Design of an Emergency Ferrying Service for Effective Distribution of Healthcare Equipment Analyzing COVID-19 Cases

by

Md. Shahriar Al Kasib Khan Sourav (160021062)
Md. Robiul Ferdous (160021135)
Md. Riaz Rahman (160021103)

BACHELOR OF SCIENCE
IN
ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Board Bazar, Gazipur-1704, Bangladesh.
February, 2021.

CERTIFICATE OF APPROVAL

The thesis titled **"Design of an Emergency Ferrying Service for Effective Distribution of Healthcare Equipment Analyzing COVID-19 Cases"** accepted as partial fulfillment of the requirement for the degree of BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING of Islamic University of Technology (IUT).

Approved by:

02/04/2021

Dr. Khondokar Habibul Kabir
(Supervisor)
Professor,
Electrical and Electronic Engineering Department,
Islamic University of Technology (IUT), Gazipur.
Date: 02-04-2021

Declaration of Candidate

It is hereby declared that this thesis report is only submitted to The Electrical and Electronic Engineering Department. Any part of it has not been submitted elsewhere for the award of any Degree or Diploma.

Md. Shahriar Al Kasib Khan Sourav
Student Id.: 160021062,
Date: 02-04-021

Ferdous

Md. Robiul Ferdous
Student Id.: 160021135,
Date: 02-04-2021

Riaz

Md. Riaz Rahman
Student Id.: 160021103,
Date: 02-04-2021

Table of Contents

List of Figures

List of Tables

List of Abbreviations

SARS	Severe Acute Respiratory Syndrome
ML	Machine Learning
ANN	Artificial Neural Network
PPE	Personal Protection Equipment
AR	Auto Regression
MA	Moving Average
ARIMA	Auto Regressive Integrated Moving Average
RMSE	Root Mean Squared Error
MSLE	Mean Squared Log Error
LSTM	Long Short Term Memory
STEMI	ST Elevation Myocardial Infarction

Acknowledgment

All praise and gratitude be to Allah, the most beneficent, the most merciful.

In this note of acknowledgment, we would like to thank everyone who has mentored, helped, and assisted us throughout our stay in IUT. We would like to express our special gratitude to our supervisor and mentor Professor Dr. Khondokar Habibul Kabir sir for guiding us throughout the whole research and assisting us in every aspect. We were truly blessed by getting the opportunity to work under his able guidance.

We would also like to thank all the faculties and staff of the Electrical and Electronic Engineering Department for their co-operation throughout the entire process.

Abstract

The recent global epidemic of the novel coronavirus infection 2019 (COVID-19) has created a catastrophic situation all over the world. To monitor and limit the spread of such infections, Machine Learning algorithms are used. In this research study, exponential and time-series Machine Learning algorithms are used to predict the number of infected people of COVID-19 in the upcoming days for a densely populated country like Bangladesh. Besides this, an emergency transportation system, i.e. Emergency Ferry is proposed, which uses the predicted data to supply essential equipment to COVID-19 infected regions. The performance of six different Machine learning algorithms is compared in terms of their predictive accuracy for forecasting COVID-19 future cases of consecutive 33 days. The highest accuracy of 93.1% is achieved using the Holt-Winter model. The calculations for best utilization of the Emergency Ferry are also performed based on the distribution rate and distribution time of essential equipment. The calculations and analysis performed in this study show that combining the predictive analysis of COVID-19 infection along with the appropriate allocation of essential resources using Emergency Ferry can benefit the community to take preventive measures for any sudden spike in the outbreak of COVID-19.

Chapter 1

Introduction

The COVID-19 outbreak is an ongoing global epidemic of the coronavirus infection 2019. This is without a doubt the century's most serious public health threat. Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) is the reason behind this epidemic. The virus has spread throughout the world in a short period [1,2]. The outbreak was initially observed in Wuhan, China, in December 2019 [3,4]. Since then, the virus is spreading throughout the world at an exponential rate. As of 17 June 2020, a total of 8,299,344 people of 213 countries around the world and two international conveyances are already affected by this pandemic [5]. The virus is highly contagious and can easily transmit through human contact. COVID-19 is declared as a pandemic on 11 March 2020 by World Health Organization(WHO) [6].

The healthcare facilities are facing a huge problem in providing treatment to the enormous number of COVID-19 patients [7]. The vaccine to fight COVID-19 is still not available to everyone. Moreover, limited healthcare facilities and mishandling of resource allotment can make the situation more critical. Predicting the spread of the virus using Machine Learning can help in the proper management and allocation of healthcare resources. This can also help in limiting the spread of the virus.

1.1 Problem Statement

COVID-19 has also affected Bangladesh, a south Asian country. Bangladesh is a densely populated developing country. As of 2013, Bangladesh had a population of 156.6 million and it is expected that the population will increase to around 218 million by 2030 [8]. The virus was first detected in Bangladesh on 8 March 2020 with three confirmed cases [9]. As of June 17, 2020, it has reached a total of 98,489 with a total death case of 1,305 [9]. The contagion nature of the virus has drastically increased the time rate of infection due to community transmission. The healthcare facility in Bangladesh is also not sufficient to provide proper medical treatment for this huge population. Table 1.1 summarizes Bangladesh's main social, fiscal, and health indicators. The support of the medical sector has also been hampered due to the increased infection rate of the virus. Since the virus is spreading at a fast pace, it is much needed to know the probable future number of infected cases to cope up with the situation.

Table 1.1: Key demographic, economic and health indicators of Bangladesh.

Subject	Indicators	Value
Population	Total Population in Million	163.1
	Crude Death rate per 1000 Population	6
Health Facilities/ Hospitals	Government Hospitals	607
	Registered Private Hospitals	5054
	Medical Colleges	105
Health Workforce in Public Facilities	Physicians	93,358
	Medical Personnel	74,985
	Technologists	5,184
	Domiciliary Staff	20,103

Sources: World Bank 2018, Bangladesh Bureau of Statistics 2019

1.1.1 Research Gap

Most of the researches done in this field are based on the data of the European countries [10–12]. The population density in those countries is lower, so the spread of COVID-19 is also different from those of densely populated countries like Bangladesh [13, 14]. So, those researches will not provide us the actual prediction of the pandemic for a densely populated region.

Again, the majority of the research works that already have been done only prioritize the prediction cases using various Machine Learning algorithms. They did not use the results of the Machine Learning algorithm to introduce any solution to the problem [15, 16].

1.1.2 Problem Identification

Since COVID-19 spreads with human contact, the chance of getting infected with the virus in a densely populated country is more. So, countries with higher population density like Bangladesh will face more critical conditions than other countries with less population density. Moreover, the healthcare sector of Bangladesh is insufficient to correspond to its large population. So, during a pandemic situation, it is very important for the authority to allocate the necessary resources properly. The problem will arise if the limited resources cannot be managed properly among this huge population.

1.1.3 Research Motivation

The motivation to our research is,

- Sudden outbreak of COVID-19 leading to the socio-economic crisis all over the world.
- Breakdown of the healthcare system due to the huge number of unpredicted patients.
- Mismanagement of resource allocation due to the unpredictable nature of the virus.
- Unable to get essential equipment due to country-wide lockdown.

1.1.4 Research Scope

This Research study focuses mainly on predicting the COVID-19 pandemic situation in Bangladesh. The spread of COVID-19 in a densely populated country will be explored in this study. Moreover, we will also focus on finding a simple solution that will help us in tackling the community spread of the virus in an efficient way.

The effect of temperature, humidity, etc. is not a subject of consideration in this research. So, the results obtained in this research may vary if this parameter changes. Though the research is based on the COVID-19 data in Bangladesh, the solution to limit the spread of the virus is applicable to any region.

1.2 Objectives

The objectives of our research work are,

- To predict the COVID-19 cases using an efficient Machine Learning Model.
- To introduce an emergency transport system, i.e. Emergency Ferry system to supply the necessary equipment to the probable affected area.

1.3 Research Outcome

The expected outcome from this research study is,

- Successful prediction of COVID-19 future cases using an efficient ML algorithm.

- Ensuring the proper management of limited healthcare resources.
- Successful simulation of the Emergency Ferry system to deliver essential resources to the infected regions.

1.4 Novelty of the Research

Though a lot of research is done in predicting the forthcoming cases of COVID-19, most of the time the impact of population density is ignored. The spreading pattern of COVID-19 in the densely populated country is different. We will run a predictive analysis to determine the COVID-19 future cases in countries with high population density. In this research study, we are also going to propose a very unique model that uses ML prediction to supply medical resources by the emergency transportation system, i.e. Emergency Ferry. The Emergency Ferry system will ensure the proper allocation of essential resources in the COVID-19 infected region and help the community to get prepared in advance.

1.5 Overview of Methodology

The whole work is divided into two parts. In the first part, there is a brief discussion about different algorithms and their implementation on the data for Bangladesh to run a predictive analysis of COVID-19 future cases. Here we have also discussed some evaluation metrics which are usually used to evaluate the performance of an algorithm. Later, the results obtained from the prediction were described and used to compare the algorithms in terms of prediction accuracy. The predicted value is then used in the second part of the research work. In this part, the model for the Emergency Ferry is introduced. There we have discussed how different calculations were performed on the data that have been collected from the predictive simulation of the ML models. At the same time, the insights of the results are shown using different plots.

1.6 Organization of the Thesis

This paper has been organized in the following sections. In Section 2, the proposed model of the research is discussed in detail. In Section 3, the methodology to achieve the objective and a brief discussion of various ML algorithms is reviewed. In Sections 4, the experimental results are described in detail. In Section 5, the conclusion and the future works of this research are mentioned briefly.

1.7 Literature Review

The main subject of this research is to determine the future confirmed cases of COVID-19 using machine learning algorithms. Later using this prediction, we will propose a solution to help the medical system handle COVID-19 patients efficiently. Due to the growing number of cases identified in recent months, as well as the resulting burden on the government and healthcare staff, certain prediction procedure are urgently needed to forecast the intensity and number of incidents in the upcoming days. Various research works have been done on the predictive analysis of COVID-19 using ML algorithms.

One of the papers used ANN to predict the future confirmed cases [17]. Though ANN is a very handy algorithm in the case of deep learning and machine learning, to get a good accuracy we should provide a huge amount of data. But COVID-19 has no historical global data and also the data that is available now is only for one year. So, it is not a wise solution to use ANN in the scenario. It may give us a better accuracy for the short term but it will fail for any long-term prediction. Another research paper has used different regression algorithms like a ridge, lasso, elastic net, etc. along with the SMSI lag method to predict the confirmed cases [18]. But as it is seen from different graphs of the COVID-19 outbreak that it follows a clear exponential trend. So, using a regression algorithm won't do better with the increase of days. Because the nature of the virus is that it spreads from person to person. So, the rate will increase with the number of people get affected. So, there is a gap in choosing perfect algorithms to predict this scenario. In another paper, the researchers had only tried a few of the classical regression and classification algorithms [19]. As a result, there are many scopes of improvement in these papers. So, in our research, we have tried to use various types of algorithms starting from regression, time series, exponential models, etc. One of the papers has also worked on predicting COVID-19 cases using deep learning where they have used chest x-ray to predict whether someone is affected with COVID-19 or not [20]. In one paper, they used the Dynamic Time Warping process to predict the upcoming COVID-19 cases [21]. This is a time-series prediction model. But forecasting using the Exponential growth model is not mentioned in this paper. In a research, the segmented Poisson algorithm was included with the power law and the exponential law to study the COVID-19 epidemic in some prominent western countries [22]. In one study the uses a data driven long-short term memory (LSTM) estimation approach [23]. The Kaplan-Meier curve was used to gather data to extract independent bio-markers for improved clinical outcomes [24]. In another study, an objective model was proposed for the efficient prediction of COVID-19 prolongation using a simple and robust process [25]. A long-short term memory based neural network for risk prediction was developed in a research [26]. An optimized Bayesian-based system was used to predict country-specific viral networks. A research was also done based on the data-set of patients from STEMI. For this different algorithms were used for predictions, like, Logistic Regression, Neural Networks, different versions of Bayesian Network Classifiers, and the Decision Tree [27]. In another research ARIMA model was used to determine the spread of COVID-19 in the 15 major European countries [28]. But the spread can be different in the South Asian countries due to its climate, demography and population density [29, 30].

Chapter 2

Proposed Model

The aim of our research study is to determine the COVID-19 cases for a densely populated country. For this purpose, we chose Bangladesh as our research scenario. The data is collected from the Institute of Epidemiology Disease Control And Research (IEDCR) database [9].

To achieve the objective, different Machine Learning algorithms were used to forecast the total confirmed cases of COVID-19 in Bangladesh. We only chose the Time-Series Forecasting Models and Exponential Models for the comparison purpose. A time series is a string of metrics that are chosen over a definite interval of time. Time-series analysis comprises methods for processing time series data to derive useful information and other features of the data. The use of a method to forecast future values dependent on the earlier observed values is known as time-series forecasting. Based on the time considered, it can be of various types, i.e. daily, weekly, monthly, quarterly, yearly, etc. These past data can be used to forecast upcoming values.

Since in this thesis we predicted the number of COVID-19 future infection based on the past infected patient number, so the data used in this research fits the exact requirement that we need for a Time Series Forecasting analysis. For this reason, only the Time Series Forecasting Models and Exponential Models are used in this research study.

In the next part, we designed an emergency transportation system, i.e. Emergency Ferry for efficient allocation of essential resources. First, we divided the whole country into several clusters based on the number of predictions of COVID-19 infections as shown in fig 2.1. The Emergency Ferry should travel to each cluster based on the demand for the essential equipment. To divide the demand among the clusters we have used a weighted method. At first, we find the ratio of the infected cases with respect to total infected cases in a day in each of the clusters, and then according to that ratio the priority to receive necessary equipment for each cluster is determined. We run a simulation of 100 days with different numbers of the emergency ferry. For a different number of the emergency ferry, it is found that the delivery rate, time to deliver,

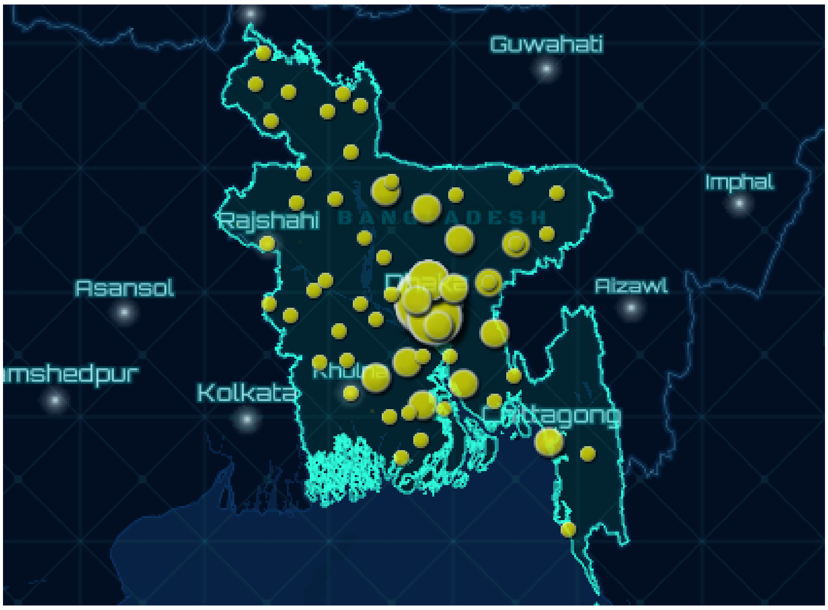

Figure 2.1: Proposed Simulation scenario with Cluster Location.

distance traveled, cluster visit rate is also different. So from this, we must decide on, how to operate the Emergency Ferry for maximum efficiency. Also what might be the approximate number of the emergency ferry to handle the situation perfectly.

Chapter 3

Methodology

The whole research is divided into two sections. In the first section, a comparison between various machine learning Algorithms is made. For the comparison purpose first of all the data-set is collected from IEDCR official sources. This data is used to train the machine learning algorithms and the result is obtained after simulation. By comparing the results, the best one is chosen based on some evaluation metrics.

In the second section, we proposed the Theory for the Emergency Ferry system. The design and simulation of the Emergency ferry system were performed using NetLogo software. For this simulation, the prediction obtained from the first section is used as the input. The simulation was performed considering various scenarios which are discussed in detail in the Result section.

3.1 The procedure of Machine Learning Forecasting

For the simulation purpose, the whole data-set is divided into two parts. One of which is for training our model and another for testing the accuracy of our model. We have considered a total of 100 days for the experiment. Out of which we already had the data of 67 days. We used the data of these 67 days for training and testing. Then we have predicted the outcome for the next 33 days using the trained model. We have performed the prediction for a total of 33 days because when we tried to predict for a longer time horizon the RMSE increases a lot. This leads to a decision that it is better to predict for a shorter time horizon. To get the prediction for a longer time horizon we should wait and collect some more data. Different algorithms have performed differently in the data-set. The results are compared based on some Evaluation Metrics. The outcome and comparison of the result obtained from the different algorithms are described in the Result section.

3.2 Overview of Machine Learning Algorithms

We have used six different ML Models to predict the COVID-19 future cases. Later the obtained are compared to find which model performs best under the given circumstances. A brief description of different Machine Learning models used in this research study is described in this section.

3.2.1 ARIMA Model

An Auto-Regressive Integrated Moving Average (ARIMA) model is a combination of Autoregressive (AR) and moving average (MA) lags that capture the auto-correlation within the time series [31]. This is a univariate time series model where only the past values of time series are used to predict the future values. Mainly 3 terms are used to characterize the ARIMA model, they are the non-negative integers p, q, and d. Here p refers to the order of the AR term, q refers to the order of the MA term and d is the differentiating number that is needed to make the time series stationary. The ARIMA forecasting equation is:

$$Y_t = a + Y_p + Y_q \tag{3.1}$$

Here Y_t is the predicted term, a is the intercept term which is a constant, Y_p is the linear combination Lags of Y up to p terms and Y_q is the linear combination of Lagged forecast up to q terms.

In a pandemic situation, the infection rate is often correlated with time. Therefore, such relations in which we try to predict the total number of confirmed infection cases can be predicted based on the ARIMA model.

3.2.2 Exponential Model

The exponential model is also another type of univariate time series growth model. This model is usually used where any exponential curve is seen [32]. The prediction that is given using exponential smoothing methods is the weighted averages of past observations. The weight associated with recent observation is higher. As the observations get older the weights associated start decaying exponentially. The equation for the exponential model is:

$$y(x) = N_0(1 + p)^{(x - X_0)} \tag{3.2}$$

Here y(x) is the total number of cases after a certain time, P is the rate of exponential growth, X_0 is the day of the the first training, x is the particular day on which we are going to predict.

COVID-19 seems to spread at an exponential rate. So, this algorithm is also suitable to predict future cases of COVID-19.

3.2.3 Bertalanffy Model

Bertalanffy model is also a time series model introduced by Ludwig von Bertalanffy in 1938. This is a growth model that is usually used to predict the growth of the population of any species [33]. The effect of COVID-19 is worsened due to community transmission. So, we have tried out this algorithm to predict COVID-19 cases. The equation or Bertalanffy model is:

$$y(x) = c * (1 - exp^{-a(x-X_0)})^b \tag{3.3}$$

Here y(x) is the total number of cases, c is the number of cases where the graph will go for asymptote, a and b are the coefficients for fitting the algorithm, X_0 is the day of the first training, x is the particular day in which we are going to predict.

3.2.4 Logistic Model

The logistic model is usually used in sustainable growth analysis. If we consider any epidemiology as sustainable growth we can use this algorithm to predict its future behavior. It is used to forecast the probability of the occurrence of any disease using the factors of the disease [34, 35]. The logistic model tries to make a linear trend from any exponential growth case, which makes it possible to ease the prediction. The equation for this model is:

$$y(x) = \frac{c}{1 + exp^{-\frac{x-b}{a}}} \tag{3.4}$$

Here y(x) is the total number of cases, c is the number of cases where the graph will go for asymptote, a and b are the coefficients for fitting the algorithm, x is the particular day in which we are going to predict.

COVID-19 seems to have exponential growth, so we also have tried this algorithm to predict the future number. We have used this algorithm to get the probabilistic growth of the pandemic.

3.2.5 Holt Model

The Holt model is a forecasting model that follows a linear trend. This model may also be termed a linear exponential smoothing model. This model is used for time series prediction in which there is a clear trend but no seasonality. One advantage of this model is, no minimum number of observations is required to be made before it begins to generate results. But to get a good result, the number of samples taken should be as high as possible. The general form of the linear exponential equation used for the Holt Model is:

$$s_t = s_{t-1} + \alpha(x_t - s_{t-1}). \tag{3.5}$$

Here $alpha$ is the smoothing factor. s_t is a simple weighted average of the current observation x_t and the previous smoothed statistic s_{t-1}.

A linear trend is there in the spreading of COVID-19 that is predictable using this model. So this model has been used in this research study to predict COVID-19 future cases.

3.2.6 Holt-Winter Model

Holt-winter model is also a time series model that is used to predict a sequence of values over a time period [36]. This is the most popular forecasting method for Time-series data. The model takes care of all the three aspects of time series behavior: value, trend, and seasonality.

Holt-Winter model is also called the Triple Exponential Smoothing model since it uses three types of exponential aspects (value, trend, and seasonality) for predicting purpose. This model uses the combined effects of these three influences to predict the current or future values.

The growth of COVID-19 is related to time. So this algorithm may give us a good result in predicting future cases of COVID-19. That's why we have used this algorithm in our work too.

3.3 Evaluation Metrics

We have used different evaluation metrics to evaluate the performance of different algorithms used to do forecasting. The evaluation metrics are important because they tell us how accurate the model is. Using a misleading Model is a bad idea since accuracy is very important in this Project. A misleading model will give a wrong prediction about the number of COVID-19 infected patients. This can confuse and create difficulty in taking appropriate measures to tackle the pandemic based on the predictions. So it is important to quantify the accuracy of the Model using the evaluation metrics. The evaluation metrics are described below:

3.3.1 R2 Score

R2 score is the proportion of the variance in the dependent variable that is predictable from the independent variable(s) [37]. This is a measure of accuracy. The value can vary from 0 to 1. The more the R2 score, the better would be the performance. R2 score close to 1 will be better for any prediction. The formula to measure R2 Score is:

$$R2score = 1 - \frac{\sum(y - y_{hat})^2}{\sum(y - y_{avg})^2}. \tag{3.6}$$

Here y is the actual value, y_{hat} is the predicted value, y_{avg} is the average of the actual values. By these metrics, we can explain how accurate the prediction is.

3.3.2 RMSE Score

RMSE stands for Root Mean Squared Error. Root Mean squared error (RMSE) is root of the average of the square of the errors. This metric helps to know the error of the prediction. The larger the number the larger the error. Error in this case means the difference between the observed values and the predicted ones . So, for the model to be accurate the RMSE value should be as low as possible. In order to keep the RMSE value low, we need to adjust various parameters in the model accordingly. RMSE is calculated as:

$$RMSE = \sqrt{\frac{(y - y_{hat})^2}{N}}. \tag{3.7}$$

Here y is the actual value, y_{hat} is the predicted value, N is the number of data. RMSE is useful to know how the algorithm is performing in terms of error. The error should be less to choose any model.

3.3.3 MSLE Score

MSLE is termed as Mean Squared Log Error. It is used to measure the relative error between the actual and predictive value. Since Logarithm ia used here, the MSLE only care about the relative difference between the true and the predicted value. So MSLE large errors between the true and predicted value same as that of the small errors, which is a very good way to measure the performance of different algorithms. The formula to measure MSLE is:

$$MSLE = \frac{\sum((log(y+1) - log(y_{hat}+1))^2)}{N}. \tag{3.8}$$

Here y is the actual value, y_{hat} is the predicted value, N is the number of data.

3.4 Overview of Simulation Scenario for Emergency Ferry System

The prediction from the machine learning part is used to divide the demand of equipment among different clusters. As of April 9, the number of test centers in Bangladesh was 18. So, we have placed a total of 18 clusters in our simulation environment in which each cluster represents one test center [9]. We run a simulation of 100 days with different numbers of the emergency ferry from 1 to 18, and found that the delivery rate, time to deliver, distance traveled, cluster visit rate is also different. From this we can determine the cluster visit priority of the Emergency Ferry.

To design the Emergency ferry system a simulation was performed using NetLogo, which is a programmable modeling environment. The number of the Emergency ferry is manually changed in the simulation. We have considered different features to complete the simulations that include the delivered amount of equipment, total distance traveled by a single vehicle, and in which cluster an emergency ferry is delivering. We have simulated for a total of 100 iterations where each iteration represents one day. Two scenarios are taken into the consideration to complete the overall simulation. They are,

- Demand Priority
- Time Priority

3.4.1 Demand Priority

In this scenario, the emergency ferry moves on the basis of demand priority. The emergency ferry first move to those clusters which have more demand for essential equipment. The demand priority can be chosen based on the prediction of the COVID-19 infected cases, as the demand is proportional to the infected cases. We have measured the demand rate by the number of affected people in each region.

3.4.2 Time priority

In this scenario, the message ferry delivers the necessary equipment to the clusters by prioritizing the total time required to deliver the equipment. So the message ferries will move in a sequence from one cluster to another cluster in such a way that the time taken to deliver the equipment is minimum. We have used shortest path algorithm of traveling salesman problem to find a sequence of clusters which will take less time to deliver the necessary equipment to all the clusters [38].

Chapter 4

Result and Discussion

In this section, we discussed the result obtained from various ML models and also the simulation outcome of the Emergency Ferry system. All the results are shown graphically. Then a comparison is made between the outcome of the ML models and the best one is identified based on both a graphical representation and Evaluation metrics.

4.1 Comparison of Result for various Machine Learning Models

For comparing the Predictive analysis of various ML Algorithms, we first used the graphical method to find out the difference between the actual case and predicted case. Then we also compared the evaluation metrics of various ML Algorithms, based on which the best Algorithm for our applied condition has been chosen.

4.1.1 Graphical Outcome of Various ML Algorithms

The graphs showing the difference between the Actual case and the Predicted case for various algorithms are plotted here.

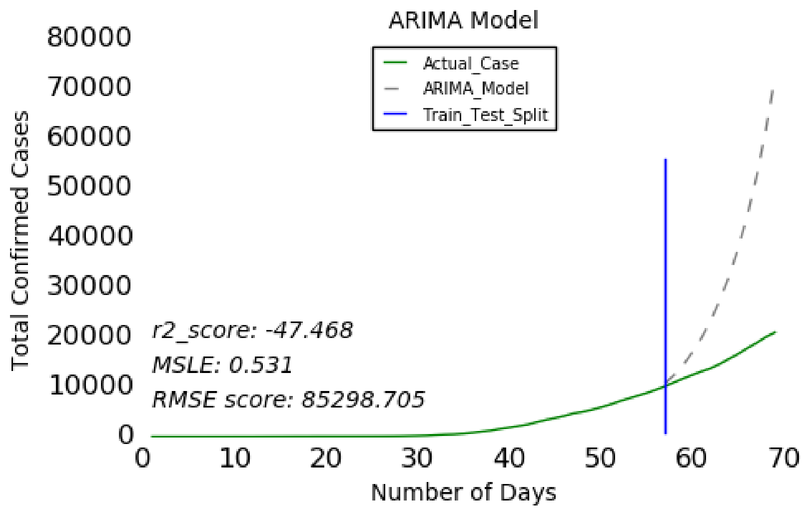

Figure 4.1: Graph of ARIMA Model predicting COVID-19 confirmed cases.

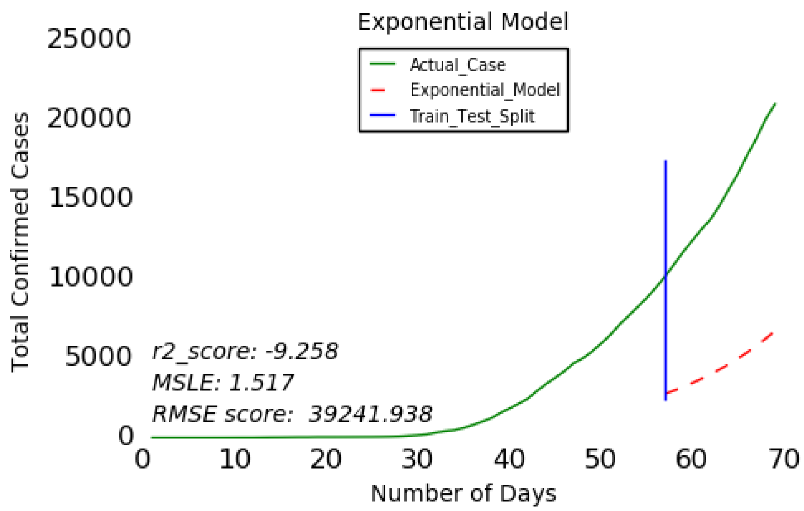

Figure 4.2: Graph of Exponential Model predicting COVID-19 confirmed cases.

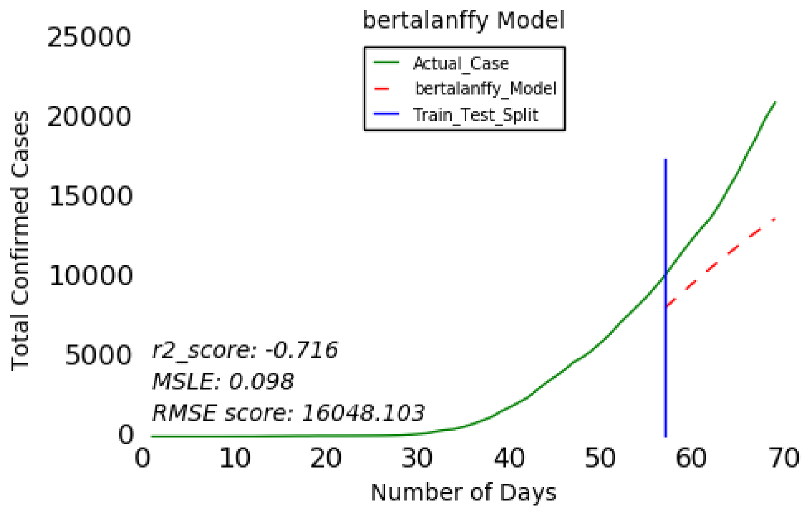

Figure 4.3: Graph of Bertalanffy Model predicting COVID-19 confirmed cases.

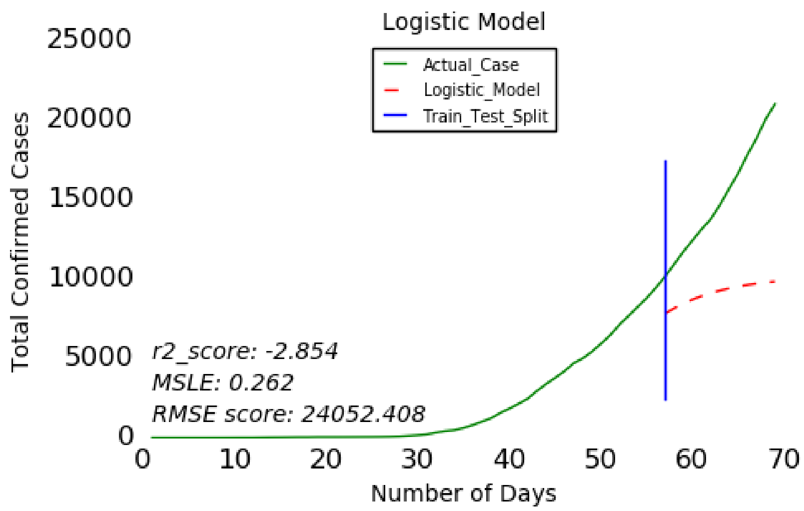

Figure 4.4: Graph of Logistic Model predicting COVID-19 confirmed cases.

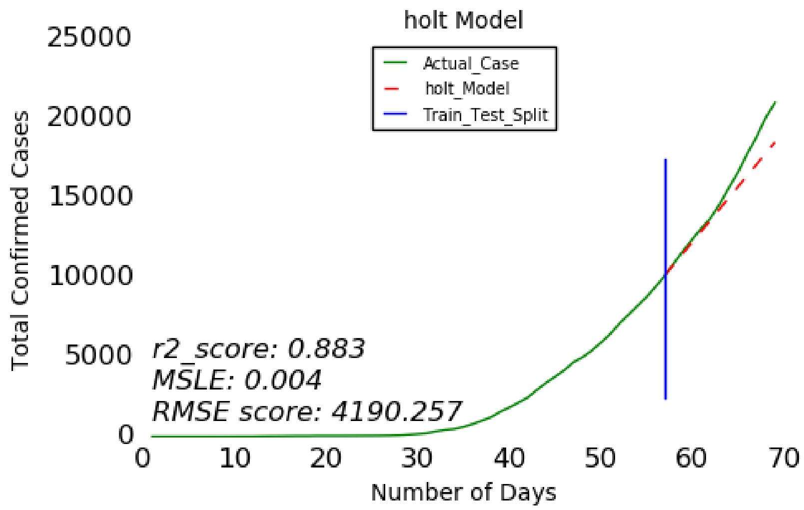

Figure 4.5: Graph of Holt Model predicting COVID-19 confirmed cases.

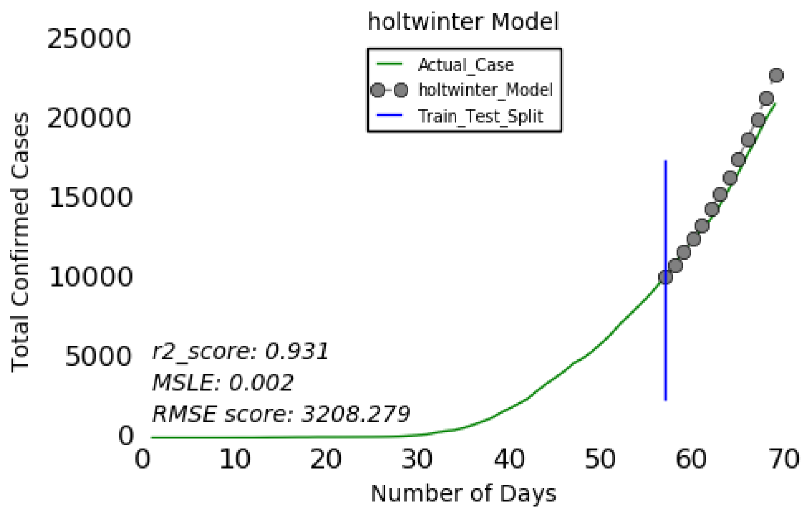

Figure 4.6: Graph of Holt-Winter Model predicting COVID-19 confirmed cases.

From the graphs, it is observed that the Holt-Winter model has performed better than any other algorithms, since the difference between the actual case and predicted case in this case is the lowest.

4.1.2 Comparison of Result based on Evaluation Metrics

In the earlier section, the importance of evaluation metrics in determining the accuracy of the ML models have been discussed. Here the evaluation metrics obtained for different algorithms obtained are presented in a tabular form for easy comparison. From table 4.1 we can see the numerical comparison of different algorithms based on the evaluation metrics.

Table 4.1: Comparison of different algorithms with respect to evaluation metrics

ALGORITHM	R2 SCORE	MSLE	RMSE
ARIMA	-47.468	0.531	85298.705
Exponential	-9.258	1.517	39241.938
Bertalanffy	-19.861	92.212	55960.480
Logistic	-2.854	0.262	24052.408
Holt	0.883	0.004	4190.257
Holt-Winter	0.931	0.002	3208.279

From the table, it is observed that the Holt-Winter model has performed better than any other algorithms concerning all the evaluation metrics. In the case of accuracy measured by R2 score, the accuracy is 0.931 which is higher than the other algorithms used in the simulation. The MSLE score is also the lowest compared to other algorithms. This is a reasonable result to predict any epidemic situation.

4.2 Calculation and Result of Emergency Ferry System

In this part, we have discussed the outcome of the Emergency Ferry System simulation. We showed the result using line plot and bar plot under various circumstances.

4.2.1 Percentage Average Delivery Rate: Line Plot

For a particular number of the emergency ferry, we have calculated the sum of delivered equipment in each day for a total of 100 days . Similarly, the total number of clusters visited in a single day is calculated for a different number of Emergency ferry. Then we have also calculated the average distance traveled by a particular number of the Emergency ferry on a particular day. From this, we have got the average time considering the speed of each Emergency ferry as 50 km per hour.

After this calculation, the delivery rate and the cluster visit rate per day is calculated using the following formula:

- **Delivery Rate on a particular day** = (Total delivered per day / Total Demand per day) * 100
- **Cluster visit Rate on a particular day** = (Visited number of cluster per day/ Total demanded cluster per day) * 100

Here, the total demanded cluster means those clusters which need the emergency equipment in that particular day. These calculations have been performed by varying the number of the Emergency ferry from 1 to 18. Thus Delivery rate, cluster visit rate, the average distance traveled, the average time taken on a particular day is calculated for a particular number of the Emergency ferry.

After the simulation is complete 4 different line plot is created from these data.

1. **Day vs Percentage Delivery Rate:** This plot indicates the variation of delivery rate for a various number of the Emergency ferry with the increase of days. In fig 4.7 and fig 4.8, we can see the plots for both of our scenarios. We can see for both the cases the delivery rate decreases as the day increases. This happens because as the day increases the number of patients increase, so the demand also increases. It is also clearly visible that with the increase of vehicles the delivery rate increases. So in this case with a total vehicle of 18, the delivery rate is the highest. The graph is plotted for a different number of the emergency ferry. In the graph, we can see different markings for different numbers of emergency ferries from 1 to 18.

The Day vs Percentage Delivery Rate for Demand Priority and Time Priority are plotted below.

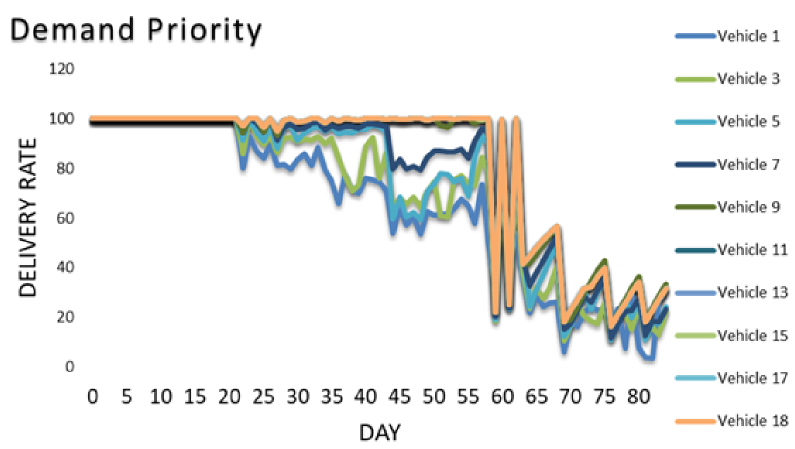

Figure 4.7: Demand Priority: Day vs Percentage Delivery Rate.

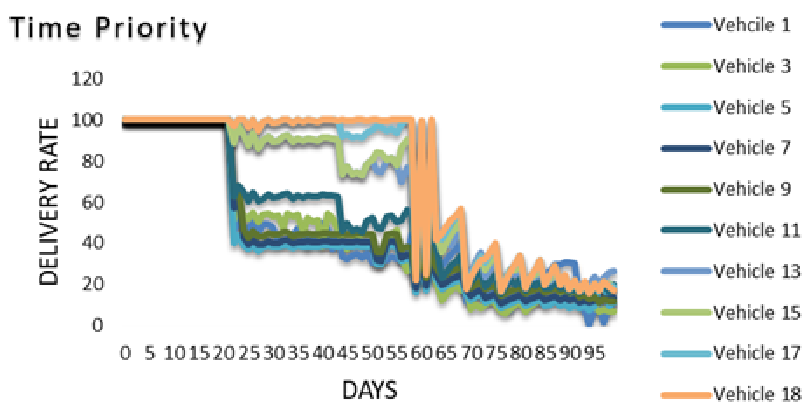

Figure 4.8: Time Priority: Day vs Percentage Delivery Rate.

2. **Day vs Percentage Cluster visited rate:** This plot indicates the variation of cluster visited rate with the increase of days for various number of the Emergency ferry. From fig 4.9 and fig 4.10, we can get an idea that as the day increases the dedicated emergency ferry cannot go to all the clusters to deliver the demand equipment. Because as the day increases the demand of a particular cluster increases. So an emergency ferry needs to move back and forth from a particular cluster to the distribution centre to collect the equipment and then to deliver it to the clusters. But the emergency ferries can't work for the whole day. We take 8 hours for standard to give a boundary that how much time the emergency ferry is available. So as the demand increases the Emergency ferries can

not move to the allocated clusters within that 8 hours boundary. So some of the clusters can not get the necessary equipment. As a result, the cluster visited rate also decrease. But when we increase the number of vehicles the cluster visited rate also increases. So with 18 vehicles, the cluster visited rate is the highest.

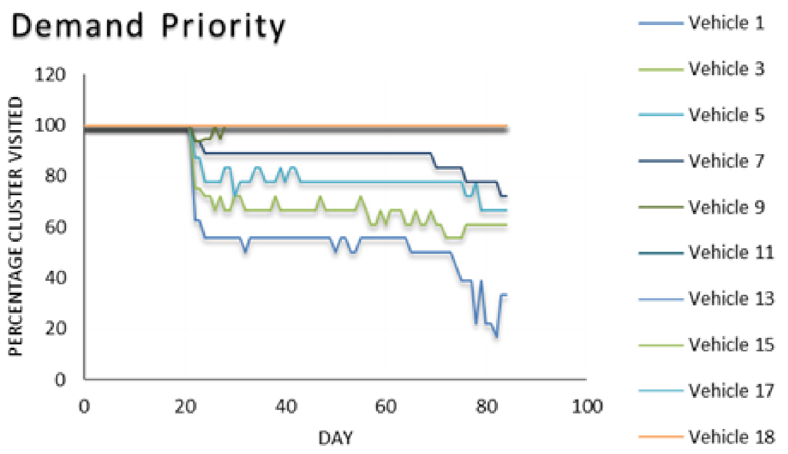

Figure 4.9: Demand Priority: Day vs Percentage Cluster Visit Rate.

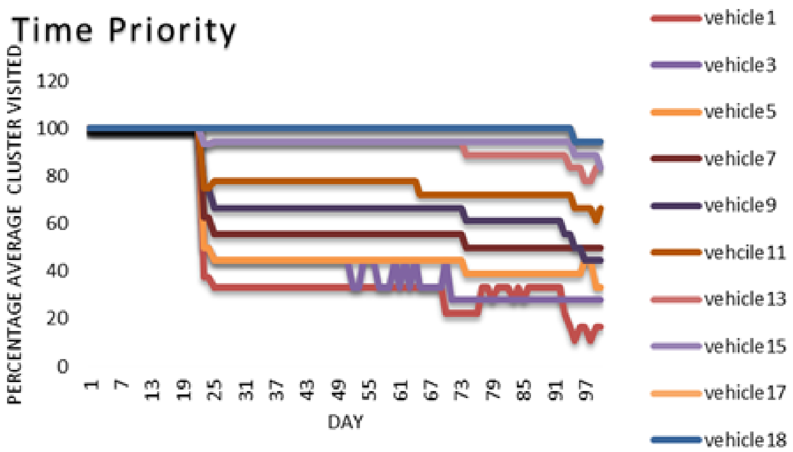

Figure 4.10: Time Priority: Day vs Percentage Cluster Visit Rate.

3. **Day vs Average Distance:** Fig 4.11 and fig 4.12 indicates the variation of average distance traveled by the emergency ferries with the increase of days and for a various number of the emergency ferry. The average distance increases as the number of days increases. Because as the number of days increases the demand for the emergency equipment also increases. So the emergency ferries need to move more distances. This plot shows us how the total average distance that needs to be traveled, increases with the number of days.

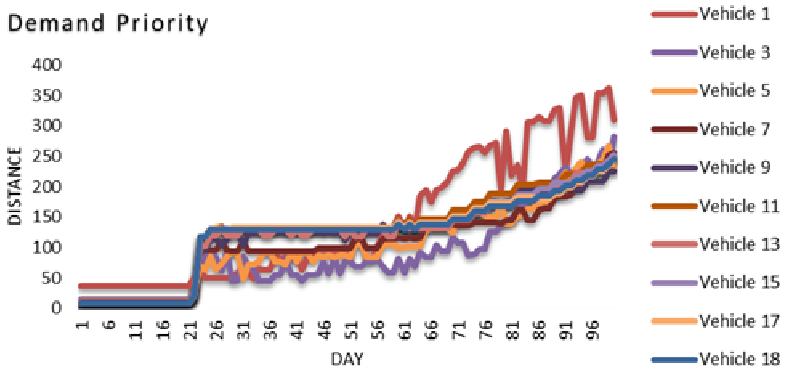

Figure 4.11: Demand Priority: Day vs Distance.

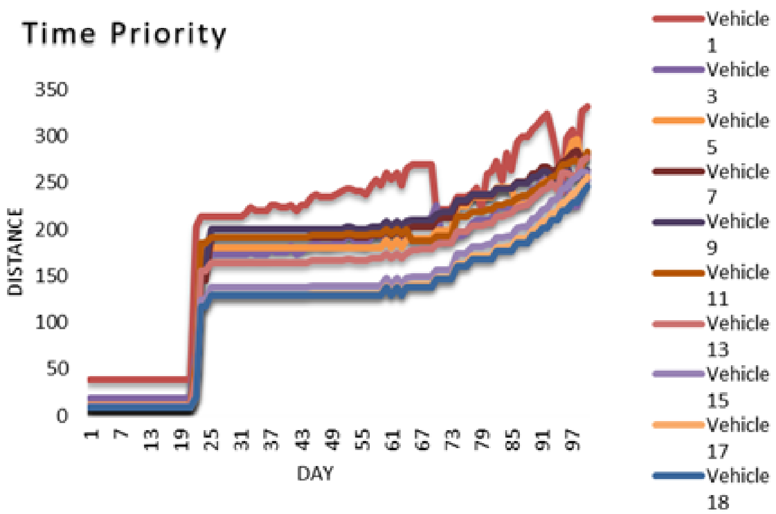

Figure 4.12: Time Priority: Day vs Distance.

4. **Day vs Average Time:** Fig 4.13 and fig 4.14 indicates the variation of average time taken by the emergency ferries to deliver the equipment with the increase of days and for a various number of the emergency ferry. This plot shows us how the total average time to deliver the equipment increases with time.

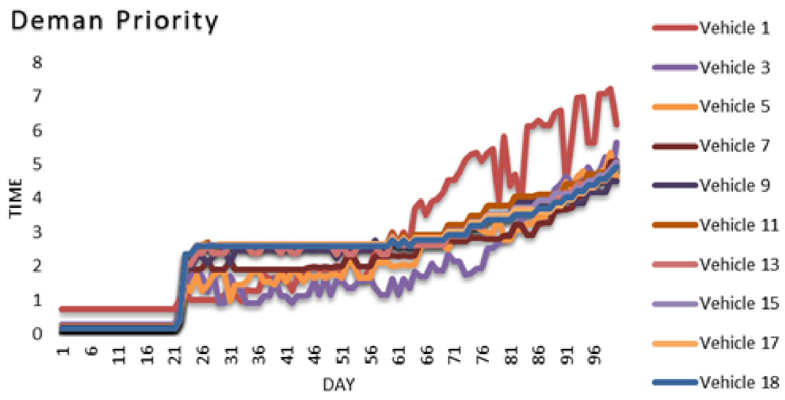

Figure 4.13: Demand Priority: Day vs Time.

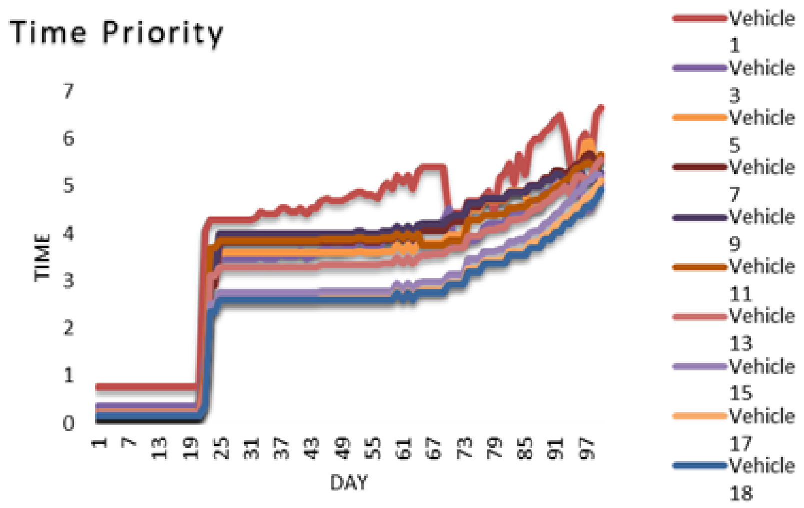

Figure 4.14: Time Priority: Day vs Time.

4.2.2 Percentage Average Delivery Rate: Bar Plot

In this section, for various numbers of Emergency ferry, the overall average percentage cluster visit rate, average percentage delivery rate, the average delivery rate per hour, and per km is calculated. To calculate the overall percentage average delivery rate for each Emergency ferry, the percentage delivery rate for each day for a different number of the Emergency ferry is calculated as before. Then the average over a total of 100 days is taken to get the overall average percentage delivery rate. Similarly, we get the percentage cluster visited as before, then the average over total 100 days is calculated to get the overall percentage average cluster visited rate. Then the delivered amount of equipment per km and per hour is taken for each day. To calculate the delivered amount per km and per hour the following formula have taken into consideration:

- **Delivered Per km each day** = (Total delivered in each day) / (total average distance traveled in each day)
- **Delivered Per hour each day** = (Total delivered in each day) / (total average time is taken in each day)

To get the average distance and average time, the average of distance and time has taken into consideration depending on a different number of emergency ferry.

- **Average distance in each day** = (total distance in each day / number of emergency ferry)
- **Average distance in each day** = (total time is taken in each day/number of the emergency ferry)

After the calculation 4 different bar plot is plotted considering these calculations.

1. **Percentage average delivery rate per emergency ferry:** For a different number of the emergency ferry, the Percentage average delivery rate is plotted in fig 4.15. In this graph, it is seen that as the number of Emergency ferries increases the delivery rate also increases. This is because more emergency ferry can carry more equipment and as a result, more equipment are delivered.

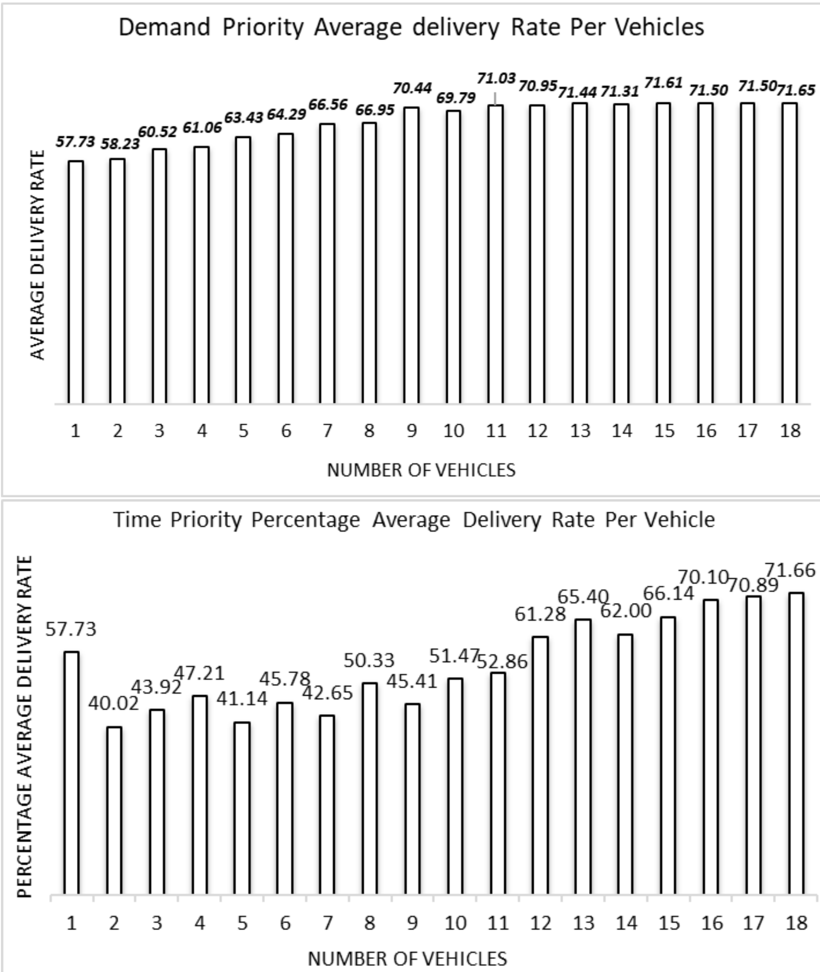

Figure 4.15: Percentage average delivery rate per Emergency ferry.

2. **Percentage average cluster visited rate per emergency ferry:** For a different number of the Emergency ferry, the Percentage average cluster visited rate is plotted in fig 4.16. This graph shows that it is possible to increase the cluster visited rate if we increase the number of the Emergency ferry. If we increase the number of the emergency ferry, the emergency ferry can visit more cluster.

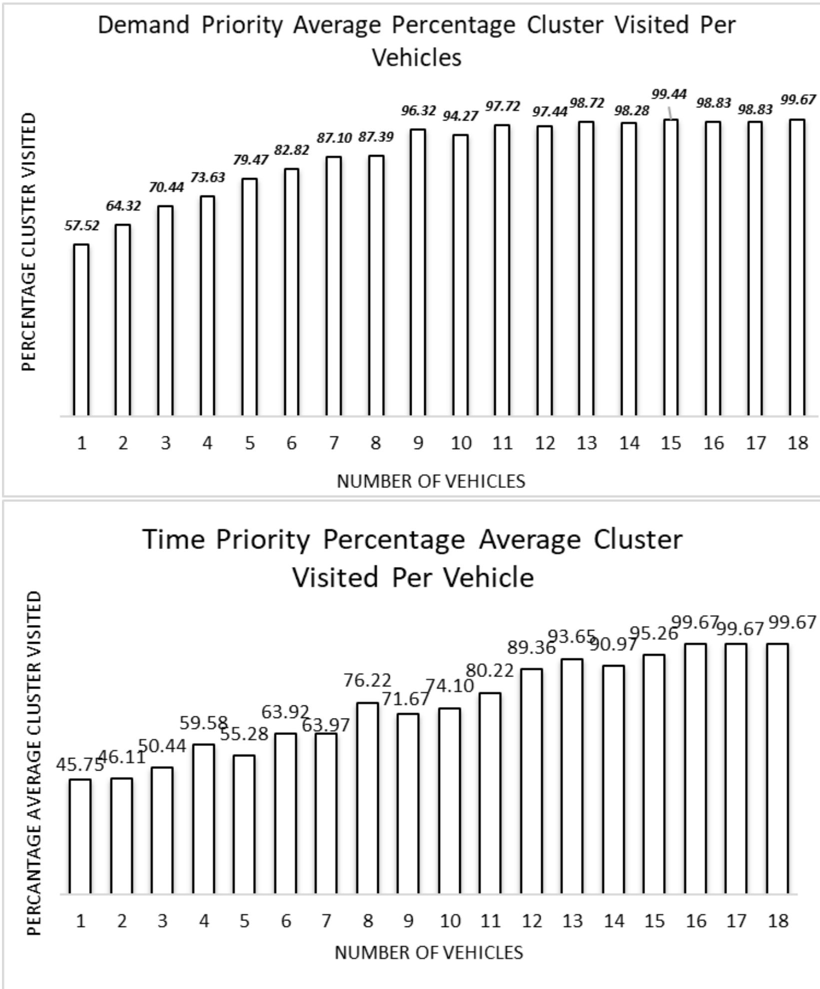

Figure 4.16: Percentage average cluster visited rate per Emergency ferry

3. **Delivery rate per km per emergency ferry:** In fig 4.17 the delivery rate per km is plotted under a different number of the emergency ferry. From the graphs, it can be seen that the more emergency ferry the delivered amount of equipment is also more per km. Because more emergency ferry can carry more equipment and so delivered more in a certain range of distance.

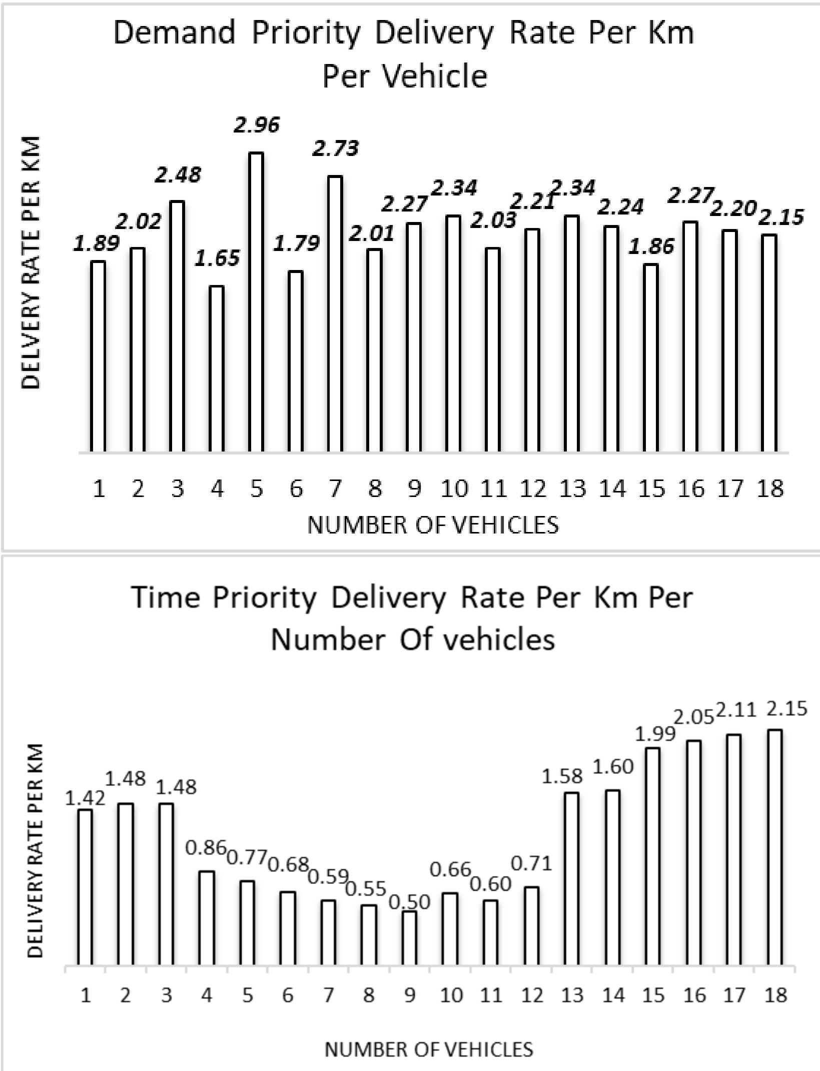

Figure 4.17: Delivery rate per km per Emergency ferry

4. **Delivery rate per hour per emergency ferry:** In fig 4.18 the delivery rate per hour is plotted under a different number of the emergency ferry. From the graphs, it can be seen that the more emergency ferry the delivered amount of equipment is also more per hour. Because more emergency ferry can carry more equipment and so delivered more in a definite time.

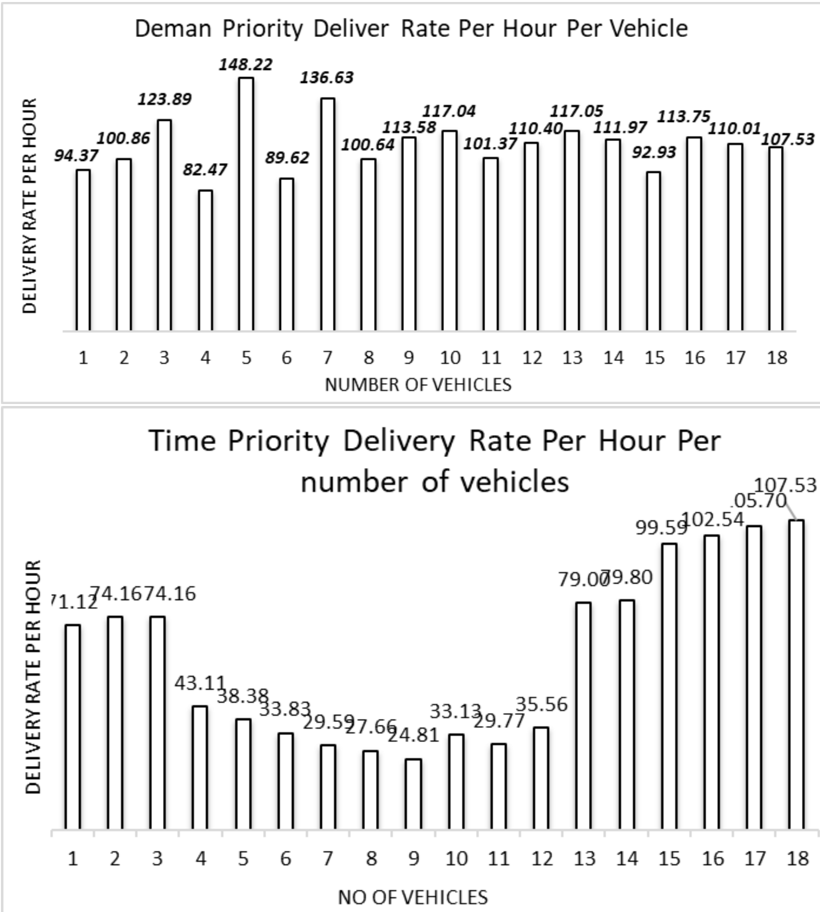

Figure 4.18: Delivery rate per hour per Emergency ferry

4.3 Analysis of Result

From the result section, it is clear that the defined objective of this research study was achieved successfully. After analyzing the result, it is seen that we can get a successful prediction of the COVID-19 future case with an accuracy of 93.1% using the Holt Winter Model. We are also able to design an Emergency Ferry system using the predicted data. Using the Emergency ferry it is possible to deliver the necessary equipment with success to different parts of the country. This will ensure the proper management of the essential resources and tackle the spread of the virus efficiently.

Chapter 5

Conclusion

In our research study, we have proposed a system that includes a Machine Learning model and an emergency ferry system. The results of both the system are satisfactory enough to go for the live implementation of our research work. We have also compared different research work of the same field and find out that our proposal of combining two things is unique from any other research work. We hope that this will help the world to reduce the bad effect of viruses like COVID-19. So from the overall architecture of the model, it can be clearly said that machine learning will be very much helpful in predicting the infected cases. Later the problem of spreading the virus can be easily solved by the proposed emergency ferry system that will be used to provide the necessary equipment. So by implementing the proposed models, the pace of this virus from spreading can be reduced.

We have also some scope of taking our work to the next level. It will be very much helpful if we can make an ML app where the data will go automatically and the prediction of the future day will take place in the app itself. After that when the emergency ferry will reach to each place and will provide sufficient things, it will be shown in the app that every patient has got their sufficient things. If we can maintain this app properly, the whole thing will be nicely organized.

REFERENCES

[1] C. Wang, P. Horby, F. Hayden, and G. Gao, "A novel coronavirus outbreak of global health concern," *The Lancet*, vol. 395, 2020.

[2] Z. Xu, L. Shi, Y. Wang, J. Zhang, L. Huang, C. Zhang, S. Liu, P. Zhao, H. Liu, L. Zhu, Y. Tai, C. Bai, T. Gao, J.-W. Song, P. Xia, J. Dong, J. Zhao, and F.-S. Wang, "Pathological findings of covid-19 associated with acute respiratory distress syndrome," *The Lancet Respiratory Medicine*, vol. 8, 2020.

[3] C. Huang, Y. Wang, X. Li, L. Ren, J. Zhao, Y. Hu, L. Zhang, G. Fan, J. Xu, X. Gu, Z. Cheng, T. Yu, J. Xia, Y. Wei, W. Wu, X. Xie, W. Yin, H. Li, M. Liu, Y. Xiao, H. Gao, L. Guo, J. Xie, G. Wang, R. Jiang, Z. Gao, Q. Jin, J. Wang, and B. Cao, "Clinical features of patients infected with 2019 novel coronavirus in wuhan, china," *The Lancet*, vol. 395, no. 10223, pp. 497–506, 2020.

[4] N. Chen, M. Zhou, X. Dong, J. Qu, F. Gong, Y. Han, Y. Qiu, J. Wang, Y. Liu, Y. Wei, J. Xia, T. Yu, X. Zhang, and L. Zhang, "Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in wuhan, china: a descriptive study," *The Lancet*, vol. 395, 2020.

[5] "Worldometer," Accessed: 17-June-2020. [Online]. Available: https://www.worldometers.info/coronavirus

[6] "World health organization," Accessed: 17-June-2020. [Online]. Available: https://www.euro.who.int/en/

[7] J. Li, K. Guo, E. Herrera-Viedma, H. Lee, J. Liu, N. Zhong, L. Gomes, f. g. Filip, S.-C. Fang, M. Sagir Ozdemir, X. Liu, G. Lu, and Y. Shi, "Culture vs policy: More global collaboration to effectively combat covid-19," *The Innovation*, vol. 1, 2020.

[8] A. Islam and T. Biswas, "Health system in bangladesh: Challenges and opportunities," *American Journal of Health Research*, vol. 2, p. 366, 2014.

[9] "Institute of epidemiology, disease control and research," Accessed: 17-June-2020. [Online]. Available: https://iedcr.gov.bd/

[10] E. Jimenez-Solem, T. S. Petersen, C. Hansen, C. Hansen, C. Lioma, C. Igel, W. Boomsma, O. Krause, S. Lorenzen, R. Selvan *et al.*, “Developing and validating covid-19 adverse outcome risk prediction models from a bi-national european cohort of 5594 patients,” *Scientific reports*, vol. 11, no. 1, pp. 1–12, 2021.

[11] T. M. Awan and F. Aslam, “Prediction of daily covid-19 cases in european countries using automatic arima model,” *Journal of Public Health Research*, vol. 9, no. 3, 2020.

[12] J. Wangping, H. Ke, S. Yang, C. Wenzhe, W. Shengshu, Y. Shanshan, W. Jianwei, K. Fuyin, T. Penggang, L. Jing *et al.*, “Extended sir prediction of the epidemics trend of covid-19 in italy and compared with hunan, china,” *Frontiers in medicine*, vol. 7, p. 169, 2020.

[13] R. Chowdhury, K. Heng, M. S. R. Shawon, G. Goh, D. Okonofua, C. Ochoa-Rosales, V. Gonzalez-Jaramillo, A. Bhuiya, D. Reidpath, S. Prathapan *et al.*, “Dynamic interventions to control covid-19 pandemic: a multivariate prediction modelling study comparing 16 worldwide countries,” *European journal of epidemiology*, vol. 35, no. 5, pp. 389–399, 2020.

[14] A. Bhadra, A. Mukherjee, and K. Sarkar, “Impact of population density on covid-19 infected and mortality rate in india,” *Modeling Earth Systems and Environment*, vol. 7, no. 1, pp. 623–629, 2021.

[15] D. Gysi, Í. Faria do Valle, M. Zitnik, A. Ameli, X. Gan, O. Varol, S. Ghiassian, J. Patten, R. Davey, J. Loscalzo, and A.-L. Barabasi, “Network medicine framework for identifying drug repurposing opportunities for covid-19,” *ArXiv*, 2020.

[16] C. Anastassopoulou, L. Russo, A. Tsakris, and C. Siettos, “Data-based analysis, modelling and forecasting of the covid-19 outbreak,” *PloS one*, vol. 15, no. 3, p. e0230405, 2020.

[17] Q. Guo and Z. He, “Prediction of the confirmed cases and deaths of global covid-19 using artificial intelligence,” *Environmental Science and Pollution Research*, vol. 28, 2021.

[18] L. Qin, Q. Sun, Y. Wang, K.-F. Wu, M. Chen, B.-C. Shia, and S.-Y. Wu, “Prediction of number of cases of 2019 novel coronavirus (covid-19) using social media search index,” *International Journal of Environmental Research and Public Health*, vol. 17, p. 2365, 2020.

[19] B. Zareie, A. Roshani, M. A. Mohammad, Mansournia, M. A. Rasouli, and G. Moradi, “A model for covid-19 prediction in iran based on china parameters,” *Archives of Iranian medicine*, vol. 23, pp. 244–248, 2020.

[20] M. Alazab, A. Awajan, A. Mesleh, A. Abraham, V. Jatana, and S. Alhyari, "Covid-19 prediction and detection using deep learning," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 12, pp. 168–181, 2020.

[21] J. Stübinger and L. Schneider, "Epidemiology of coronavirus covid-19: Forecasting the future incidence in different countries," *Healthcare*, vol. 8, p. 99, 2020.

[22] W. wei and X. Zhang, "An updated analysis of turning point, duration and attack rate of covid-19 outbreaks in major western countries with data of daily new cases," *Data in Brief*, vol. 31, p. 105830, 2020.

[23] A. Tomar and N. Gupta, "Prediction for the spread of covid-19 in india and effectiveness of preventive measures," *Science of the Total Environment*, vol. 728, 2020.

[24] A. Nalbant, C. Varım, S. Yaylaci, T. Kaya, and H. Cinemre, "Can the neutrophil/lymphocyte ratio (nlr) have a role in the diagnosis of coronavirus 2019 disease (covid-19)," *Revista da Associação Médica Brasileira*, vol. 66, pp. 746–751, 2020.

[25] F. Petropoulos and S. Makridakis, "Forecasting the novel coronavirus covid-19," *PloS one*, vol. 15, no. 3, p. e0231236, 2020.

[26] R. Pal, A. A. Sk, S. Kar, and D. Prasad, "Neural network based country wise risk prediction of covid-19," *Applied Sciences*, 2020.

[27] J. Vomlel, H. Kruzık, P. Tuma, J. Precek, and M. Hutyra, "Machine learning methods for mortality prediction in patients with st elevation myocardial infarction," *In proceedings of Workshop on Uncertainty Processing*, pp. 204–213, 2012.

[28] P. Kumar, H. Kalita, S. Sharma, Y. Sharma, C. Nanda, M. Rani, J. Rahmani, and A. Bhagavathula, "Forecasting the dynamics of covid-19 pandemic in top 15 countries in april 2020: Arima model with machine learning approach," *MedRxiv*, 2020.

[29] A. Khakharia, V. Shah, S. Jain, J. Shah, A. Tiwari, P. Daphal, M. Warang, and N. Mehendale, "Outbreak prediction of covid-19 patients for dense and populated countries using machine learning," *SSRN Electronic Journal*, 2020.

[30] R. Sujath, J. M. Chatterjee, and A. E. Hassanien, "A machine learning forecasting model for covid-19 pandemic in india," *Stochastic Environmental Research and Risk Assessment*, vol. 34, pp. 959–972, 2020.

[31] A. Luceño and D. Peña, *Autoregressive Integrated Moving Average (ARIMA) Modeling*. John Wiley & Sons, 2008.

[32] H. Sverdrup and K. Ragnarsdottir, "Natural resources in a planetary perspective," *Geochem. Perspect.*, vol. 3, 2014.

[33] T. Essington, J. Kitchell, and C. Walters, "The von bertalanffy growth function, bioenergetics, and the consumption rates of fish," *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 58, pp. 2129–2138, 2001.

[34] A. Tsoularis and J. Wallace, "Analysis of logistic growth models," *Mathematical biosciences*, vol. 179, pp. 21–55, 2002.

[35] P. Wang, X. Zheng, J. Li, and B. Zhu, "Prediction of epidemic trends in covid-19 with logistic model and machine learning technics," *Chaos, Solitons & Fractals*, vol. 139, p. 110058, 2020.

[36] E. Valakevicius and M. Bražȯnas, "Application of seasonal holt-winter model for the prediction of exchange rate volatility," *Engineering Economics*, vol. 26, 2015.

[37] "Investopedia," Accessed: 20-March-2021. [Online]. Available: https://www.investopedia.com/terms/t/timeseries.asp

[38] S. Bakar and M. Ibrahim, "Optimal solution for travelling salesman problem using heuristic shortest path algorithm with imprecise arc length," in *AIP Conference Proceedings*, vol. 1870, 2017, p. 040061.

APPENDICES

All the Codes used in this research is included here.

Importing Libraries:

```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
#from sklearn.linear_model import LinearRegression,ElasticNet
#from sklearn.svm import SVR
#from sklearn.preprocessing import PolynomialFeatures
#from sklearn.metrics import r2_score,mean_absolute_error
import math
#from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import matplotlib.pyplot as plt
#from sklearn.linear_model import Ridge, RidgeCV, ElasticNet, LassoCV, LassoLarsCV,Lasso
from sklearn.model_selection import cross_val_score
import sklearn.metrics as sklm
from scipy.optimize import curve_fit
from scipy.optimize import fsolve
import warnings
warnings.filterwarnings("ignore")
```

Reading Data Frame:

```
frame=pd.read_excel("COVID-19.xlsx")
frame.drop(["geoId","day","month","year","countryterritoryCode","continentExp","Unnamed: 11","Unnamed: 12"],axis=1,inplace=True)
frame.columns=[["Date","Confirmed","Deaths","Country","Population"]]
frame.drop("Population",axis=1,inplace=True)
frame.columns = frame.columns.get_level_values(0)
frame["Date"]=pd.to_datetime(frame["Date"],format="%d/%m/%Y")
frame.head()
```

Data preprocessing:

```
frame["Total_Confirmed"]=frame.groupby("Country")["Confirmed"].cumsum()
frame["Total_Deaths"]=frame.groupby("Country")["Deaths"].cumsum()
df_Bangladesh=frame[frame["Country"]=="Bangladesh"]
df_world=frame[frame["Country"]!="Bangladesh"]
df_world=df_world[df_world["Total_Confirmed"]>0]
df_world['Days'] = (df_world['Date'] - df_world.groupby('Country')['Date'].transform('first'))
df_Bangladesh['Days'] = (df_Bangladesh['Date'] - df_Bangladesh.groupby('Country')['Date'].transform('first'))
a=df_world["Country"].unique()
df_world["Days"]=df_world["Days"].astype(str)
df_Bangladesh["Days"]=df_Bangladesh["Days"].astype(str)
df_world["Days"] = df_world["Days"].str.split(' ').str[0]
df_Bangladesh["Days"] = df_Bangladesh["Days"].str.split(' ').str[0]
df_world["Days"]=df_world["Days"].astype(int)
df_Bangladesh["Days"]=df_Bangladesh["Days"].astype(int)
df_world["Days"]=df_world["Days"]+1
df_Bangladesh["Days"]=df_Bangladesh["Days"]+1
df_USA=df_world[df_world["Country"]=="United_States_of_America"]
df_USA=df_USA[["Days","Total_Confirmed"]]
df_UK=df_world[df_world["Country"]=="United_Kingdom"]
df_UK=df_UK[["Days","Total_Confirmed"]]
df_Italy=df_world[df_world["Country"]=="Italy"]
df_Italy=df_Italy[["Days","Total_Confirmed"]]
df_Spain=df_world[df_world["Country"]=="Spain"]
df_Spain=df_Spain[["Days","Total_Confirmed"]]
df_France=df_world[df_world["Country"]=="France"]
df_France=df_France[["Days","Total_Confirmed"]]
df_India=df_world[df_world["Country"]=="India"]
df_India=df_India[["Days","Total_Confirmed"]]
df_Pakistan=df_world[df_world["Country"]=="Pakistan"]
df_Pakistan=df_Pakistan[["Days","Total_Confirmed"]]
df_Bangladesh=df_Bangladesh[["Days","Total_Confirmed"]]
df_Bangladesh=df_Bangladesh.set_index("Days")
df_Bangladesh.head()
```

Train test split:

```
train,test=train_test_split(df_Bangladesh,test_size=0.2,shuffle=False)
```

Data plotting:

```
import plotly.express as px
fig = px.line(train,x=train.index,y="Total_Confirmed" ,title='Total_Confirmed casec by day')
fig.show()
fig = px.line(test,x=test.index,y="Total_Confirmed" ,title='Total_Confirmed casec by day')
fig.show()
```

Check the seasonality and trend of the data:

```
from statsmodels.tsa.stattools import adfuller
def test_stationarity(timeseries):

    #Determing rolling statistics
    rolmean = pd.Series(timeseries).rolling(window=12).mean()
    rolstd = pd.Series(timeseries).rolling(window=12).std()

    #Plot rolling statistics:
    orig = plt.plot(timeseries, color='blue',label='Original')
    mean = plt.plot(rolmean, color='red', label='Rolling Mean')
    std = plt.plot(rolstd, color='black', label = 'Rolling Std')
    plt.legend(loc='best')
    plt.title('Rolling Mean & Standard Deviation')
    plt.show(block=False)

    #Perform Dickey-Fuller test:1
    print ('Results of Dickey-Fuller Test:')
    dftest = adfuller(timeseries, autolag='AIC')
    dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags Used','Number of Observations Used'])
    for key,value in dftest[4].items():
        dfoutput['Critical Value (%s)'%key] = value
    print (dfoutput)

test_stationarity(train["Total_Confirmed"])
ts_log_train = np.log(train)
plt.plot(ts_log_train)
ts_log_diff_train = ts_log_train - ts_log_train.shift(1)
plt.plot(ts_log_diff_train)
ts_log_diff_train.dropna(inplace=True)
test_stationarity(ts_log_diff_train["Total_Confirmed"])
from statsmodels.tsa.stattools import acf, pacf
lag_acf_train = acf(ts_log_diff_train, nlags=10)
lag_pacf_train = pacf(ts_log_diff_train, nlags=10, method='ols'

#Plot ACF:
plt.Figure(figsize=(2,3))
plt.plot(lag_acf_train)
plt.axhline(y=0,linestyle='--',color='gray')
plt.axhline(y=-1.96/np.sqrt(len(ts_log_diff_train)),linestyle='--',color='gray')
plt.axhline(y=1.96/np.sqrt(len(ts_log_diff_train)),linestyle='--',color='gray')
plt.title('Autocorrelation Function')

#Plot PACF:
plt.plot(lag_pacf_train)
plt.axhline(y=0,linestyle='--',color='gray')
plt.axhline(y=-1.96/np.sqrt(len(ts_log_diff_train)),linestyle='--',color='gray')
plt.axhline(y=1.96/np.sqrt(len(ts_log_diff_train)),linestyle='--',color='gray')
plt.title('Partial Autocorrelation Function')
plt.tight_layout()
from statsmodels.tsa.arima_model import ARIMA
from statsmodels.tsa.arima_model import ARIMAResults

model = ARIMA(ts_log_train, order=(1,1,0))
results_ARIMA = model.fit(disp=0)
plt.plot(ts_log_diff_train)
plt.plot(results_ARIMA.fittedvalues, color='red')
a=sum((results_ARIMA.fittedvalues-ts_log_diff_train["Total_Confirmed"])**2)
plt.title('RSS: '+str(a))
#predicted=ARIMAResults.forecast(results_ARIMA,steps=30)[0]
results_ARIMA.summary()
predictions_ARIMA_diff = pd.Series(results_ARIMA.fittedvalues, copy=True)
print(predictions_ARIMA_diff.head())
predictions_ARIMA_diff_cumsum = predictions_ARIMA_diff.cumsum()
predictions_ARIMA_diff_cumsum.head()
predictions_ARIMA_log = pd.Series(ts_log_train.iat[1,0], index=ts_log_train.index)
predictions_ARIMA_log = predictions_ARIMA_log.add(predictions_ARIMA_diff_cumsum,fill_value=0)
predictions_ARIMA_log.head()

df_Bangladesh.reset_index(inplace=True)
train.reset_index(inplace = True)
test.reset_index(inplace = True)
```

Model training and prediction:
```
def logistic_model(x,a,b,c):
    return c/(1+np.exp(-(x-b)/a))
def expo_model(x,p,N0=3,X0=1):
    return N0*(1+p)**(x-X0)
def gompertz_model(x,a,b,c):
    return c*np.exp(-b*np.exp(-x/a))
def bertalanffy_model(x,a,b,c,X0=1):
    return (c*(1-np.exp(-a*(x-X0))))**b

turn_off_Gomp = True
x = np.array(list(train['Days'].values))
y = np.array(list(train['Total_Confirmed'].values))
y0 = y[0]
yf = y[-1]
print("Initial number of cases: ", y0)
print("Current number of cases: ", yf)

print('>>>')
print('>>> ARIMA Model')
predicted=ARIMAResults.forecast(results_ARIMA,steps=11)[0]
predicted=np.exp(predicted)
predicted=predicted.astype(int)
test_case_ARIMA={"Days":test.Days,"Total_Confirmed":predicted}
test_case_ARIMA=pd.DataFrame(test_case_ARIMA)
MSLE=sklm.mean_squared_log_error(test["Total_Confirmed"],predicted)
print("Mean squared log error (MSLE): ", '{:.3f}'.format(MSLE))
print("Exp of RMSLE: ", '{:.3f}'.format(np.exp(np.sqrt(MSLE))))
print("R2 score: ", '{:.3f}'.format(sklm.r2_score(test["Total_Confirmed"],predicted)))
RMSE=np.sqrt(sum((predicted-test["Total_Confirmed"])**2))
print("RMSE score: ", '{:.3f}'.format(RMSE))

print('>>>')
print('>>> Logistic Model')
fit_i = curve_fit(logistic_model,x,y,p0=[3,50,100000])#, bounds=([0,0,0],[10,100,150000]))
ai,bi,ci = fit_i[0]
sigma_ai, sigma_bi, sigma_ci = np.sqrt(np.diag(fit_i[1]))
print([ss+'='+'{:.3f}'.format(xx) for ss, xx in zip(('a','b','c'),(ai,bi,ci))])
print([ss+'='+'{:.3f}'.format(xx) for ss, xx in zip(('sigma_a','sigma_b','sigma_c'),(sigma_ai,sigma_bi,sigma_ci))])
y_pred_logistic = logistic_model(test["Days"],ai,bi,ci)
pred_upper_logistic=logistic_model(test["Days"],ai+sigma_ai,bi+sigma_bi,ci+sigma_ci)
pred_lower_logistic=logistic_model(test["Days"],ai-sigma_ai,bi-sigma_bi,ci-sigma_ci)
test_case_logistic={"Days":test.Days,"Total_Confirmed":y_pred_logistic}
MSLE=sklm.mean_squared_log_error(test["Total_Confirmed"],y_pred_logistic)
print("Mean squared log error (MSLE): ", '{:.3f}'.format(MSLE))
print("Exp of RMSLE: ", '{:.3f}'.format(np.exp(np.sqrt(MSLE))))
print("R2 score: ", '{:.3f}'.format(sklm.r2_score(test["Total_Confirmed"],y_pred_logistic)))
RMSE=np.sqrt(sum((y_pred_logistic-test["Total_Confirmed"])**2))
print("RMSE score: ", '{:.3f}'.format(RMSE))

print('***')
print('*** Gompertz Model')
ini_guess = [1,10,ci]  ##take initial guess from result of Logistic model
fit_i = curve_fit(gompertz_model,x,y,p0=ini_guess)
ai,bi,ci = fit_i[0]
sigma_ai, sigma_bi, sigma_ci = np.sqrt(np.diag(fit_i[1]))
print([ss+'='+'{:.3f}'.format(xx) for ss, xx in zip(('a','b','c'),(ai,bi,ci))])
print([ss+'='+'{:.3f}'.format(xx) for ss, xx in zip(('sigma_a','sigma_b','sigma_c'),(sigma_ai,sigma_bi,sigma_ci))])
y_pred_gompertz = gompertz_model(test["Days"],ai,bi,ci)
pred_upper_gompertz=gompertz_model(test["Days"],ai+sigma_ai,bi+sigma_bi,ci+sigma_ci)
pred_lower_gompertz=gompertz_model(test["Days"],ai-sigma_ai,bi-sigma_bi,ci-sigma_ci)
test_case_gompertz={"Days":test.Days,"Total_Confirmed":y_pred_gompertz}
MSLE=sklm.mean_squared_log_error(test["Total_Confirmed"],y_pred_gompertz)
print("Mean squared log error (MSLE): ", '{:.3f}'.format(MSLE))
print("Exp of RMSLE: ", '{:.3f}'.format(np.exp(np.sqrt(MSLE))))
print("R2 score: ", '{:.3f}'.format(sklm.r2_score(test["Total_Confirmed"],y_pred_gompertz)))
RMSE=np.sqrt(sum((y_pred_gompertz-test["Total_Confirmed"])**2))
print("RMSE score: ", '{:.3f}'.format(RMSE))

print('***')
print('*** bertalanffy Model')
ini_guess = [1,10,ci]  ##take initial guess from result of Logistic model
fit_i = curve_fit(bertalanffy_model,x,y,p0=ini_guess,maxfev = 10000)
ai,bi,ci = fit_i[0]
sigma_ai, sigma_bi, sigma_ci = np.sqrt(np.diag(fit_i[1]))
print([ss+'='+'{:.3f}'.format(xx) for ss, xx in zip(('a','b','c'),(ai,bi,ci))])
print([ss+'='+'{:.3f}'.format(xx) for ss, xx in zip(('sigma_a','sigma_b','sigma_c'),(sigma_ai,sigma_bi,sigma_ci))])
y_pred_bertalanffy= bertalanffy_model(test["Days"],ai,bi,ci)
```

```
pred_upper_bertalanffy=bertalanffy_model(test["Days"],ai+sigma_ai,bi+sigma_bi,ci+sigma_ci)
pred_lower_bertalanffy=bertalanffy_model(test["Days"],ai-sigma_ai,bi-sigma_bi,ci-sigma_ci)
test_case_bertalanffy={"Days":test.Days,"Total_Confirmed":y_pred_bertalanffy}
MSLE=sklm.mean_squared_log_error(test["Total_Confirmed"],y_pred_bertalanffy)
print("Mean squared log error (MSLE): ", '{:.3f}'.format(MSLE))
print("Exp of RMSLE: ", '{:.3f}'.format(np.exp(np.sqrt(MSLE))))
print("R2 score: ", '{:.3f}'.format(sklm.r2_score(test["Total_Confirmed"],y_pred_bertalanffy)))
RMSE=np.sqrt(sum((y_pred_bertalanffy-test["Total_Confirmed"])**2))
print("RMSE score: ", '{:.3f}'.format(RMSE))

print('###')
print('### Exponential Model')
fit_exp = curve_fit((lambda tt,pp: expo_model(tt,pp,N0=y0)),x,y,p0=[0.10])  ##Take N0 from initial number of reported cases
pi, sigma_pi = fit_exp[0][0], np.sqrt(fit_exp[1]).flatten()[0]
print([ss+'='+'{:.3f}'.format(xx) for ss, xx in zip(('p','sigma_p'),(pi,sigma_pi))])
y_pred_expo = expo_model(test["Days"],pi)
pred_upper_expo=expo_model(test["Days"],pi+sigma_pi)
pred_lower_expo=expo_model(test["Days"],pi-sigma_pi)
test_case_expo={"Days":test.Days,"Total_Confirmed":y_pred_expo}
MSLE=sklm.mean_squared_log_error(test["Total_Confirmed"],y_pred_expo)
print("Mean squared log error (MSLE): ", '{:.3f}'.format(MSLE))
print("Exp of RMSLE: ", '{:.3f}'.format(np.exp(np.sqrt(MSLE))))
print("R2 score: ", '{:.3f}'.format(sklm.r2_score(test["Total_Confirmed"],y_pred_expo)))
RMSE=np.sqrt(sum((y_pred_expo-test["Total_Confirmed"])**2))
print("RMSE score: ", '{:.3f}'.format(RMSE))

Printing the results:
plt.figure(figsize=(20,5))
plt.plot(df_Bangladesh["Days"],df_Bangladesh["Total_Confirmed"],c='g',label="Actual_Case")
plt.plot(test_case_expo["Days"],test_case_expo["Total_Confirmed"],c='b',label="Exponential_Model")
plt.plot(test_case_gompertz["Days"],test_case_gompertz["Total_Confirmed"],c='r',label="Gompertz_Model")
plt.plot(test_case_logistic["Days"],test_case_logistic["Total_Confirmed"],c='r',label="Logistic_Model",alpha=1)
plt.plot(test_case_ARIMA["Days"],test_case_ARIMA["Total_Confirmed"],c='grey',label="ARIMA_Model",alpha=1)
plt.fill_between(test["Days"],pred_upper_expo,pred_lower_expo,color='red',alpha=0.5)
plt.fill_between(test["Days"],pred_upper_gompertz,pred_lower_gompertz,color='orange',alpha=0.5)
plt.fill_between(test["Days"],pred_upper_logistic,pred_lower_logistic,color='black',alpha=1)
plt.axvline(test_case.Days[0], ymin=0.1, ymax=0.7,label = 'Train_Test_Split')
plt.legend()
plt.figure(figsize=(20,10))
plt.plot(df_Bangladesh["Days"],df_Bangladesh["Total_Confirmed"],c='g',label="Actual_Case")
plt.plot(test_case_bertalanffy["Days"],test_case_bertalanffy["Total_Confirmed"],c='r',label="bertalanffy_Model",alpha=1)
plt.fill_between(test["Days"],pred_upper_bertalanffy,pred_lower_bertalanffy,color='orange',alpha=0.2)
plt.axvline(test_case.Days[0], ymin=0, ymax=0.7,label = 'Train_Test_Split')
plt.legend()
```