National College of Ireland

Forecasting the Novel Coronavirus(COVID-19) using Time Series Model

MSc Research Project
Data Analytics

Harsh Chudasama
Student ID: X18187340

School of Computing
National College of Ireland

Supervisor: Dr. Muhammad Iqbal

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Harsh Chudasama
Student ID:	X18187340
Programme:	Data Analytics
Year:	2020
Module:	MSc Research Project
Supervisor:	Dr. Muhammad Iqbal
Submission Due Date:	17/07/2020
Project Title:	Forecasting the Novel Coronavirus(COVID-19) using Time Series Model
Word Count:	3900
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	27th September 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Forecasting the Novel Coronavirus(COVID-19) using Time Series Model

Harsh Chudasama
X18187340

Abstract

The 2019 novel coronavirus (COVID-19), which originated from China, has spread quickly among individuals living in different nations and spreading quickly throughout the globe with nearly 20 million cases overall as indicated by the insights of the European Center for Disease Prevention and Control. Researchers from all over the world are working together to develop the vaccine as this virus is highly contagious which spreads through human contact and while taking into account the high pace of the disease spread and the critical number of fatalities. Scientists have made good progress in developing the vaccines which are at the early stages of the clinical trials[1] and hoping soon for a cure, but in the meantime death toll is increasing day by day. This research primarily focusses on the forecasting of confirmed, death, and recovered cases using the time series model. In this research, various models were used namely LSTM, Prophet, ARMA, and ARIMA for forecasting the spread of virus-based in India, and results were evaluated.LSTM outperformed the other three models based on the evaluation matrix with least MAPE of 1.18 % and R^2 of 0.9997.

1 Introduction

1.1 Background & Motivation

COVID-19 is characterized as another kind of coronavirus that spreads quickly from individual to individual and turns into a significant pandemic that causes an extraordinary misfortune (Ceylan; 2020).Covid-19 also known as Novel Coronavirus which has evolved from its predecessors namely the Severe Acute Respiratory Syndrome Coronavirus (SARS-CoV) and the Middle East Respiratory Syndrome Coronavirus (MERS-CoV) since the last decade (Andersen et al.; 2020) . To date, no vaccine has been developed and it's still under clinical trials. To date, no vaccine has been developed and it's still under clinical trials. Hence, it's important to analyze the key symptoms along with forecasting the spread of viruses using machine learning models to gather data and help researchers to develop vaccines in a quick period.

1.2 Importance & Objective

In December 2019, Novel Coronavirus (COVID-19) showed up in Wuhan city, China. As of August 11, 2020, more than 2 million COVID-19 cases were confirmed worldwide,

[1]Data Source : `https://clinicaltrials.gov/`

including 741,787 deaths where more than 65% have recovered [2].There is a pressing

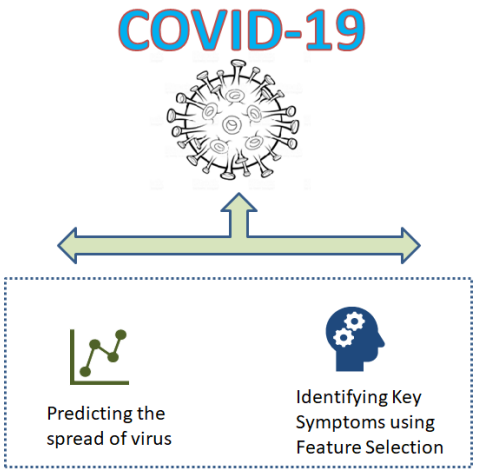


Figure 1: Project Objective

need to screen and anticipate COVID-19 predominance to control this spread all the more adequately. The current advancement in the field of machine learning, Time Series Forecasting models have evolved and proved to be efficient in predicting the impact of the spread of the virus at early stages and taking the required actions to respond to this catastrophe. The project objective is to accurately predict the COVID-19 Cases in India (Confirmed, Recovered, and Deaths) along with the extraction of key symptoms using feature selection (Refer to Figure 1).Machine learning models such as LSTM Neural Network and Time Series model ARIMA & ARMA along with the FB Prophet model will be deployed for testing.

1.3 Research Question

" Can machine learning algorithms like Time series model accurately predict the impact of COVID-19 in India at early stages ?"

1.4 General structure of a document

This research paper aims at providing accurate and latest research related to analyzing the impact of COVID-19 using Machine learning approach. The structure of the document is as follows: Section 2 will outline the current research done using various techniques and their limitations. Followed by (Section 3) Research Methodology (i.e. KDD) will be explained in detail. Section 4 describes the project implementation along with the techniques used on different models. Section 5 describes the performance matrix used to evaluate the results and Finally, the report will be concluded along with its future scope. All references have been cited using Harvard style at the end of the report by following the referencing guidelines provided by NCI Library [3].

[2]COVID-19 pandemic updates `https://www.worldometers.info/coronavirus/`

[3]NCI Library Referencing Guide : `https://libguides.ncirl.ie/ld.php?content_id=32356248`

2 Related Work

2.1 Introduction

The medical field has been evolving in the current era with a cure for almost all diseases compared to research done 20 years ago. Technology has played an important role as it has made a detailed analysis process easier with predefined computer software and simulations. Forecasting the amount of rainfall in a region or predicting the impact of COVID-19 these fall under the category of Time series forecasting models. In this project, Time Series Model like ARIMA, ARMA, and Prophet will be deployed and tested. Refer to Figure 2 for the Popular techniques.

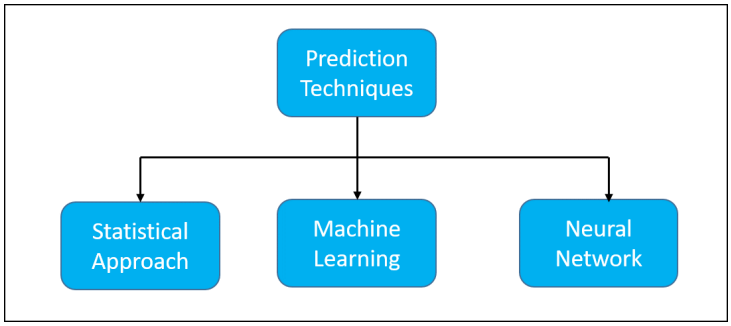


Figure 2: Popular techniques for predicting COVID-19 Cases

Refer below Subsection 2.2 , Subsection 2.3 and subsection 2.4 to know more about popular techniques being used by researchers to predict COVID-19 Cases.

2.2 Statistical Approach

Based on research done by the Zhong, Mu, Li, Wang, Yin and Liu (2020), they had compiled a list of data gathered through the Chinese government and data shared by the WHO during the early stages of the spread of the virus way back in Dec 2019. That time the impact of the virus was at the minimal stages with majority cases seen in Wuhan, China. Researchers applied a simple mathematical model to forecast the cases for February 2020 i.e. Analyzing the count of the affected person by virus after 3 months. The simplified SIR model was implemented with the basic assumption that population change won't impact the outcome.

The result of the SIR Model predicted the cases to rise over 3 months with predicted cases in the range of 76,000 to 2,30,000. If necessary precautions are taken then cases could reduce up to 45%. But as the majority of data was missing with incomplete information so this model was rejected at the end by the research community.

Nesteruk (2020) focused on predicting the impact of the spread of viruses based on Mainland China. They developed the SIR model with detailed analysis and provided accurate predictions with the highest correlation coefficient (r) of 0.997966487046645 along with a Susceptible count (S) of 45,579. Their research was taking the time to validate the results due to the unavailability of the latest daily count report.

2.3 Machine Learning Approach

(Tárnok; 2020) is a Cytometry expert who was trying to find out whether a single cell of the virus can provide vital information such as the patient is COVID positive or not. He proposed to import the data collected from the cell and give it as input to binary classification algorithm by defining rules to predict accurately whether a person had mild or severe symptoms based on multiple parameters extracted from the cell genome also called as multi-OMICS.

(Attila; 2020) proposed a new approach to improve the rapid testing by using point-of-care (PoC) devices which provide the test results within a couple of hours. In this, they created a prototype which was running background job of classification of a test result being positive or negative. A machine learning code was proposed to analyze the same and provide the outcome. Decision trees and Random forest models were given as a pretrained model.

In one case study, (Zhao1 et al.; 2020) analyzed the patient data and extracted common symptoms that show early signs of the patient being affected by COVID-19. Their research focused on detecting fever at early stages using the Linear regression model and the results were outstanding with accuracy of 0.951 along with the precision 0.943.

(Jang, Seongpil et al.; 2016) performed the analysis and compared the MERS COV and SARS COV using various machine learning algorithms like Decision tree, SVM, and Apriori Algorithm. They analyzed the protein structure as both diseases exhibit similar characteristics and symptoms. The outcome of the result was that both diseases are alike but at the same time, differently based on rules extracted by the Apriori where fewer structures of the protein were similar (Borgelt; 2004).

2.4 Applications of Neural Network

(Narin et al.; 2020) found out that due to the quick spread of the virus around the globe and due to the unavailability of testing kits, it was difficult to control the spread of the virus. Hence, they found out a unique solution where patients were being identified of corona positive using X-ray Images provided by the hospital. They had deployed different Deep Convolutional Neural Network model including ResNet50 , InceptionV3, and Inception ResNetV2). The results were outstanding where RestNet50 outperformed the other two models with an accuracy of 97%. This process turned out to be quicker where results could be found out within 2 hours rather than waiting for 48 hours for testing kit results.

(Jelodar et al.; 2020) performed a sentimental analysis based on COVID-19 data gathered from social media and forums. The natural language process (NLP) was applied to data using the LSTM recurrent neural network (Sainath et al.; 2015). (Oyebode et al.; 2020) shed light on the significance of utilizing popular suppositions and appropriate computational strategies to comprehend issues encompassing COVID-19 and to control related dynamics.

(İsmail Kırbaş, Sözen, Tuncer and Şinasi Kazancıoğlu; 2020) had performed a relative examination and provided estimates of COVID-19 cases in different European nations

with ARIMA, LSTM, and NARNN models. LSTM outperformed the others with MAPE of 0.1.

After analyzing and exploring different research papers it was recommended that this exploration will utilize ARIMA, ARMA forecasting model along with LSTM neural network, and the Prophet model created by the Data Scientists at Facebook for predicting the COVID-19 cases(Taylor and Letham; 2018).

3 Methodology

The procedure approach used for executing this research project is KDD which is an industry-proven technique for data mining developments and study. The Knowledge Discovery in Databases (KDD) specifically for Data Mining is entirely flexible as it gives you the alternative to change the approach by adapting as per projects need while providing a road-map for the project development lifecycle.The entire process flow has been explained in detail of KDD methodology (Refer to Figure 3).

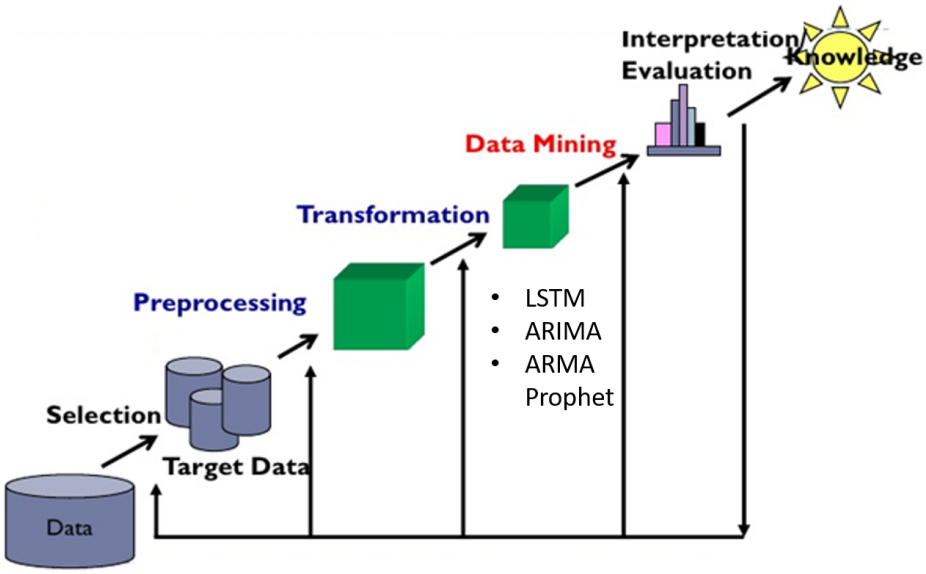


Figure 3: KDD Methodology

3.1 Data Cleaning

It is defined as the removal of noisy and inappropriate data from storage or warehouse. Data cleaning consist of various tasks:

- Removal of the missing values.
- Imputing missing values with either mean, median, or mode based on the data distribution plots.
- Checking for variance in data and normalizing them.

3.2 Data Integration

This step involves collecting data from multiple sources and storing them in one common storage area also known as Data Warehouse. In this project, there is no external database being used, and all the data collected/read are stored in the data frame using pandas library.

3.3 Data Selection

It can be defined as the procedure where information applicable to the project objective is decided and that particular data/information is extracted from the database. In this project, the following steps were performed :

- Unwanted features were eliminated using python scripts.
- Selective data frames with key information were prepared for different time-series models.

3.4 Data Transformation

This is the most crucial step in the project development lifecycle after data cleaning. This step help in refining the model and its outcome. In this project, the following steps were performed :

- Text related information was transformed to extract symptom names.
- Data types were converted as per the requirements of the models.
- The mortality rate was calculated by deriving the formula.
- As data was not stationary, Log transformation was performed on the actual data.

3.5 Data Mining

The four models used for time-series forecasting of the COVID-19 reported cases in this research are:

- ARIMA

  This is a hybrid time series model where AR denotes for AutoRegressive, I denote for Integrated and MA denotes for Moving average. ARIMA works on the same principles of time-series models like ARMA where it takes historical data as input and produces future outcomes by observing the trend. The objective of ARIMA is to define autocorrelations between data points (Zhang; 2003).

  Data is shifted by one position to create one lag and similarly, multiple lags are produced of data that is given to the model for analyzing the trend and predicting the reported cases. (Tandon et al.; 2020).

- ARMA

  ARMA is a combination of two models AutoRegressive(AR) that determines lag and Moving Average(MA) which find the average error between lags. This model takes past data as input in the form of lags i.e. AR component and predicts outcome in the form of series.
  This model typically encompasses model identification, parameter assessment, and analysis (Maleki et al.; 2020).

- Prophet

  This model was designed and developed from scratch by the Data Scientist team at Facebook for forecasting the trends and the nature of the model is additive. So the FB prophet model can predict the daily, weekly, monthly and seasonal trends based on the requirement.

  This model was specifically designed to predict non-linear trends as mentioned above. It comes as a pretrained model where just Input is provided along with the trend configuration which produces the output. Fundamentally, Prophet is a versatile and flexible model that breaks down various time series data and produces a scalable output (Taylor and Letham; 2018).

- LSTM

  This is a special Recurrent Neural Network(RNN) model that specializes in remembering or storing data of longer sequence for predicting accurate trends. LSTM stands for Long Short-Term Memory networks. This implies they are very well fit for recollecting data for extended data when contrasted with RNN. LSTM model falls under the category of Deep learning algorithm where provides optimal results for larger data set with minimal error (Ayyoubzadeh et al.; 2020).

  (Hochreiter and Schmidhuber; 1997) developed the LSTM model which consists of multiple hidden layers having the capability to activate different non-linear functions and includes one input and output layer respectively. As LSTM speciality is to remember long term data, logic gates are implemented where conditions are checked whether this is optimal output or not and trains again based on previous output until it achieves optimal results.

3.6 Pattern Evaluation

Based on the predicted and actual values the model's performance is evaluated using MAPE, MSE, RMSE, and R^2 matrics. For matrics, RMSE, MSE, and MAPE values should be lowest as possible and R^2 value should me maximum ranging between 0 to 1 ,where 1 denotes the highest correlation and variance in data. The results obtained for all four models are discussed in a detailed manner in Section 6 .

3.7 Knowledge representation

Based on the results obtained from data mining models, Visualization tools like Excel or Python scripts are used to plot the graphs.

4 Design Specification

The project was implemented successfully with the following System Architecture (Refer to Figure 4). The proposed framework has been divided into the following five steps:

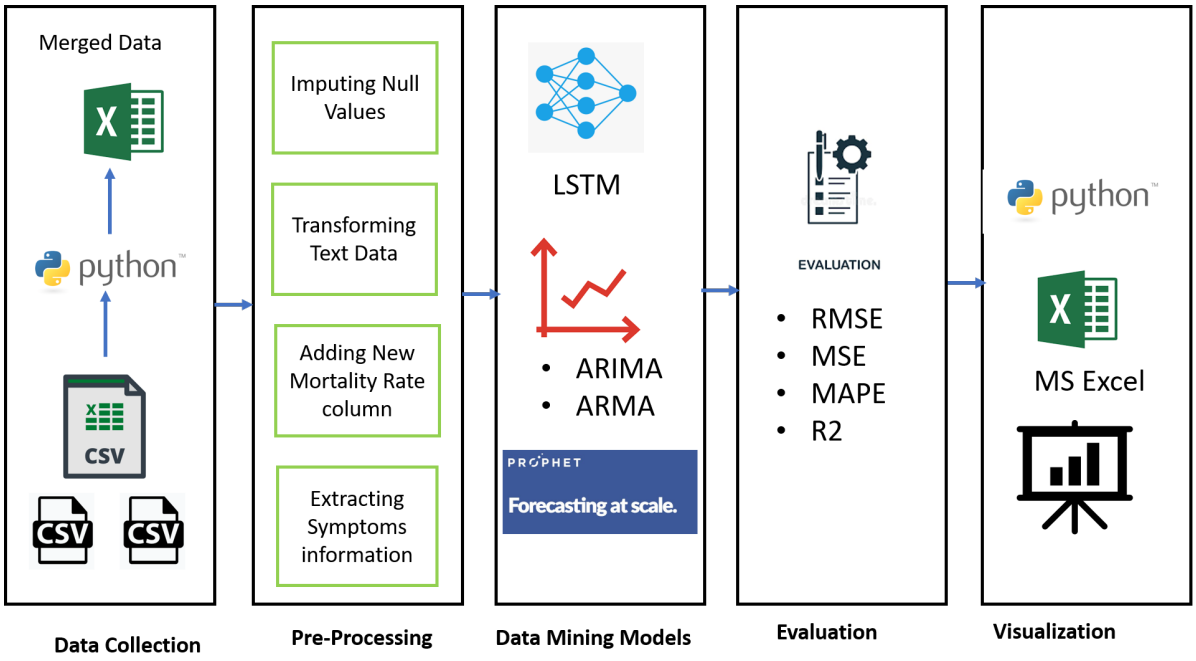


Figure 4: Flow Diagram

1. **Data Collection**
   Data was downloaded from Github repository which is maintained by John Hopkins University (2020) and the dataset is public which gets updated daily . There are multiple CSV files but for this project, we are extracting data from three files namely :

   - COVID19_open_line_list.csv
   - covid_19_india.csv
   - StatewiseTestingDetails.csv

   Collectively data from three CSV files consists of 20,000 records including 40 columns(features). Dataset is loaded in Jupyter Notebook(python)[4] using the pandas[5] library and Data preprocessing is carried out which is explained in the next step.

[4]Jupyter Notebook(IDE) `https://jupyter.org/`
[5]Pandas Library `https://pandas.pydata.org/`

2. **Data Pre-Processing**
Once the dataset is loaded in Jupyter Notebook successfully, Exploratory Data Analysis(EDA) is performed. Before proceeding for EDA, Data needs to be validated by checking for Missing values, Empty Text fields, and Incorrect data type formats.. For this project, the following action items are carried out:

- Dataset consisted of 5% missing values and imputing them with mean, median, or mode would be incorrect as this data represents actual reported cases throughout the globe. Hence, that 5% of data was removed using pandas in python.
- Key Symptoms were extracted from the Symptoms feature where a customized function was created to remove the unnecessary information and retain the symptom part(Refer to Figure 5). .

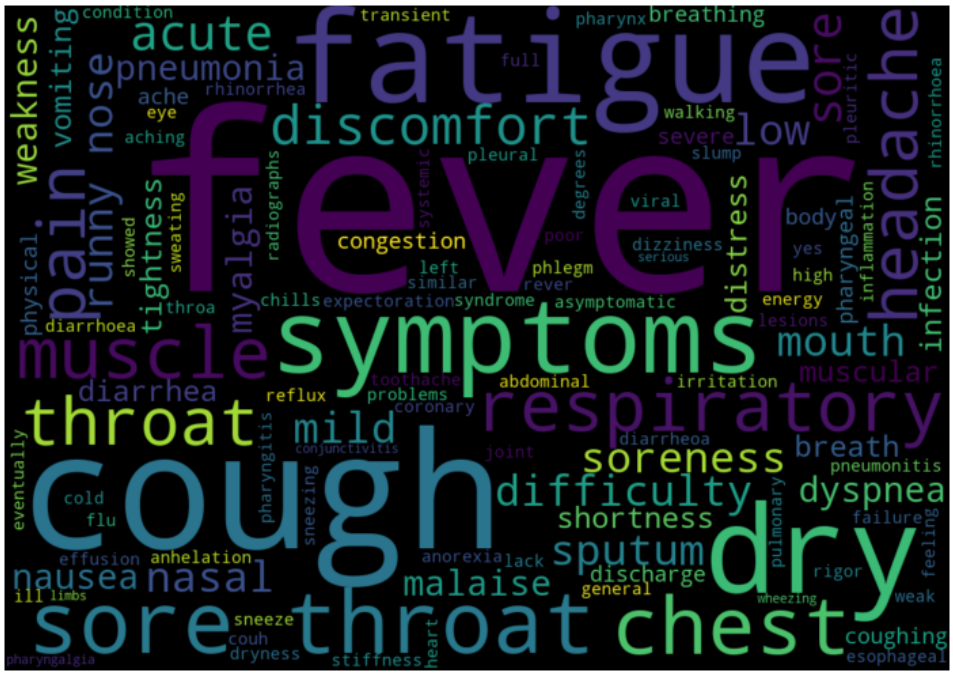


Figure 5: Word Cloud of Key Symptoms observed

- The mortality rate was calculated by deriving the formula and the column was appended to exiting the data frame.

Post data processing completion, EDA was performed by plotting graphs and different plots to analyze the data. Below are the unique results obtained from the EDA:

- After grouping data as per states, It was observed that Maharashtra is the worst affected state in terms of most deaths followed by Tamil Nadu. Hence, COVID-19 sample testing needs to be increased in the following affected states(Refer to Figure 6).
- Based on data, Comparative analysis was done between Maharashtra and Delhi states and both states fall under the category most number of reported cases.

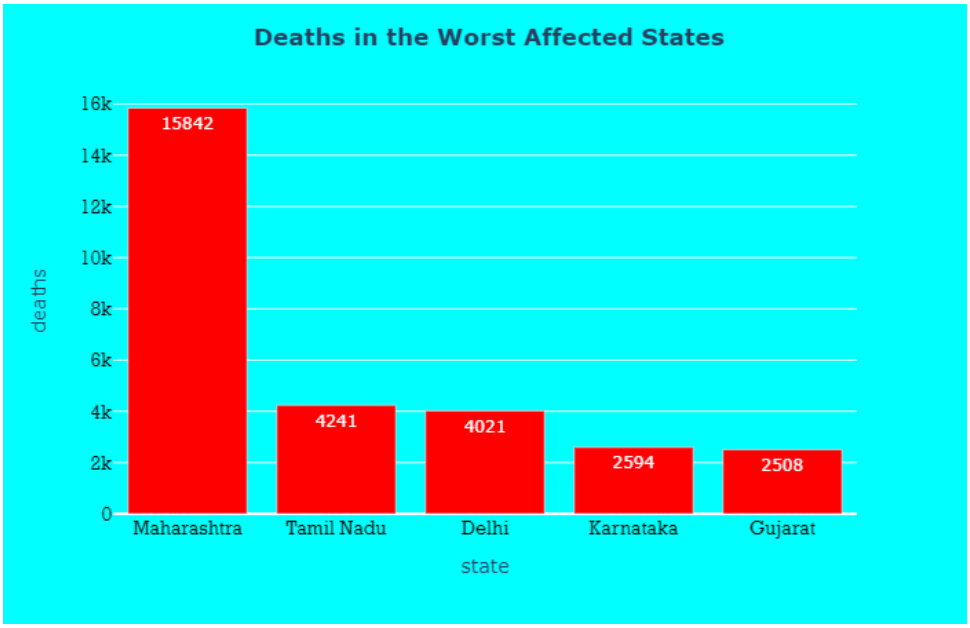


Figure 6: Deaths in the Worst Affected States

It was interesting to observe that the Post lockdown period, Delhi's count of newly reported cases had stabilized while Maharashtra was growing at an exponential rate. Hence, Lockdown wasn't effective in Maharashtra to control the spread of the virus(Refer to Figure 7).

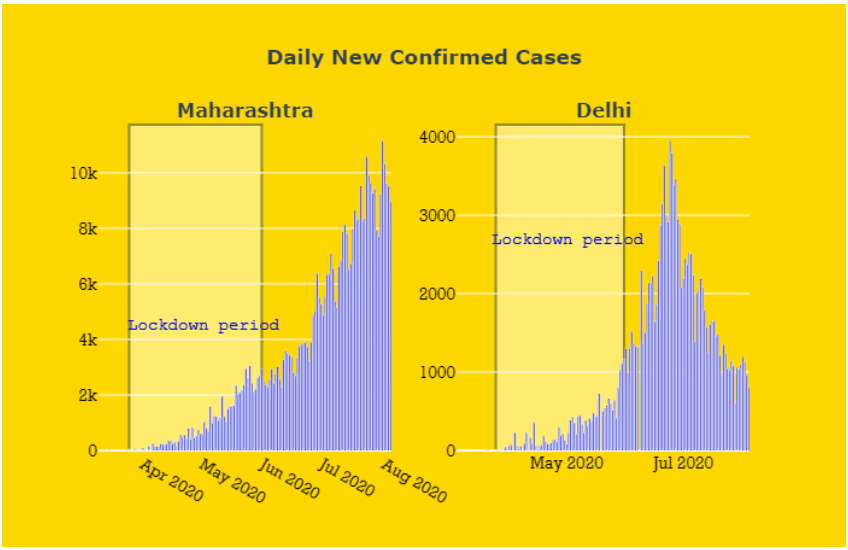


Figure 7: Daily New Reported Confirmed Cases

- The key to stopping the spread of the virus is by performing sample testing at a larger scale for tracking the source of virus spread from the positively identified patient. Tamil Nadu has successfully flattened the curve by performing maximum testing throughout India i.e. 12% followed by Uttar Pradesh(11%) and Maharashtra being the worst affected ranks at number 3 with 10% testing(Refer to Figure 8).

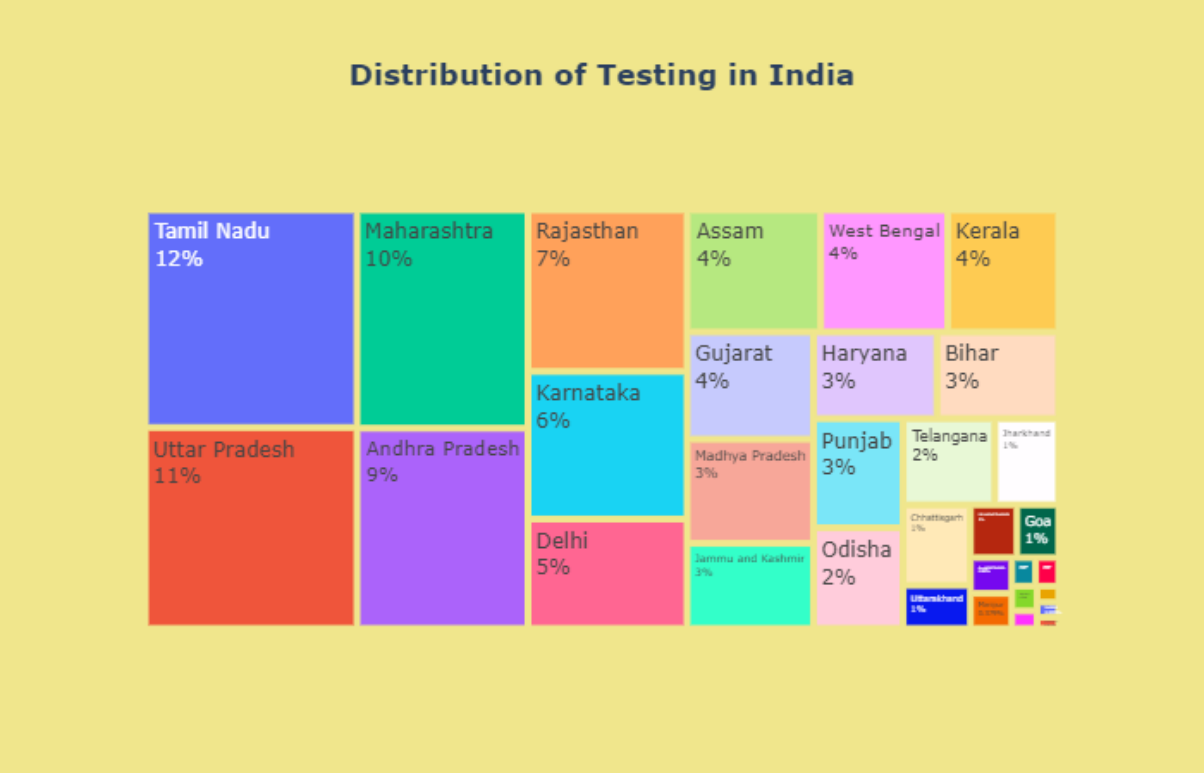


Figure 8: Distribution of Sample Testing in India

3. **Data Mining Models**
   As explained in the previous Section 3.5, Time series models like ARIMA, ARMA, and Prophet were applied along with the LSTM Neural network.

4. **Evaluation**
   Forecasted series data output obtained from the models was evaluated based on RMSE, MSE, MAPE, and R^2 value as explained in Section 6 .

5. **Visualization**
   Graphs and plots were drawn using Excel and Plotly Express library[6] in python.

5 Implementation

Once the data was Preprocessed and transformed, data is being prepared as input for the time series model. In our project, Columns **'Date'** was renamed to **'ds'** and the target variable **i.e. Confirmed Cases** was renamed to **'y'** as per the model configurations. Data models were subjected to input features**(ds,y)** and output as predicted values appended to the data frame for the respective dates.

Summary of Model implemented along with its configuration is shown below :

[6]Plotly Express library `https://plotly.com/python/plotly-express/`

5.1 LSTM Model

- In this project, as data was not normalized, so initially data were scaled using Sklearn Package with function as MeanMaxScaler which transformed the input data ranging between 0 to 1.
- Once data was transformed, now it was time to rename the columns as per the model conventions i.e. 'ds' and 'y' as input features.
- All the necessary Libraries were loaded before training the model. Firstly, batches were prepared using a generator function with input as scaled_train data, target variable, and batch size (Refer to Figure 9).

In [172]:
```
#Importing the required Package
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()

#Scaling the date
scaler.fit(train_data)
scaled_train_data = scaler.transform(train_data)
scaled_test_data = scaler.transform(test_data)

#Creating Dataframe of Scaled Data as Model Input
data = pd.DataFrame(columns = ['ds','y'])
data['ds'] = train_data.index
data['y'] = scaled_train_data
#Creating Model
#from keras.preprocessing.sequence import TimeseriesGenerator
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
import tensorflow as tf

n_input = 12 # Number of Forecasting Days
n_features= 1 #Input Columns

#generating batches of temporal data
generator = TimeseriesGenerator(scaled_train_data, #Scaled Input Train Data
                                scaled_train_data, #Target Data
                                length=n_input, #Length as per input size=12
                                batch_size=1) #Batches generated

#Starting of Model in sequential manner
lstm_model = Sequential()
#First LSTM Layer of size (12,1) with 500 hidden layers
lstm_model.add(LSTM(500, activation='tanh', input_shape=(n_input, n_features)))
#Drop out layer
lstm_model.add(Dropout(0.10))
#Adding Dense Layer
lstm_model.add(Dense(1))
#Compiling the model with Optimiszer as adam
lstm_model.compile(optimizer='adam', loss='mse')
#Displaying Summary of the model
lstm_model.summary()
#Generating the model with 50 Epochs
lstm_model.fit_generator(generator,epochs=50)
```

```
Model: "sequential_18"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_19 (LSTM)               (None, 500)               1004000
_________________________________________________________________
dropout_10 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_13 (Dense)             (None, 1)                 501
=================================================================
Total params: 1,004,501
Trainable params: 1,004,501
Non-trainable params: 0
_________________________________________________________________
```

Figure 9: Code for LSTM Model

- Once the batch is ready, Model was created with the following layers:
  - Single LSTM Layers(500) consisting of hidden nodes with activation function as tanh

  - Single LSTM Layers(500) consisting of hidden nodes with activation function as tanh.
  - Single-layer of Dropout.
  - Single Output Dense layer
  - Model Compiler with Optimizer as 'adam' and loss function as 'mse'

- Based on the generated model, Train data is used as input and test data is for forecasting the values.

5.2 Prophet Model

- Comparatively Fb prophet model is easier to implement as it's a pretrained model where we just need to provide the necessary configurations.
- The model required a simple input data along with the trend as the number of days to forecast values (Refer to Figure 10).

In [130]:
```
#importing fbprophet
from fbprophet import Prophet

#model
m = Prophet()

#fitting the model
m.fit(df3)

#forecasting Future dates
future = m.make_future_dataframe(periods= 12)
future.tail(12)

#rename the column
fb_res.columns = ['ds','FBProphet']
fb_res['FBProphet'] = fb_res['FBProphet'].astype(int)
result = pd.concat([df2,fb_res],axis=1)
del result['deaths']
del result['cured']
del result['ds']
result.FBProphet = result.FBProphet.replace(np.nan, 0)
out = result.tail(12)
result.tail(12)
```

Out[130]:

	date	confirmed	FBProphet
176	2020-07-24	1287945	1148408.0
177	2020-07-25	1336861	1171513.0
178	2020-07-26	1385522	1195112.0
179	2020-07-27	1435453	1218749.0
180	2020-07-28	1483156	1242036.0
181	2020-07-29	1531669	1265485.0
182	2020-07-30	1583792	1288950.0
183	2020-07-31	1638870	1307538.0
184	2020-08-01	1695988	1330644.0
185	2020-08-02	1750723	1354242.0
186	2020-08-03	1803695	1377879.0
187	2020-08-04	1855745	1401167.0

Figure 10: Code for FB Prophet Model

5.3 ARIMA & ARMA Model

- These two-time series models were implemented in MS Excel by using the NumXL Tool Plugin [7].Similar Steps were performed for Both the models.
- As the time series data was not stationary, Target Variable was converted using Log Transformation(Refer to Figure 11)(Kelvin et al.; 2020).

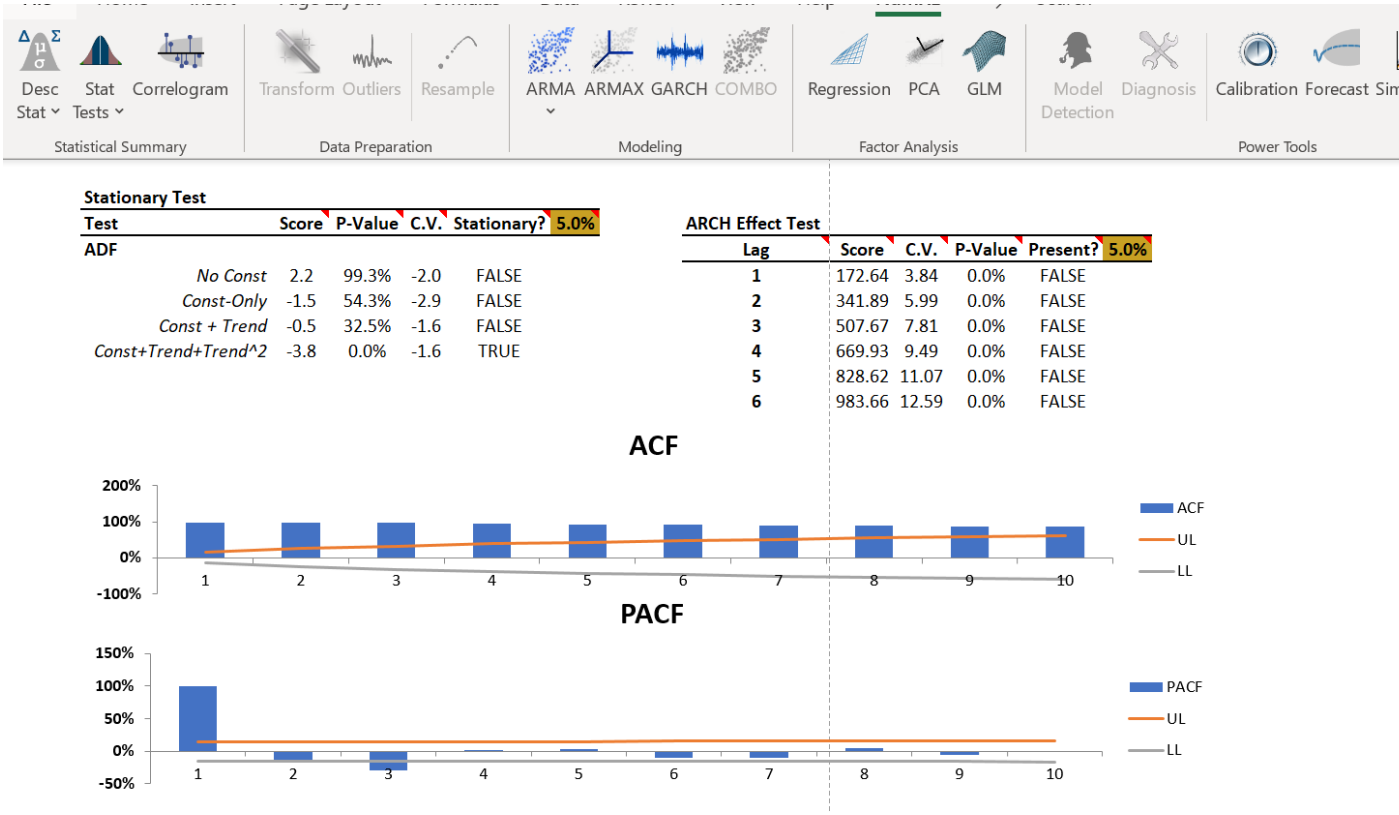
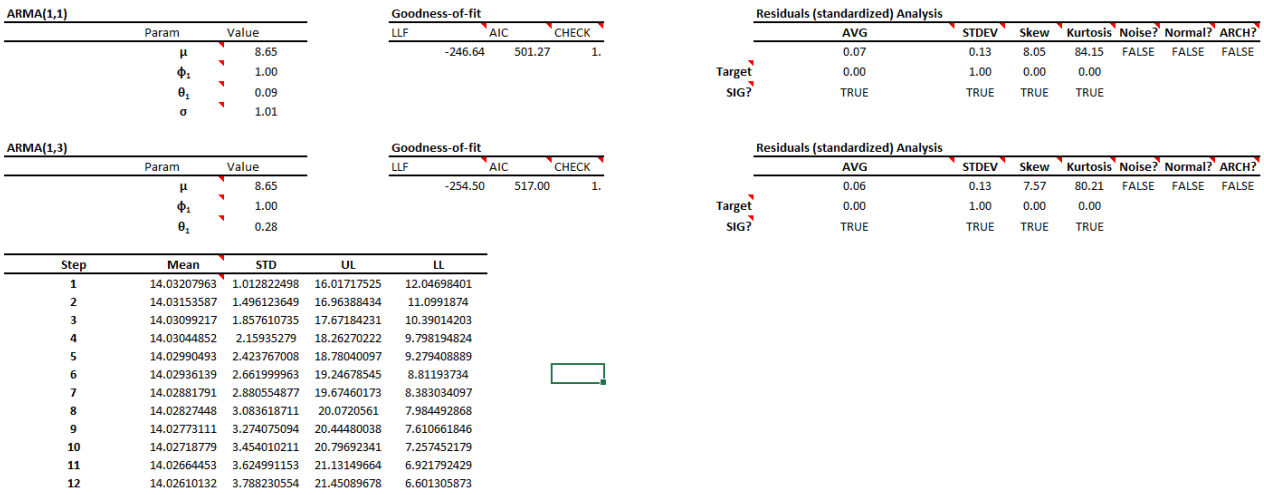


Figure 11: Stationary Test Check for ARIMA & ARMA Model

- By observing the stationary test , it is clear that the model exhibits ARCH effect i.e. Data Trend is exponential and growth is not constant.
- By Plotting the Autocorrelation Plot (ACF), It is clear that the AR Component will have Value as '1' and Partial Correlation Plot(PACF) indicates the MA component values as '1' for the ideal model(Kaushik; 2020).
- Lastly, The model is calibrated based on initial components and coefficients, and the outcome is predicted with the desired steps to forecast(Refer to Figure 12).
- As the result obtained is log value, it is converted to exponential form, and the Model is evaluated.

[7]NumXL Tool Plugin : `https://www.numxl.com/products/numxl`



ARMA(1,1)

Param	Value
μ	8.65
ϕ_1	1.00
θ_1	0.09
σ	1.01

Goodness-of-fit

LLF	AIC	CHECK
-246.64	501.27	1.

Residuals (standardized) Analysis

	AVG	STDEV	Skew	Kurtosis	Noise?	Normal?	ARCH?
	0.07	0.13	8.05	84.15	FALSE	FALSE	FALSE
Target	0.00	1.00	0.00	0.00			
SIG?	TRUE	TRUE	TRUE	TRUE			

ARMA(1,3)

Param	Value
μ	8.65
ϕ_1	1.00
θ_1	0.28

Goodness-of-fit

LLF	AIC	CHECK
-254.50	517.00	1.

Residuals (standardized) Analysis

	AVG	STDEV	Skew	Kurtosis	Noise?	Normal?	ARCH?
	0.06	0.13	7.57	80.21	FALSE	FALSE	FALSE
Target	0.00	1.00	0.00	0.00			
SIG?	TRUE	TRUE	TRUE	TRUE			

Step	Mean	STD	UL	LL
1	14.03207963	1.012822498	16.01717525	12.04698401
2	14.03153587	1.496123649	16.96388434	11.0991874
3	14.03099217	1.857610735	17.67184231	10.39014203
4	14.03044852	2.15935279	18.26270222	9.798194824
5	14.02990493	2.423767008	18.78040097	9.279408889
6	14.02936139	2.661999963	19.24678545	8.81193734
7	14.02881791	2.880554877	19.67460173	8.383034097
8	14.02827448	3.083618711	20.0720561	7.984492868
9	14.02773111	3.274075094	20.44480038	7.610661846
10	14.02718779	3.454010211	20.79692341	7.257452179
11	14.02664453	3.624991153	21.13149664	6.921792429
12	14.02610132	3.788230554	21.45089678	6.601305873

Figure 12: ARMA Model Predictions

6 Evaluation

All four models ARIMA, ARMA, Prophet, and LSTM were successfully deployed and implemented. Evaluation matrix used for the following models are:

- *Root Mean Square Error(RMSE)*
- *Mean Square Error(MSE)*
- *Mean Absolute Percentage Error(MAPE*
- R^2

6.1 Evaluation Matrix for ARMA Model

Based on the predictions of the ARMA model, It is evident that the model is underfitting the values as it comes to the end. Although R^2 is excellent ,the RMSE value 4,71,048.67 is high that shows the error in predicting the outcome(Refer to Table 1)

MAPE	MSE	RMSE	R^2
19.86	140259000000	327680.92	0.9992

Table 1: Evaluation Matrix for ARMA Model

6.1.1 Discussion

The major drawback of the ARMA model is that when data is not stationary, it becomes difficult to analyze the trend as the lag or difference is not constant throughout the dataset.By plotting the graph of Actual vs Predicted values, ARMA is underfitting the model(Refer to Figure 13).

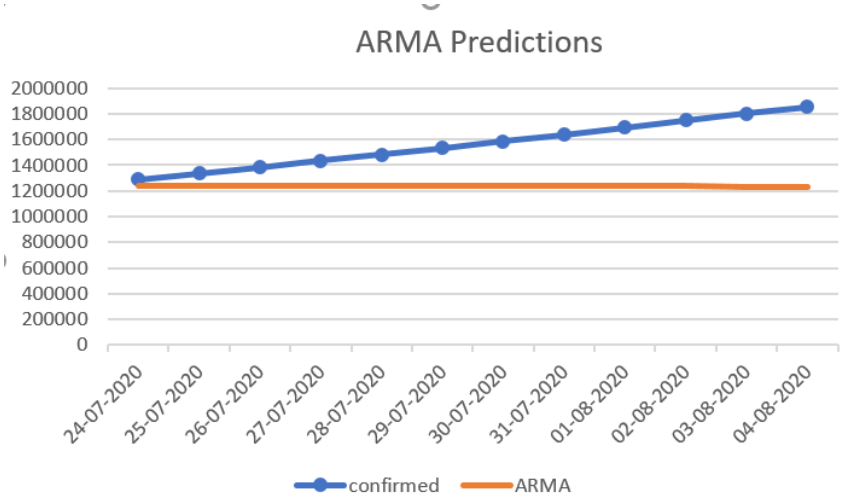


Figure 13: ARMA Model Forecast

6.2 Evaluation Matrix for ARIMA Model

By Observing the Predicted value it is evident that the ARIMA model is not performing well with almost predicted value double the actual values when it comes to the end. Irrespective of R^2 value being excellent the reason being model was performed well for half the values and hence variability was accounted for that. ARIMA exhibits the MAPE of 27.94 % which is highest amongst all the models.(Refer to Table 2)

MAPE	MSE	RMSE	R^2
27.94	346270000000	471048.67	0.9895

Table 2: Evaluation Matrix for ARIMA Model

6.2.1 Discussion

ARIMA is the worst performing model when compared to others. The problem occurred in ARIMA was after the midstage after, date 28-07-2020 the predicted cases were doubled and a similar trend was observed for the remaining values. Hence, RMSE and MSE have the maximum values which make the model as the least trustworthy model.(Refer to Figure 14).

6.3 Evaluation Matrix for LSTM Model

This is the best performing model in terms of R^2 and other factors like RMSE, MSE, and MAPE. LSTM was predetermined to outperform other models based on previous research papers (Arora et al.; 2020) and also, it can store previous results of the model for longer duration and update the values of the model after every iteration.LSTM has the least MAPE of 1.18 %.(Refer to Table 3).

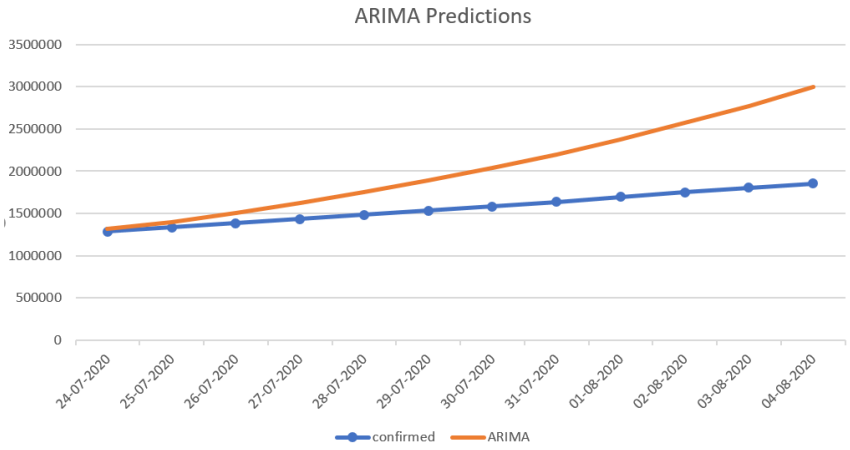


Figure 14: ARIMA Model Forecast

MAPE	MSE	RMSE	R^2
1.18	448705420	19255.19	0.9997

Table 3: Evaluation Matrix for LSTM Model

6.3.1 Discussion

LSTM is a Recurrent Neural Network that consists of multiple hidden layers having the ability to perform independently on different non-linear activation functions and produce an outcome similar to the trend.LSTM is the perfect model that fits the data and predicts output with minimal error.(Refer to Figure 15).

6.4 Evaluation Matrix for Prophet Model

The prophet model is an average performing model and it undermining the data model by predicting lower values than expected. After LSTM, Prophet is the second best performing model.It has a MAPE of 18.07 %.(Refer to Table 4).

MAPE	MSE	RMSE	R^2
18.07	94530537356	290641.33	0.9984

Table 4: Evaluation Matrix for Prophet Model

6.4.1 Discussion

Although the model is not accurate in prediction, results are close to actual values as compared to other models (Refer to Figure 16).

7 Conclusion and Future Work

The Thesis project was successfully implemented with its primary objective to accurately forecast the reported cases of COVID-19. Additional work was done to extract the key

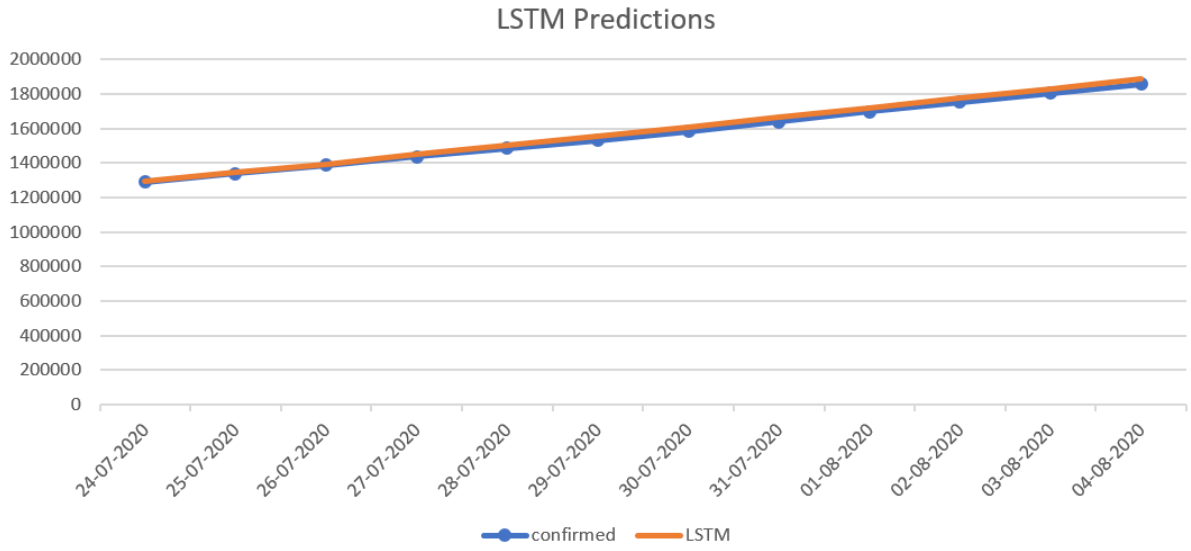


Figure 15: LSTM Model Forecast

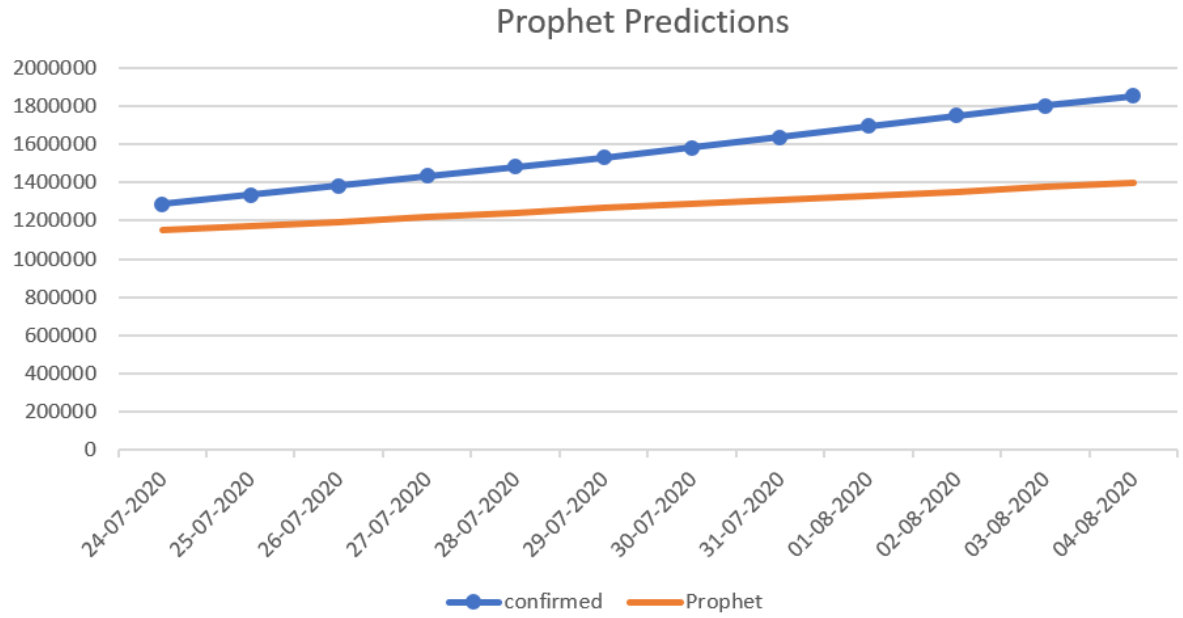


Figure 16: Prophet Model Forecast

symptoms that show early signs of a person being affected by a virus or not. LSTM model outperformed the other models(ARIMA, ARMA, and Prophet) with the least error values for RMSE, MSE, and MAPE along with a maximum R^2 of 0.9997.

Timeseries models are capable of predicting other factors apart from COVID-19 reported cases like predicting the sales of the product. The key takeaway from this research project is that not all datasets stationary in nature and need to be treated differently as per the situation. Future scope for this project is how to efficiently predict the cases without using Neural network models and refining the time series models (ARIMA and ARMA).

References

Andersen, K. G., Rambaut, A., Lipkin, W. I., Holmes, E. C. and Garry, R. F. (2020). The proximal origin of sars-cov-2, *Nature Medicine* **26**: 450–452.

Arora, P., Kumar, H. and Panigrahi, B. K. (2020). Prediction and analysis of covid-19 positive cases using deep learning models: A descriptive case study of india, *Chaos, Solitons & Fractals* **139**: 110017.
URL: *http://www.sciencedirect.com/science/article/pii/S096007792030415X*

Attila, T. (2020). 2019 novel coronavirus disease (covid-19): Paving the road for rapid detection and point-of-care diagnostics, *Micromachines* **11**(3): 300–306.
URL: *https://www.mdpi.com/665214*

Ayyoubzadeh, S. M., Ayyoubzadeh, S. M., Zahedi, H., Ahmadi, M. and R Niakan Kalhori, S. (2020). Predicting covid-19 incidence through analysis of google trends data in iran: Data mining and deep learning pilot study, *JMIR Public Health Surveill* **6**(2): e18828.
URL: *https://doi.org/10.2196/18828*

Borgelt, C. (2004). Recursion pruning for the apriori algorithm, *Computers & Electrical Engineering* **42**: 31–32.

Ceylan, Z. (2020). Estimation of covid-19 prevalence in italy, spain, and france, *Science of The Total Environment* **729**: 138817.
URL: *https://doi.org/10.1016/j.scitotenv.2020.138817*

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory, *Neural Comput.* **9**(8): 1735–1780.
URL: *https://doi.org/10.1162/neco.1997.9.8.1735*

Jang, Seongpil, Lee, Seunghwan, Choi, Seong-Min, Seo, Junwon, Choi, Hunseok and Yoon, Taeseon (2016). Comparison between sars cov and mers cov using apriori algorithm, decision tree, svm, *MATEC Web of Conferences* **49**: 08001.
URL: *https://doi.org/10.1051/matecconf/20164908001*

Jelodar, H., Wang, Y., Orji, R. and Huang, H. (2020). Deep sentiment classification and topic discovery on novel coronavirus or covid-19 online discussions: Nlp using lstm recurrent neural network approach.
URL: *http://dx.doi.org/10.1101/2020.04.22.054973*

Kaushik, K. (2020). *Forecasting and analysis of covid-19 pandemic*, Master's thesis, Dublin, National College of Ireland.
URL: *http://trap.ncirl.ie/4311/*

Kelvin, D., Rubino, S. and Kelvin, N. (2020). Similarity in case fatality rates (cfr) of covid-19/sars-cov-2 in italy and china, *J Infect Dev Ctries* **14**(2): 125–128.
URL: *https://jidc.org/index.php/journal/article/view/32146445*

Maleki, M., Mahmoudi, M. R., Heydari, M. H. and Pho, K.-H. (2020). Modeling and forecasting the spread and death rate of coronavirus (covid-19) in the world using time series models, *Chaos, Solitons & Fractals* **140**: 110151.
URL: *http://www.sciencedirect.com/science/article/pii/S0960077920305476*

Narin, A., Kaya, C. and Pamuk, Z. (2020). Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks.

Nesteruk, I. (2020). Statistics based predictions of coronavirus 2019-ncov spreading in mainland china, *medRxiv* .
URL: *https://www.medrxiv.org/content/early/2020/02/13/2020.02.12.20021931*

Oyebode, O., Ndulue, C., Adib, A., Mulchandani, D., Suruliraj, B., Orji, F. A., Chambers, C. T., Meier, S. and Orji, R. (2020). Health, psychosocial, and social issues emanating from covid-19 pandemic based on social media comments using natural language processing, *ArXiv* **abs/2007.12144**.

Sainath, T. N., Vinyals, O., Senior, A. W. and Sak, H. (2015). Convolutional, long short-term memory, fully connected deep neural networks, *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* pp. 4580–4584.

Tandon, H., Ranjan, P., Chakraborty, T. and Suhag, V. (2020). Coronavirus (covid-19): Arima based time-series analysis to forecast near future.

Taylor, S. J. and Letham, B. (2018). Forecasting at Scale, *The American Statistician* **72**(1): 37–45.
URL: *https://ideas.repec.org/a/taf/amstat/v72y2018i1p37-45.html*

Tárnok, A. (2020). Machine learning, covid-19 (2019-ncov), and multi-omics, *Cytometry Part A* **97**(3): 215–216.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1002/cyto.a.23990*

University, J. H. (2020). Covid-19 dataset.
URL: *https://github.com/CSSEGISandData/COVID-19*

Zhang, G. (2003). Time series forecasting using a hybrid arima and neural network model, *Neurocomputing* **50**: 159 – 175.
URL: *http://www.sciencedirect.com/science/article/pii/S0925231201007020*

Zhao1, C.-H., Wu3, H.-T., Che, H.-B., Song, Y.-N., Zhao, Y.-Z., Li, K.-Y., Xiao2, H.-J. and Zhai, Y.-Z. (2020). Prediction of fatal adverse prognosis in patients with fever-related diseases based on machine learning: A retrospective study, *Chinese Medical Journal* **133**(5): 583–589.
URL: *https://www.mdpi.com/665214*

Zhong, L., Mu, L., Li, J., Wang, J., Yin, Z. and Liu, D. (2020). Early prediction of the 2019 novel coronavirus outbreak in the mainland china based on simple mathematical model, *IEEE Access* **8**: 51761–51769.
Ωİsmail Kırbaş et al.

İsmail Kırbaş, Sözen, A., Tuncer, A. D. and Şinasi Kazancıoğlu, F. (2020). Comparative analysis and forecasting of covid-19 cases in various european countries with arima, narnn and lstm approaches, *Chaos, Solitons & Fractals* **138**: 110015.
URL: *http://www.sciencedirect.com/science/article/pii/S0960077920304136*