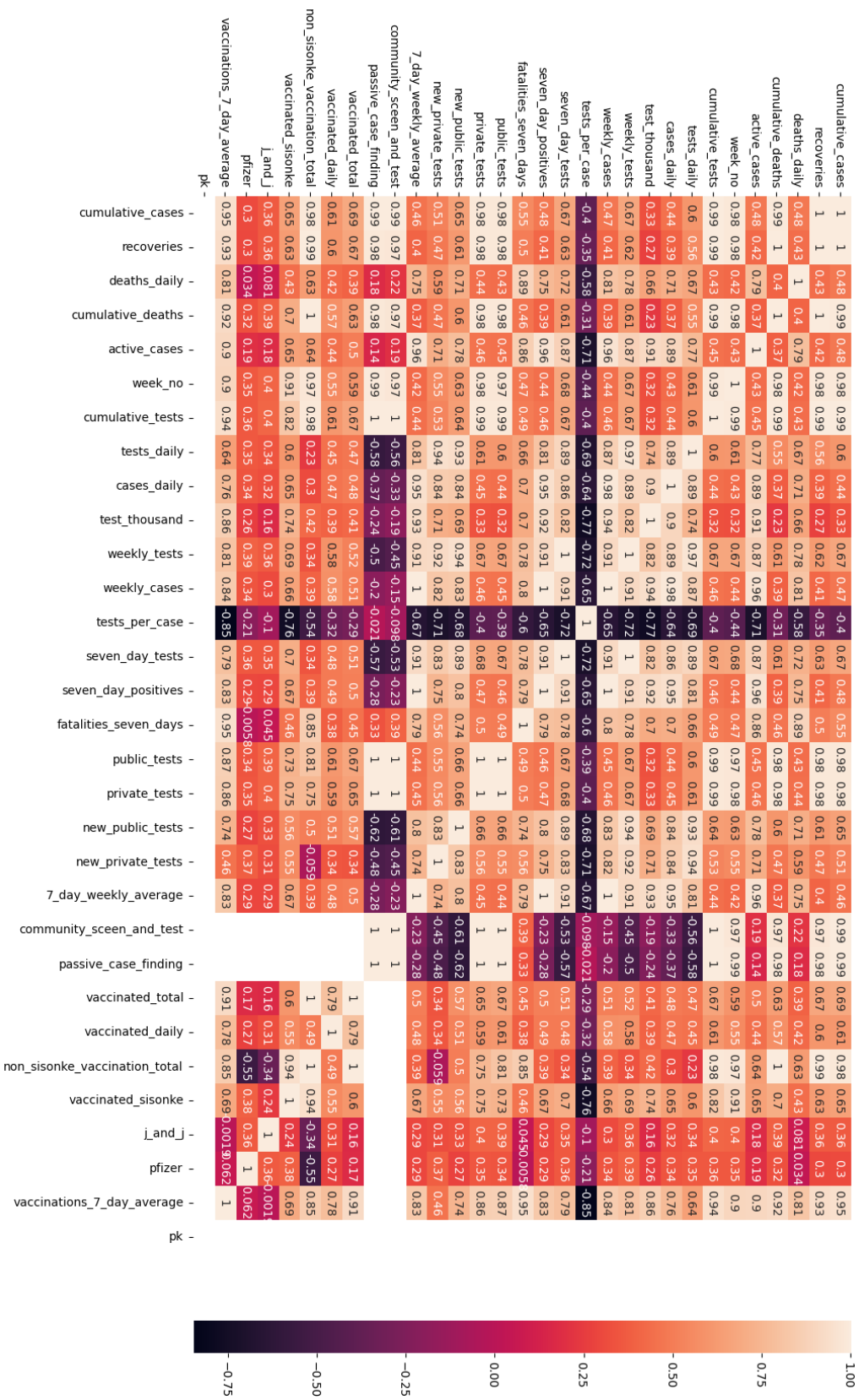

Figure 4.1: Correlation matrix of the features in the COVID-19 dataset

The correlation matrix was carefully examined and the relevant features were identified to be those related to the number of tests, the number of deaths, the number of active cases

as well as a feature related to vaccination. For modeling purposes, the features that will be considered are **cumulative_cases**, **recoveries**, **deaths_daily**, **active_cases**, **tests_daily**, **vaccinated_daily**, and **cases_daily**. These are the unique features in the dataset and all the other features have been obtained by performing some level of aggregation on these basic features and hence exhibit high correlation to either one of the unique features. For the purpose of this work, the features that were considered were the ones that are on a daily level and not those that have been aggregated further. This is due to ensuring that there is a bigger dataset that can be used to train the different deep learning models. For more accurate predictions, a deep learning model requires a larger dataset to be used when training the model. Hence, the variables that are on a daily level are ideal to use in this case.

Figure 4.2, Figure 4.3, Figure 4.4, Figure 4.5, Figure 4.6, Figure 4.7, and Figure 4.8 shows plots containing time series graphs of the selected relevant variables so as to visualise their trend over time.

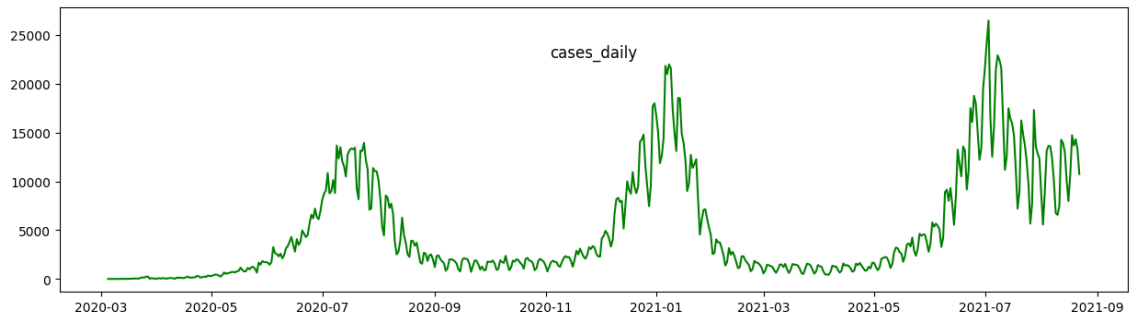

Figure 4.2: Time series plot for cases_daily

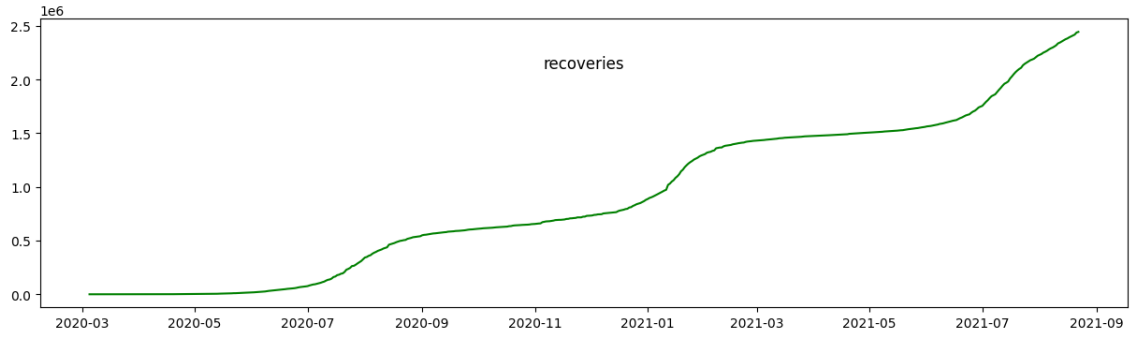

Figure 4.3: Time series plot for recoveries

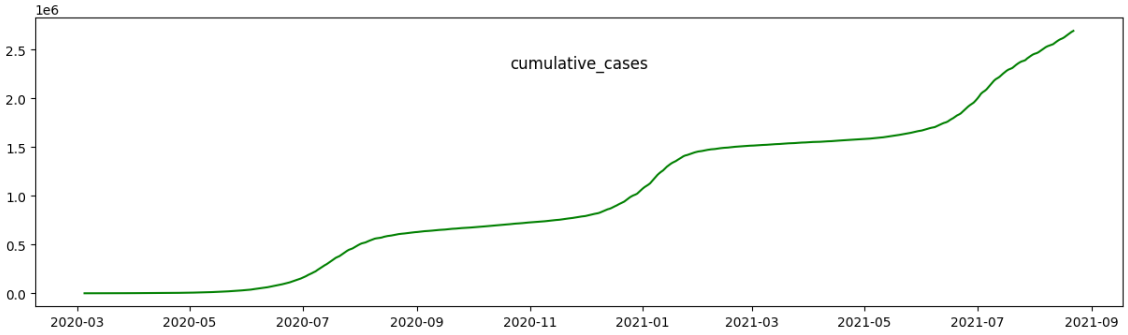

Figure 4.4: Time series plot for cumulative_cases

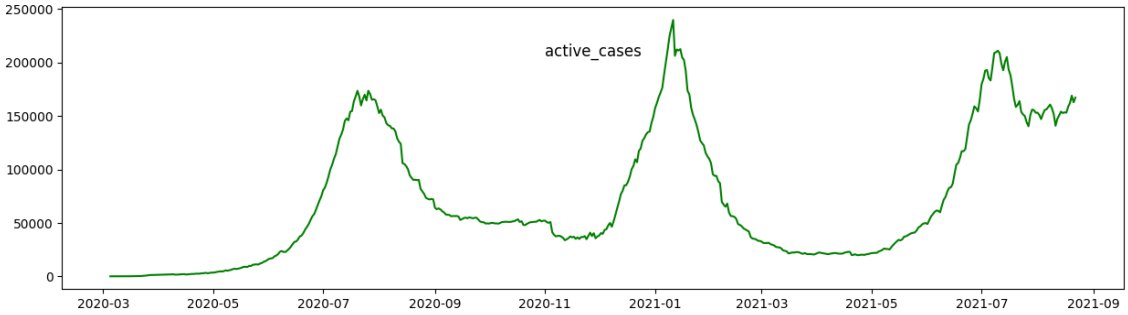

Figure 4.5: Time series plot for active_cases

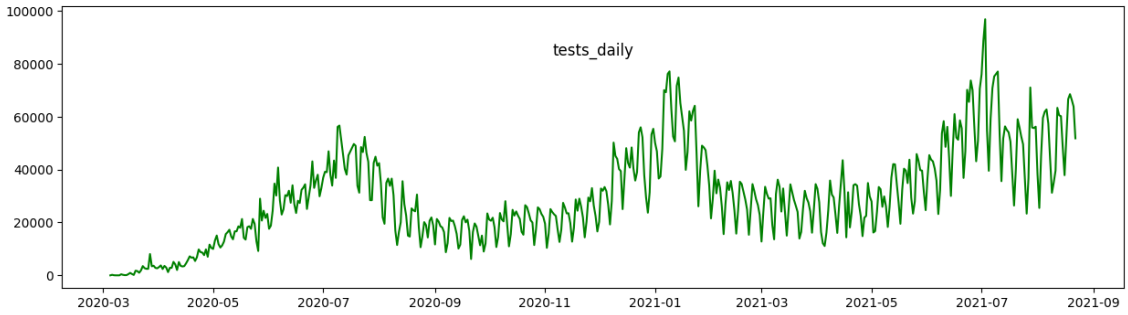

Figure 4.6: Time series plot for tests_daily

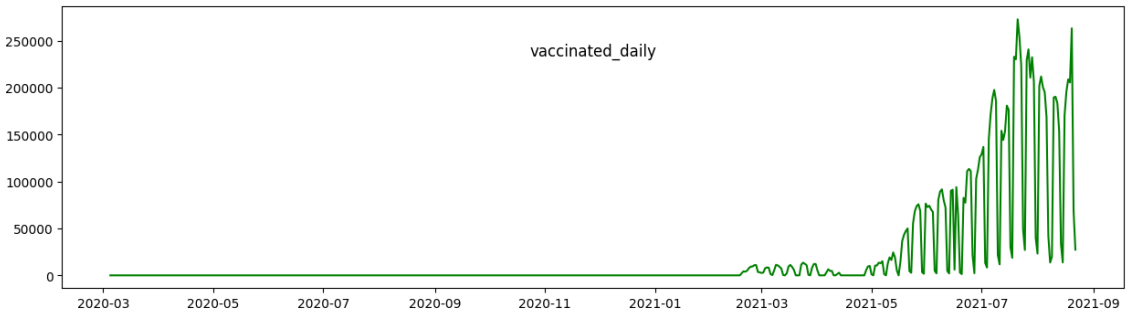

Figure 4.7: Time series plot for vaccinated_daily

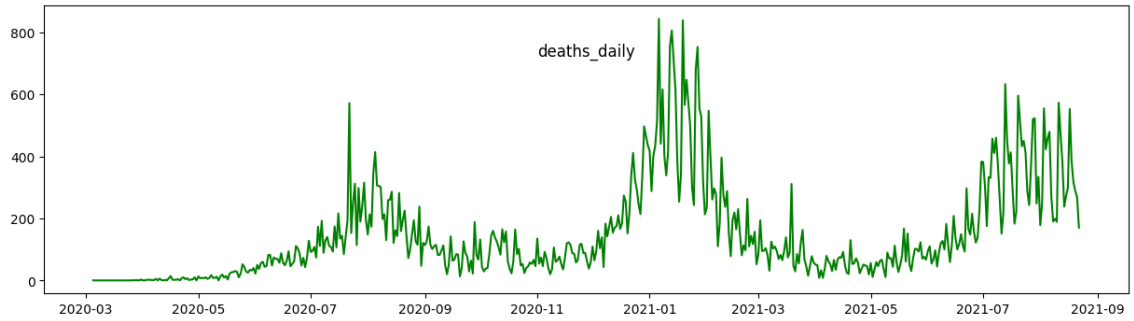

Figure 4.8: Time series plot for deaths_daily cases

It can be seen from Figure 4.2, Figure 4.5, Figure 4.6, Figure 4.8 that deaths_daily, active_cases, tests_daily, and cases_daily follow a similar trend and whenever there are spikes in each respective graph, they usually occur around the same time.

Figure 4.9, Figure 4.10, Figure 4.11, Figure 4.12, Figure 4.13, Figure 4.14, and Figure 4.15 shows plots containing box and whisker diagrams of the relevant variables to help with identifying if there are any outliers in the dataset.

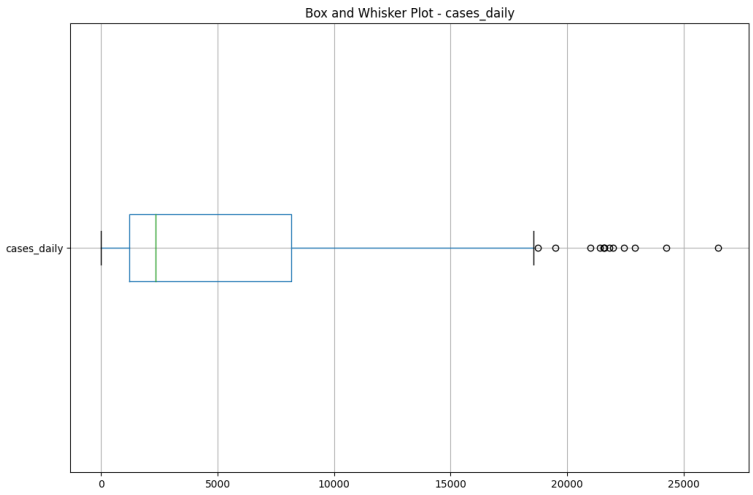

Figure 4.9: Boxplot for cases_daily

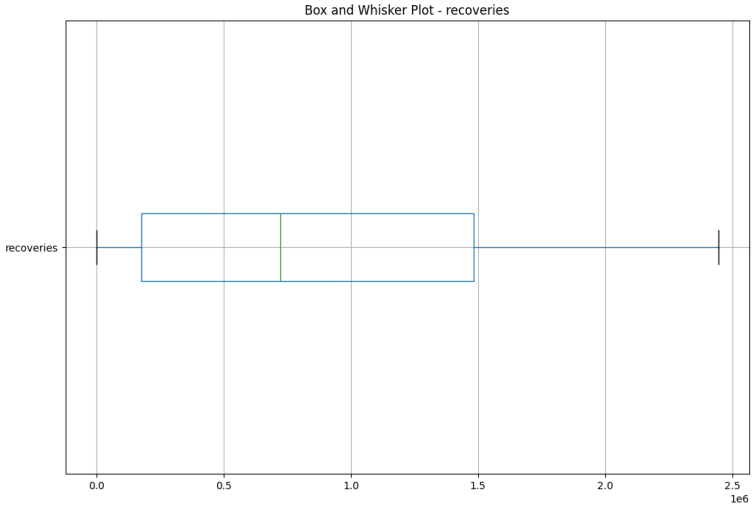

Figure 4.10: Boxplot for recoveries

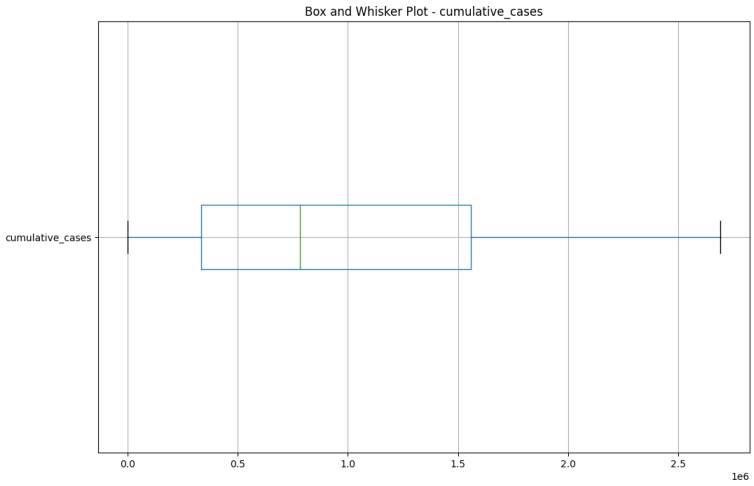

Figure 4.11: Boxplot for cumulative_cases

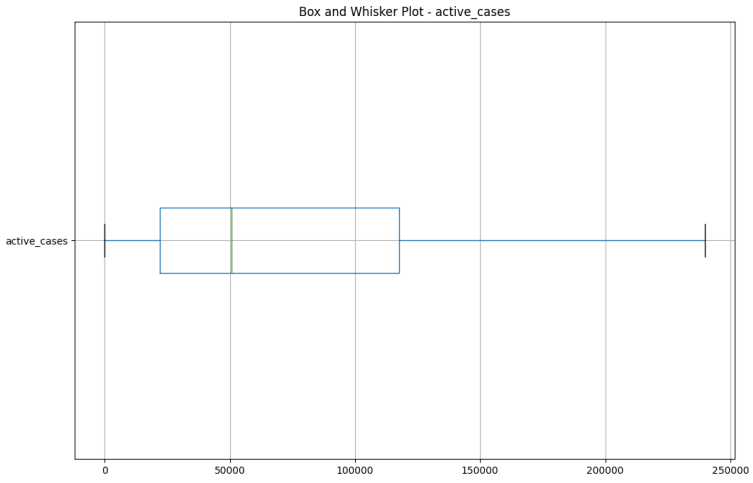

Figure 4.12: Boxplot for active_cases

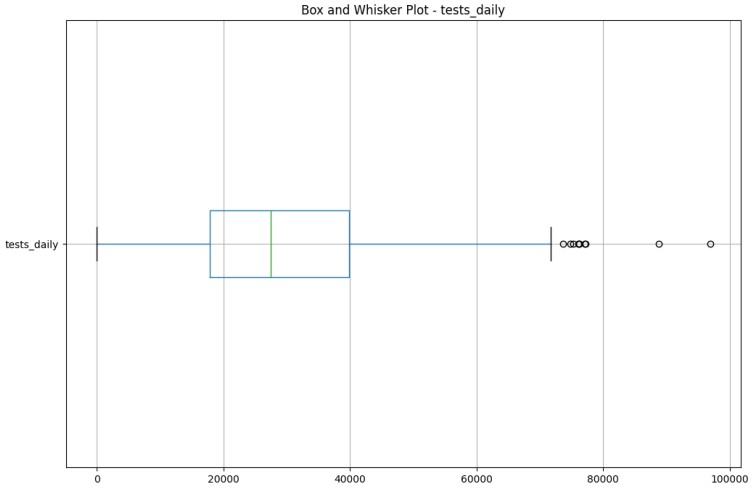

Figure 4.13: Boxplot for tests_daily

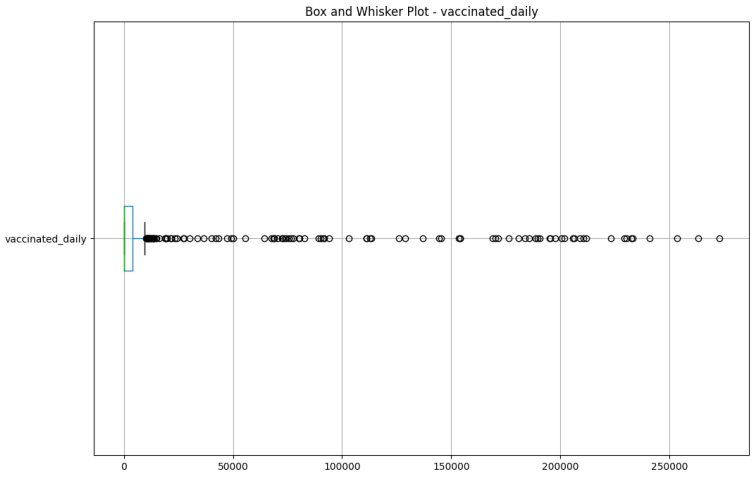

Figure 4.14: Boxplot for vaccinated_daily

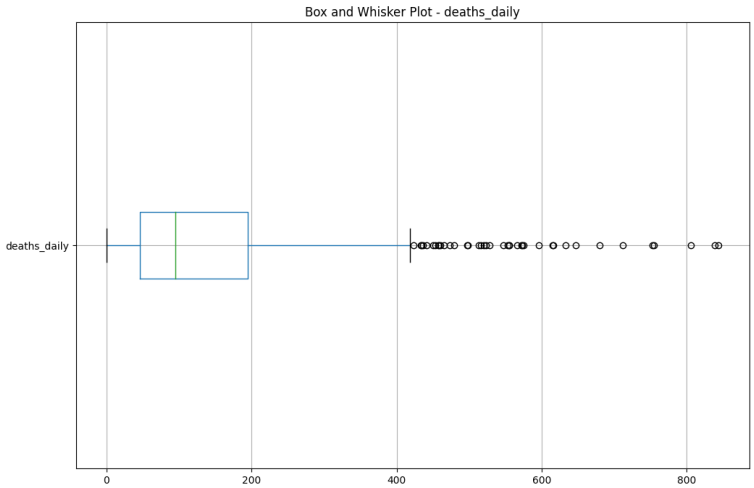

Figure 4.15: Boxplot for deaths_daily cases

From Figure 4.9, Figure 4.12, Figure 4.13 and Figure 4.15 it can be observed from the different plots that deaths_daily, active_cases, tests_daily, and cases_daily do contain some outliers in them. The outliers are a result when there is a spike observed in the deaths, active infections, number of tests as well as new infections. This occurs whenever there is a surge in the number of newly discovered infections, with spikes observed for different features. However, when looking at the box plot for the vaccinated_daily feature, most of the data points seem to be coming up as outliers. However, this is understandable when looking at the time series plot above for this particular feature it can be observed that vaccinations started about a year later after the first COVID-19 case was recorded in South Africa. Hence, the distribution of this particular feature exhibits positive skewness. Positive skewness in a dataset indicates that the tail of the distribution extends towards the right, while the majority of the data points are concentrated on the left side of the distribution. This suggests that there are relatively more lower values in the dataset compared to higher values. In other words, the data is skewed towards the higher end of the range, and there may be outliers or extreme values on the higher side of the distribution. The positive skewness can evidently be observed as well from Figure 4.16 which is a frequency distribution plot for the vaccinated_daily feature. It can be seen that the bulk of the data points are concentrated more towards the left side.

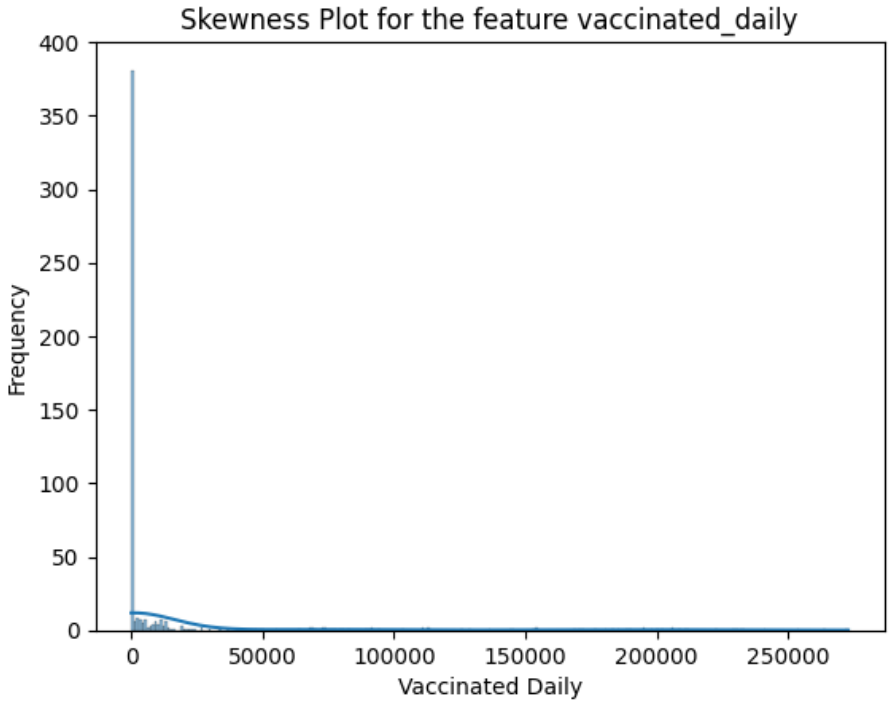

Figure 4.16: Frequency distribution plot for the vaccinated_daily feature

4.3 Results of the Training of the Models

This section presents the analysis of the learning curves which depict how the trained models performed during training and testing. Learning curves are widely used in ma-

chine learning for algorithms that learn incrementally over time. The metric used to evaluate learning could be maximising, meaning that a larger score/error indicates more learning. The converse also holds, that for metrics that are minimising, lower scores/errors indicate better learning. Ideally, for loss function curves, the error is expected to gradually decrease over training iterations or epochs.

The different LSTM models were trained using scaled data which was transformed using the MinMax scaler. Below, the loss functions for the different models that were trained will be provided as well as the hyperparameters that were used to train each model.

Persistence Model Figure 4.17 shows the predictions made by the persistence model indicated by the red line.

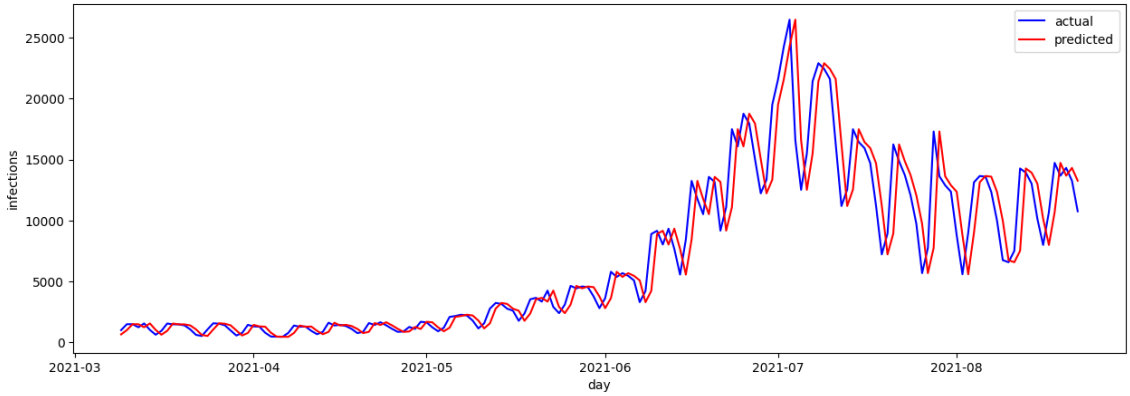

Figure 4.17: Baseline model for predicting COVID-19 infections

For this persistence model, the $RMSE = 1670.991$. This implies that on average, the predictions that were made using the persistence model are off by a magnitude of approximately 1671 infections daily. This model will act as a baseline to compare the other complex models and a model that will exhibit better performance will have a lower RMSE than the persistence model. The persistence will be used as a borderline to measure how effective the rest of the models in predicting new infections.

Basic LSTM Model

Figure 4.18 shows the learning curve depicting how the model performed during training and testing for the basic LSTM model.

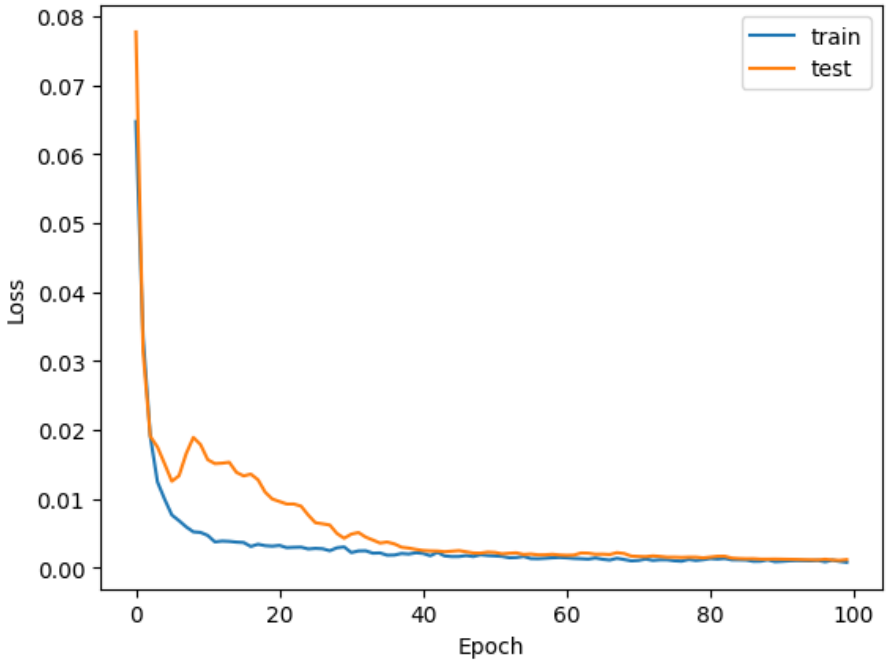

Figure 4.18: Learning curve for basic LSTM model

The learning loss curve function in Figure 4.18 behaves as expected. It can be observed that the testing and training loss curve functions decrease as the model is training. This implies that the model can fit the data.

Stacked LSTM Model

Figure 4.19 shows the learning curve depicting how the model performed during training and testing for the stacked LSTM model.

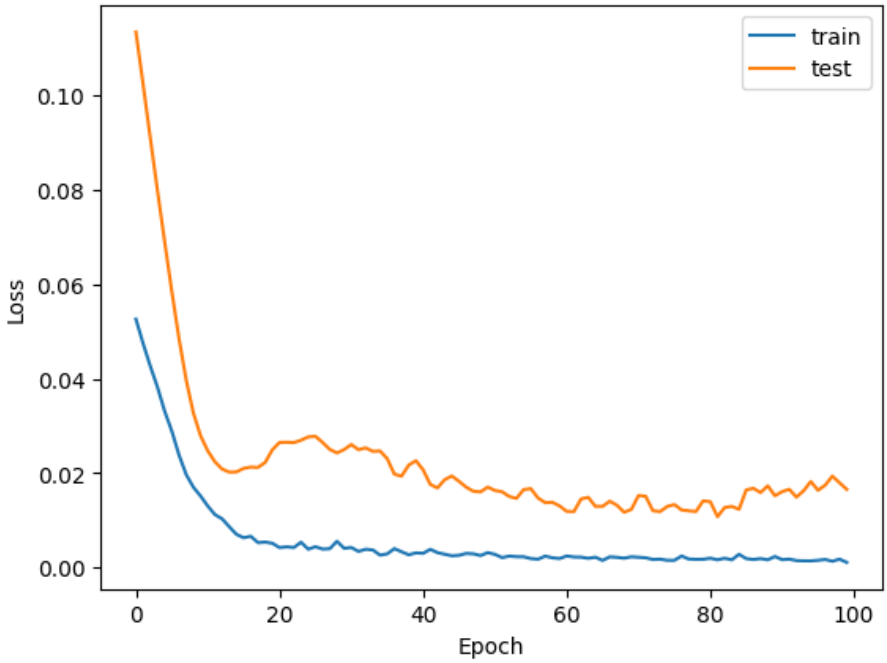

Figure 4.19: Learning curve for the stacked LSTM model

The same can be observed in the case of the learning curve for the stacked LSTM model. The training and testing are reduced with training. This implies that with more epochs, the model tends to fit the data.

Bidirectional LSTM Model

Figure 4.20 shows the learning curve depicting how the model performed during training and testing for the stacked LSTM model.

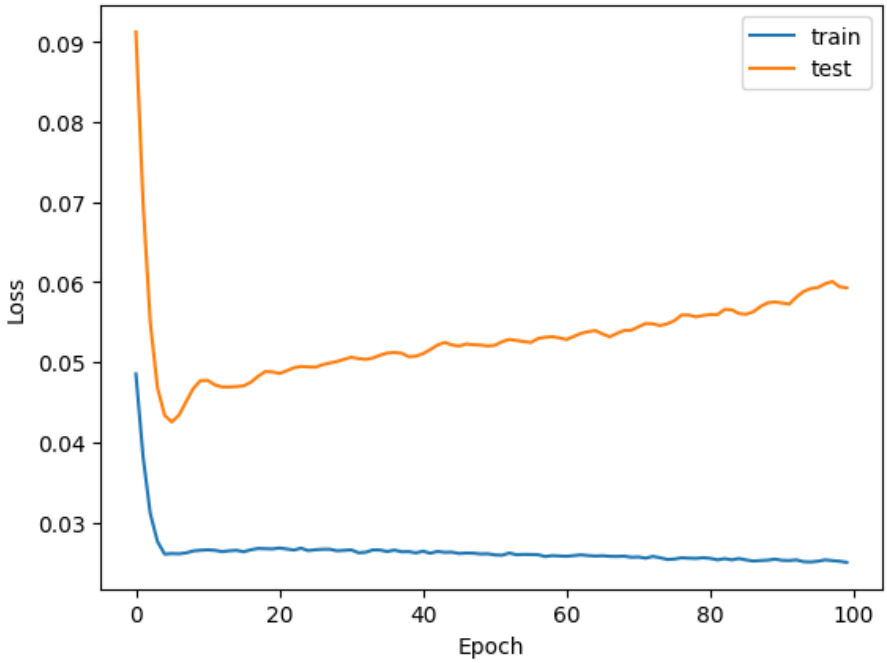

Figure 4.20: Learning curve for the bidirectional LSTM model

The learning curve for the bidirectional LSTM starts off on a good trajectory but over time the testing error curve can be seen increasing with training. This is not good as the expectation is that the testing error should also be reduced with training. This scenario may potentially suggest overfitting/underfitting the data.

4.4 Empirical Prediction Results

In this section, the different models that were trained will be used to make predictions for COVID-19 infections and check how well each model is performing by looking at the error associated with each model.

Basic LSTM Model

Figure 4.21 presents the predictions generated by the basic LSTM model for COVID-19 infections in South Africa. The figure visualises the model's forecasts alongside the actual values, providing a clear comparison of their performance. The x-axis represents the date whereas the y-axis represents the number of new infections. This graphical representation offers valuable insights into the accuracy and behavior of the model's predictions, shedding light on its capabilities and potential areas for improvement.

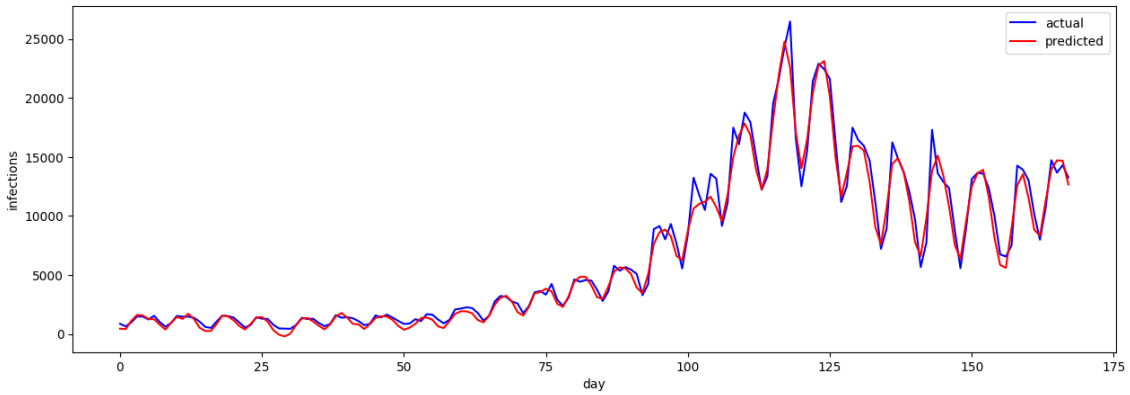

Figure 4.21: Predictions for COVID-19 infections using the basic LSTM model

Figure 4.21 provides an illustration of the predicted infections comparatively with the actual infections and the $RMSE$ for the predictions is 1071.551. The RMSE suggests that on average, the predictions being made for COVID-19 new infections using the basic LSTM model are off by a magnitude of approximately 1072 infections daily. The RMSE obtained from making predictions using the basic LSTM model is lower than that of the persistence model, thus suggesting that the basic LSTM model outperforms the persistence model when predicting COVID-19 infections in South Africa and yields better predictions.

Stacked LSTM Model

Similarly, the predictions made using the stacked LSTM model are depicted in Figure 4.22.

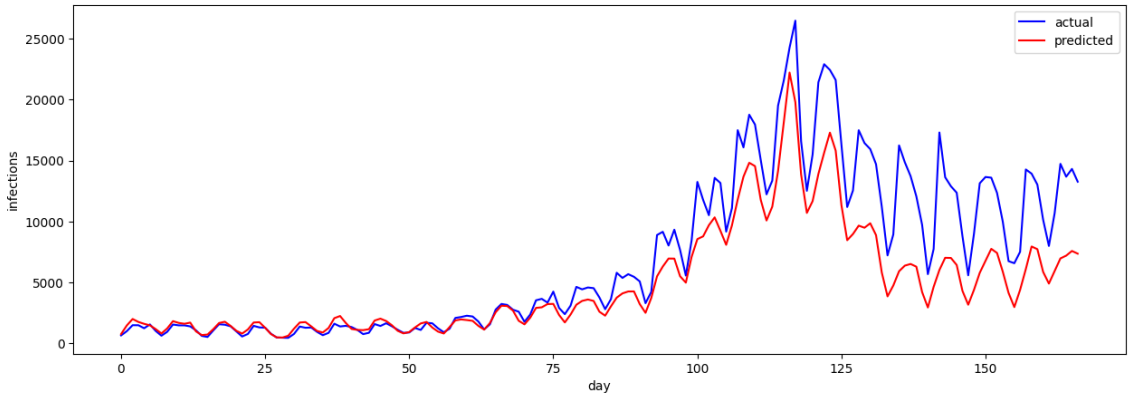

Figure 4.22: Predictions for COVID-19 infections using the stacked LSTM model

Figure 4.22 provides an illustration of the predicted infections comparatively with the actual infections and the $RMSE$ for the predictions made using the stacked LSTM model is 3098.076. The RMSE suggests that on average, the predictions being made for COVID-19 new infections using the stacked LSTM model are off by a magnitude of approximately 3098 infections daily. The RMSE obtained from making predictions using the stacked LSTM model is greater than that of the persistence model, thus suggesting that the stacked LSTM model does not outperform the persistence model when predicting

COVID-19 infections in South Africa and yields less accurate predictions.

Bidirectional LSTM Model

Similarly, the predictions made using the bidirectional LSTM model are depicted in Figure 4.23.

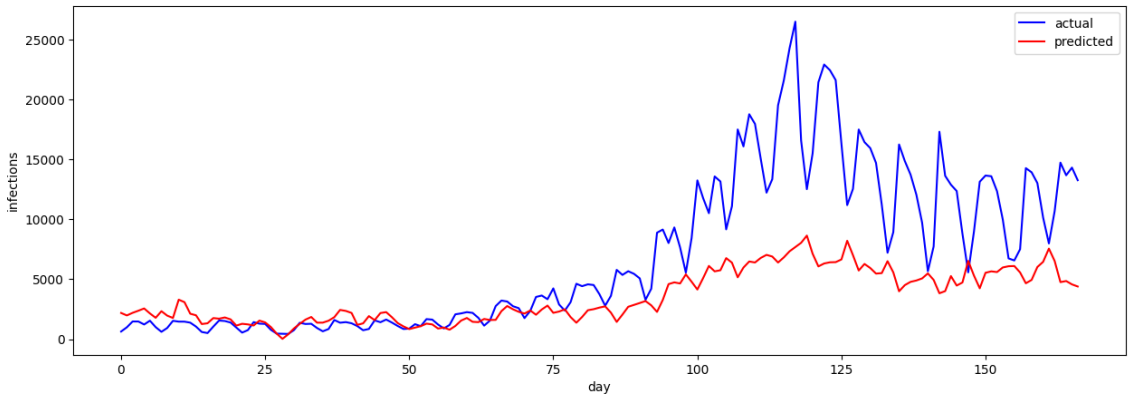

Figure 4.23: Predictions for COVID-19 infections using the bidirectional LSTM model

Figure 4.23 provides an illustration of the predicted infections comparatively with the actual infections and the $RMSE$ for the predictions made using the bidirectional LSTM model is 6417.021. The RMSE suggests that on average, the predictions being made for COVID-19 new infections using the bidirectional LSTM model are off by a magnitude of approximately 6417 infections daily. The RMSE obtained from making predictions using the bidirectional LSTM model is greater than that of the persistence model, thus suggesting that the bidirectional LSTM model does not outperform the persistence model when predicting COVID-19 infections in South Africa and yields less accurate predictions.

4.5 Discussions

There are different LSTM network architectures available and this work focused on building the basic LSTM network, stacked LSTM as well as the bidirectional LSTM. Table 4.2 provides a summary of the results that were presented in Section 4.3.

Table 4.2: Model performances

Model	**RMSE**
Persistence model	1670.991
Basic LSTM	1071.551
Stacked LSTM	3098.076
Bidirectional LSTM	6417.021

The persistence model serves as a borderline to check how well the created models are performing. It can be seen that only the basic LSTM had a root mean squared error below the persistence model and the other models were performing worse than the persistence model. When comparing different models based on the RMSE, a lower RMSE suggests that a model is better at making predictions compared to a model with a higher RMSE.

During model building, common hyperparameters that do not depend on model architecture were kept consistent across all three LSTM models. This was done to ensure a fair comparison between the models, as differences in performance can be attributed to architectural variations rather than differences in training conditions.

The predictions made using the different LSTM network architectures look promising in the sense that the basic LSTM model and the stacked LSTM model were able to predict the wave of infections between day 75 to about day 150 on Figure 4.21 and Figure 4.22. This is a major improvement when compared to the work done by Djilali and Ghanbari (2020) which failed to capture the second wave of infections and suggested COVID-19 to become obsolete in South Africa by around November 2020, which was not the case and can be supported by Figure 1.1 which shows new COVID-19 infections post November 2020.

The disparity between this study and the study by Djilali and Ghanbari (2020) lies in their employed methodologies. While this study uses a deep learning approach to predict COVID-19 infections in South Africa, the study by Djilali and Ghanbari (2020) utilised a mathematical modeling approach to model COVID-19 transmission. Despite these methodological differences, both studies share commonalities in terms of geographical location and overarching objective. Geographically, they both concentrated on the same country South Africa, ensuring a consistent contextual framework. Moreover, the primary goal of both studies was aligned, aiming to explore and understand the transmission of COVID-19 infections in South Africa. It is important to highlight that the principal objective of the the study by Djilali and Ghanbari (2020) was to forecast the initial peak of infections. This study showcased notable enhancement by successfully capturing additional waves of infections.

This study also shows a bigger improvement when compared to the findings of Chandra et al. (2022). The main disparity between this study and the study by Chandra et al. (2022) is the geographical location where the two studies took place. One study makes predictions of new infections using a dataset from India whereas the other study makes predictions using a dataset from South Africa. Also, the nature of studies was different in the sense that the study by Chandra et al. (2022) only focused on regions that were COVID-19 hotspot areas so essentially did not include the whole population when modeling was done whereas this study included all regions irrespective of whether it is a hotspot or not. This results in predictions that are inclusive of the whole population rather than a subset of the population.

Looking at the different RMSEs for the LSTM models developed by Chandra et al. (2022),

the bidirectional LSTM always performed better than the basic LSTM model which is not the case for our study. From the learning curve of the bidirectional model depicted by Figure 4.20, it can be seen that the model may be overfitting. This is evident as the testing loss function increases as training progresses. Also, this model does not perform well over unseen data, this is evident when looking at Figure 4.23 and also looking at the RMSE that was obtained when making predictions on testing data. The model tends to be performing the worst when compared to the other models.

There were some challenges and gaps identified in this study, one of them being that no hyperparameter tuning was performed but the hyperparameters that were used were based on existing literature and there is absolutely no certainty if those are the best hyperparameters that can be used to train the deep learning models for this particular dataset. It would be interesting to investigate using some common methods like grid search, random search, genetic algorithms, and other relevant methods to check if no other hyperparameters can perform better than the current hyperparameters used. Another shortcoming of this study is that the predictions were made for South Africa overall using a dataset that was already aggregated at a country level and did not cater to subpopulations individually. The demographics are not necessarily similar across the provinces and it would be interesting to see how they differ and also test how the overarching model perform when making prediction on subpopulations.

Chapter 5: Conclusion and Recommendations

In this work, the main aim was to employ deep learning techniques for predicting COVID-19 infections in South Africa, with a focus on enhancing predictive capacity modeling on COVID-19 new virus infections. There were some concerns regarding the surge of COVID-19 infections and the effects it might have on the decision-makers in the healthcare sector to efficiently manage the strain the virus would have on the healthcare system in terms of resource availability as well as resource allocation. Section 1.1 provided background examples of similar studies conducted by other researchers, revealing gaps and challenges. The gaps and challenges that were identified were also mentioned. Various methodologies were employed, including statistical predictive analysis (logistic regression model), mathematical epidemiological modeling (SEIR), and machine learning approaches (deep learning). Studies using mathematical and statistical modeling had limitations in predicting future infection waves, unlike those using machine learning. Hence, the main aim of undertaking this study was to explore the feasibility of employing a specific subset of machine learning, specifically deep learning, to forecast COVID-19 infections in South Africa.

To achieve this aim, three objectives were outlined. The first objective involved performing a predictive analysis using the proposed LSTM networks. This was accomplished by training various LSTM networks with the SA COVID-19 dataset and assessing their performance on testing data. The study's results were then compared to historical studies, starting with a comparison to Djilali and Ghanbari (2020). The key difference between the two studies lies in their methodologies, with one using mathematical modeling and the other employing deep learning. It is important to highlight that the principal objective of the mathematical modeling was to forecast the initial peak of infections. The deep learning approach showcased notable enhancement by successfully capturing additional waves of infections. Additionally, a comparison was made with the study by Chandra et al. (2022), revealing similarities in the approach despite different datasets from various countries. The conclusion is that the performance of the deep learning models appears commendable, exhibiting enhanced predictability compared to previous studies. A notable improvement lies in their ability to accurately predict an additional wave of infections, a capability that was lacking in the study conducted by Chandra et al. (2022).

The second objective of this study was to empirically test the proposed machine learning algorithms on a South African COVID-19 dataset. It was fulfilled by testing and doing predictions of new infections in South Africa and plots were provided that showed the predictions in comparison to the testing set. A baseline persistence model was created and was used to test how effective the trained LSTM models were in predicting new infections. The RMSE was used as a benchmark metric to comment on how effective the models

were performing when making predictions for new infections. The basic LSTM was the best model to use to make predictions as it had the lowest RMSE value. When looking at the RMSE for the stacked LSTM model and the bidirectional, they were greater than the RMSE for the baseline model, thus suggesting that the two models did not perform better when making predictions for new COVID-19 infections compared to the baseline model. The results from this study contradict the results from the study by Chandra et al. (2022), as in this study the model that was shown to do better predictions comparatively to the other models was the basic LSTM model and the bidirectional proved to be the worst performing as it had the greatest RMSE. The learning curve that was obtained when training the bidirectional LSTM model suggested that there was the potential of overfitting on the data whereas in the study by Chandra et al. (2022), the bidirectional LSTM was the better model overall in making predictions and this was evident from the RSME that were obtained when the different models they created were used to make predictions across all the different regions. The conclusion was that the basic LSTM model was the best model for predicting COVID-19 infections in South Africa as it had the lowest RMSE.

The last objective that had to be met was to provide recommendations on the gaps and challenges identified in this study and how they could be addressed for future studies. The main challenge and gap identified in this work was that the predictions were made for South Africa overall using a dataset that was already aggregated at a country level and did not cater to subpopulations individually. This study therefore recommends that future studies be conducted to explore and recognise diverse trends within various provinces in SA and check if the new cases follow a similar trend and have a wave of infections occurring at the same time. If not, then multiple models can be developed each catering to the unique dynamics exhibited by the different provinces, this would ensure accuracy in the predictions. After the multiple varying models have been created, the models would have to be harmonised and unified into a single comprehensive solution which would be making predictions at a country level and then performing rigorous validation and cross-validation to evaluate the performance of each model independently and in combination. This approach would allow for the validation of the viability of combined models while preserving the precision of individual province models.

It was observed that the bidirectional LSTM model which was performing worst in this study was a better model in a different study that was making predictions for COVID-19 infections and considering only hotspot areas. This study recommends testing if this would be the case using a South African dataset, only focusing on COVID-19 hotspot areas and checking if the bidirectional LSTM would give the best results. Also, perform an extensive analysis to compare the differences and similarities between the datasets used in this study and the study by Chandra et al. (2022) to understand why the worst performing model in this study was the best performing model in the study by Chandra et al. (2022).

Another shortcoming of this work is that the created forecasting models were not deployed and are not in production. Decision-makers cannot utilise the models. In the

future, this study recommends creating a dashboard based on the predictions made using the trained models which can be utilised by an external user to retrieve the results of the forecasts and make informed decisions on time if there is a wave of infections without having to rely much on the developer to provide the results of forecasts to them.

Bibliography

Allen, L. J. 2008. An introduction to stochastic epidemic models. In *Mathematical epidemiology* (pp. 81–130). Springer.

Ba, J. & Frey, B. 2013. Adaptive dropout for training deep neural networks. *Advances in neural information processing systems*, *26*.

Bayes, T. 1968. Naive bayes classifier. *Article Sources and Contributors*, (pp. 1–9).

BBC News 2021. COVID-19 Africa: What is happening with vaccine supplies? [Online; accessed August 12, 2021] `https://www.bbc.com/news/56100076`.

Berrar, D. 2018. Bayes' theorem and naive Bayes classifier. *Encyclopedia of bioinformatics and computational biology: ABC of bioinformatics*, *403*, 412.

Bhorat, H., Köhler, T., Oosthuizen, M., Stanwix, B., Steenkamp, F., & Thornton, A. 2020. The economics of COVID-19 in South Africa: Early impressions.

Blumberg, L. H., Jassat, W., Mendelson, M., & Cohen, C. 2020. The COVID-19 crisis in South Africa: Protecting the vulnerable.

Chandra, R., Jain, A., & Singh Chauhan, D. 2022. Deep learning via LSTM models for COVID-19 infection forecasting in India. *PloS one*, *17*(1), e0262708.

Cohen, J. 2021. South Africa suspends use of AstraZeneca's COVID-19 vaccine after it fails to clearly stop virus variant. *Science*, *10*.

Cooper, I., Mondal, A., & Antonopoulos, C. G. 2020. A SIR model assumption for the spread of COVID-19 in different communities. *Chaos, Solitons & Fractals*, *139*, 110057.

Coronavirus in South Africa 2021. Daily confirmed cases. [Online; accessed August 12, 2021] `https://mediahack.co.za/datastories/coronavirus/dashboard/`.

Djilali, S. & Ghanbari, B. 2020. Coronavirus pandemic: A predictive analysis of the peak outbreak epidemic in South Africa, Turkey, and Brazil. *Chaos, Solitons & Fractals*, *138*, 109971.

Farzad, A., Mashayekhi, H., & Hassanpour, H. 2019. A comparative performance analysis of different activation functions in LSTMnetworks for classification. *Neural Computing and Applications*, *31*(7), 2507–2521.

Fusion Medical Animation 2020. New visualisation of the COVID-19 virus. [Online; accessed August 19, 2021] `https://unsplash.com/photos/rnr8D3FNUNY`.

Gaeta, G. 2020. A simple SIR model with a large set of asymptomatic infectives. *arXiv preprint arXiv:2003.08720*.

Goyal, P., et al. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

Gupta, V. K., Gupta, A., Kumar, D., & Sardana, A. 2021. Prediction of COVID-19 confirmed, death, and cured cases in India using random forest model. *Big Data Mining and Analytics*, *4*(2), 116–123.

Gurney, K. 2018. *An introduction to neural networks*. CRC press.

Hinton, G. & Sejnowski, T. J. 1999. *Unsupervised learning: foundations of neural computation*. MIT press.

Ihianle, I. K., Nwajana, A. O., Ebenuwa, S. H., Otuka, R. I., Owa, K., & Orisatoki, M. O. 2020. A deep learning approach for human activities recognition from multimodal sensing devices. *IEEE Access*, *8*, 179028–179038.

Izza, Y., Ignatiev, A., & Marques-Silva, J. 2020. On explaining decision trees. *arXiv preprint arXiv:2010.11034*.

James, G., Witten, D., Hastie, T., Tibshirani, R., et al. 2013. *An introduction to statistical learning*, volume 112. Springer.

Jolliffe, I. T. 1990. Principal component analysis: a beginner's guide—i. introduction and application. *Weather*, *45*(10), 375–382.

Kecman, V. 2005. Support vector machines–an introduction. In *Support vector machines: theory and applications* (pp. 1–47). Springer.

Khalilpourazari, S. & Hashemi Doulabi, H. 2021. Designing a hybrid reinforcement learning based algorithm with application in prediction of the COVID-19 pandemic in Quebec. *Annals of Operations Research*, (pp. 1–45).

Khanzode, K. C. A. & Sarode, R. D. 2020. Advantages and disadvantages of artificial intelligence and machine learning: A literature review. *International Journal of Library & Information Science (IJLIS)*, *9*(1), 3.

Kubat, M. & Kubat 2017. *An introduction to machine learning*, volume 2. Springer.

Kurniawan, R., Abdullah, S. N. H. S., Lestari, F., Nazri, M. Z. A., Mujahidin, A., & Adnan, N. 2020. Clustering and correlation methods for predicting coronavirus COVID-19 risk analysis in pandemic countries. In *2020 8th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1–5).: IEEE.

Li, H., Xu, Z., Taylor, G., Studer, C., & Goldstein, T. 2018. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, *31*.

Li, Y., Zhu, Z., Kong, D., Han, H., & Zhao, Y. 2019. EA-LSTM: Evolutionary attention-based LSTM for time series prediction. *Knowledge-Based Systems*, *181*, 104785.

Lone, S. A. & Ahmad, A. 2020. Covid-19 pandemic–an african perspective. *Emerging microbes & infections*, *9*(1), 1300–1308.

Madhi, S. 2021. COVID-19 herd immunity v. learning to live with the virus. *South African Medical Journal*.

Martcheva, M. 2015. *An introduction to mathematical epidemiology*, volume 61. Springer.

Mbuvha, R. & Marwala, T. 2020. Bayesian inference of COVID-19 spreading rates in South Africa. *PloS one*, *15*(8), e0237126.

McCaffrey, J. D. 2018. Why you should use cross-entropy error instead of classification error or mean squared error for neural network classifier training. *Last accessed: Feb 2022*.

Medsker, L. R. & Jain, L. 2001. Recurrent neural networks. *Design and Applications*, *5*(64-67), 2.

Million, E. 2007. The hadamard product. *Course Notes*, *3*(6), 1–7.

Ndiaye, B. M., Tendeng, L., & Seck, D. 2020. Comparative prediction of confirmed cases with COVID-19 pandemic by machine learning, deterministic and stochastic SIR models. *arXiv preprint arXiv:2004.13489*.

Odeku, K. O. 2021. The impact of the COVID-19 pandemic on black-owned small and micro-businesses in South Africa. *Academy of Entrepreneurship Journal*, *27*, 1–5.

Palei, S. K. & Das, S. K. 2009. Logistic regression model for prediction of roof fall risks in bord and pillar workings in coal mines: An approach. *Safety science*, *47*(1), 88–96.

Panigutti, C., Monreale, A., Comandé, G., & Pedreschi, D. 2022. Ethical, societal and legal issues in deep learning for healthcare. In *Deep Learning in Biology and Medicine* (pp. 265–313). World Scientific.

Parbat, D. & Chakraborty, M. 2020. A python based support vector regression model for prediction of COVID-19 cases in India. *Chaos, Solitons Fractals*, *138*, 109942, `https://doi.org/https://doi.org/10.1016/j.chaos.2020.109942 https://www.sciencedirect.com/science/article/pii/S0960077920303416`.

Patel, E. & Kushwaha, D. S. 2020. Clustering cloud workloads: K-means vs gaussian mixture model. *Procedia computer science*, *171*, 158–167.

Phaisangittisagul, E. 2016. An analysis of the regularization between L2 and dropout in single hidden layer neural network. In *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)* (pp. 174–179).: IEEE.

Pradhan, R. P. & Kumar, R. 2010. Forecasting exchange rate in India: An application of artificial neural network model. *Journal of Mathematics research*, *2*(4), 111.

Ramachandran, P., Agarwal, S., Mondal, A., Shah, A., & Gupta, D. 2021. S++: A fast and deployable secure-computation framework for privacy-preserving neural network training. *arXiv preprint arXiv:2101.12078*.

Ramachandran, P., Zoph, B., & Le, Q. V. 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

Rath, S., Tripathy, A., & Tripathy, A. R. 2020. Prediction of new active cases of coronavirus disease (COVID-19) pandemic using multiple linear regression model. *Diabetes & metabolic syndrome: clinical research & reviews*, *14*(5), 1467–1474.

Rios, R. A., Nogueira, T., Coimbra, D. B., Lopes, T. J., Abraham, A., & Mello, R. F. d. 2021. Country transition index based on hierarchical clustering to predict next COVID-19 waves. *Scientific reports*, *11*(1), 15271.

Romadhon, M. R. & Kurniawan, F. 2021. A comparison of naive bayes methods, logistic regression and KNN for predicting healing of COVID-19 patients in Indonesia. In *2021 3rd east Indonesia conference on computer and information technology (eiconcit)* (pp. 41–44).: IEEE.

Safdar, N. M., Banja, J. D., & Meltzer, C. C. 2020. Ethical considerations in artificial intelligence. *European Journal of Radiology*, *122*, 108768, `https://doi.org/https://doi.org/10.1016/j.ejrad.2019.108768` `https://www.sciencedirect.com/science/article/pii/S0720048X19304188`.

Shah, N. H. & Gupta, J. 2013. SEIR model and simulation for vector borne diseases.

Singhal, A., Singh, P., Lall, B., & Joshi, S. D. 2020. Modeling and prediction of COVID-19 pandemic using Gaussian mixture model. *Chaos, Solitons & Fractals*, *138*, 110023.

Sutton, R. S. & Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Sykes, A. O. 1993. An introduction to regression analysis.

Tamang, S., Singh, P., & Datta, B. 2020. Forecasting of COVID-19 cases based on prediction using artificial neural network curve fitting technique. *Global Journal of Environmental Science and Management*, *6*(Special Issue (COVID-19)), 53–64.

Tátrai, D. & Várallyay, Z. 2020. COVID-19 epidemic outcome predictions based on logistic fitting and estimation of its reliability. *arXiv preprint arXiv:2003.14160*.

Tornatore, E., Buccellato, S. M., & Vetro, P. 2005. Stability of a stochastic SIR system. *Physica A: Statistical Mechanics and its Applications*, *354*, 111–126.

Uyanık, G. K. & Güler, N. 2013. A study on multiple linear regression analysis. *Procedia-Social and Behavioral Sciences*, *106*, 234–240.

Wang, H. 2003. Hierarchical clustering. *Introduction to Computational Intelligence*, (pp. 191).

Wang, P., Zheng, X., Li, J., & Zhu, B. 2020. Prediction of epidemic trends in COVID-19 with logistic model and machine learning technics. *Chaos, Solitons Fractals*, *139*, 110058, `https://doi.org/https://doi.org/10.1016/j.chaos.2020.110058` `https://www.sciencedirect.com/science/article/pii/S0960077920304550`.

Weiss, H. H. 2013. The SIR model and the foundations of public health. *Materials matematics*, (pp. 0001–17).

Yacim, J. A. & Boshoff, D. G. B. 2018. Impact of artificial neural networks training algorithms on accurate prediction of property values. *Journal of Real Estate Research*, *40*(3), 375–418.

Zisad, S. N., Hossain, M. S., Hossain, M. S., & Andersson, K. 2021. An integrated neural network and SEIR model to predict COVID-19. *Algorithms*, *14*(3), 94.