Machine learning-based approaches to forecast COVID-19(SARS-CoV-2) cases, death, recovery and comparative analysis between them from the perspective of Bangladesh

Chayti Saha
Comilla University
Dept. of CSE
chaytisaha02121998@gmail.com

Fozilatunnesa Masuma
Comilla University
Dept. of CSE
simin.cou@gmail.com

***Abstract*—This research attempts to predict the infected, death and recovery cases for Covid-19 using ML techniques FB Prophet, ARIMA, SARIMAX, deep learning technique LSTM and compare between them to find out the best model for prediction. For case 'Detected' and 'Recovery', LSTM performs better and for case 'Death', SARIMAX performs better.**

***Index Terms*—Coronavirus disease 2019 (COVID-19), Pandemic, Machine Learning (ML), Facebook Prophet (FB Prophet), AutoRegressive Integrated Moving Average (ARIMA), Seasonal AutoRegressive Integrated Moving Averages with eXogenous regressors (SARIMAX), Long short-term memory (LSTM) and Root Mean Square Error (RMSE)**

I. Introduction

Coronavirus disease 2019 (COVID-19) is a contagious disease caused by a novel coronavirus. Now it is called severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2; formerly called 2019-nCoV), which was first identified in Wuhan City, Hubei Province, China. It was initially reported to the WHO on December 31, 2019. On January 30, 2020, the WHO declared the COVID-19 outbreak a global health emergency which has become a global pandemic according to the declaration of the WHO on March 11, 2020. The number of infected patients around the world is increasing daily at an alarming exponential growth rate. In Bangladesh, infected patients were first found on March 08, 2020. The number is increasing day by day.

Some Corona viruses can be transmitted from one person to another person. But it usually happens after close contact with an infected patient. It has some common signs like respiratory symptoms, cough, fever, different types of breathing difficulties. In more severe cases, infection can cause pneumonia, severe acute respiratory syndrome, kidney failure and the ultimate outcome is death. Most of the infected people experience mild to moderate respiratory illness and recover without requiring any type of special treatment. Older people and those having other medical problems like cardiovascular disease, chronic respiratory disease, diabetes and cancer are more likely to develop serious illness.

Situation data and insight regarding those data is always important in various sectors also in that pandemic case. From the situation data analysis on COVID-19, we can decide the future steps and take decisions like, which area to put under strict lockdown, which area is less risky or which area has a sudden growth of infected cases, how much cautions we should take. A comprehensive dataset for this analysis and possible future usage in the field of artificial intelligence is prepared here which can also predict infected cases, death cases and recovery cases. Besides, here we have used ML techniques like FB prophet, ARIMA, SARIMAX and deep learning technique LSTM to predict each of these cases and also compared between them.

II. Literature Review

As COVID-19 is a global pandemic which has not exterminated yet, the size of the dataset is increasing day by day. Many of the research works have done related this area and trying to analyze it from all aspects. But only few of them are found and the methodology of the most of these works are descriptive [1], like commentary [2] from the perspective of Bangladesh. Authors in [1] propose the study to develop something relating to SMEs and their management. Authors in [2] propose to highlight the situation of COVID-19 in Bangladesh, Government steps to manage this unprecedented condition and increasing public health challenges. Same types of works are done in papers [3], [4]. A very small number of research works in this field perspective Bangladesh using machine learning have found. Authors in [5] propose to analyze the scenario of Bangladesh because of COVID-19 with gender and age vulnerability till June 2020 and to compare the situation among the eight divisions of Bangladesh using the machine learning techniques. From the above list of works, it can be said, the research domain is still active and yet to contribute for providing with more new aspects.

III. METHODOLOGY

This research work is divided into several parts and each module is developed in order. The steps to develop this system can be broadly divided into the following module:

- Data Collection:
- Data Preparation:
- Model Development:
- Implementation:

A brief overview of each module is discussed below:

A. Data Collection

The first step is to have time series dataset. The data collection was mainly from authentic sources [6] – [11] and from the live Facebook situation briefing by IEDCR [12]. Few data had to be collected and verified by national newspapers [13] – [14]. Our dataset compiles all the information disclosed by the government in excel file. The dataset contains many attributes like 'Detected per day', 'Death per day', 'Recovery per day' etc. from March 08, 2020 to December 31, 2020. Data on a regular basis will be collected and added to this dataset. This dataset will open up the doors for future research work under a suitable license easily.

B. Data Preparation

The raw data contains some missing values that may degrade system performance and may also create unpredictable problems. In this step, the raw data is analyzed to remove these missing values. All of the missing fields were filled with the value of following field. Other processing methods are also used according to the algorithm technique.

C. Model Development

ML algorithms like FB prophet, ARIIMA, SARIMAX and deep learning algorithm are used for building 4 models where each of the four models are used for predicting each of the three cases (Detected per day, death per day, recovery per day) and at last comparison is also done for getting the model with better result for specific task.

FB Prophet is used to forecast time series data based on an additive model. Here, non-linear trends are fit with yearly, weekly and daily seasonality having holiday effects. It works best with time series that have strong seasonal effects. It carefully handles missing data and shifts in the trend.

The FB Prophet uses a decomposing time series model with three main model components like trend, seasonality, and holidays. These can be combined in the following equation:

$$y(t) = g(t) + s(t) + h(t) + \epsilon t$$

- g(t): piece wise linear or logistic growth curve for the modeling non-periodic changes in time series.
- s(t): periodic changes (e.g, weekly/yearly seasonality).
- h(t): the effects of holidays (user provided) with irregular schedules.
- ϵt: error term accounts for any unusual changes not accommodated by the model.

The Growth Function (and change points): The growth function gives a model for the overall trend of data. This can be presented at all points in the data or can be altered at change points. Change points are moments in the data where the data shifts direction.

The growth function has three main options:

- Linear Growth: This is a default setting for Prophet. It uses a set of piece wise linear equations with differing slopes between change points. When linear growth is used, the growth term will look similar to the classic $y = mx + b$ from middle school, except the slope (m) and offset (b) are variable and will change value at each change point.
- Logistic Growth: This setting is useful when time series has a cap or a floor in which the values we are modeling becomes saturated and can't surpass a maximum or minimum value. The growth term is similar to a typical equation for a logistic curve, except the carrying capacity (C) which will vary as a function of time and the growth rate (m) and the offset (m) are variable and will change the value at each change point.
- Flat: A flat trend is chosen when there is no growth over time but there still may be seasonality.

The Seasonality Function: This function is simply a Fourier Series as a function of time. Sine and cosine terms are multiplied by some coefficient. This sum can approximate nearly any curve or in the case of the FB Prophet.

ARIMA explains a given time series based on its own past values like its own lags and lagged forecast errors to use equation to forecast future values. It can be used for any 'non-seasonal' time series that exhibits patterns and is not a random white noise.

An ARIMA model is characterized by 3 terms like p, d, q. Here,

p is the order of the AR term

q is the order of the MA term

d is the number of differences required to make the time series stationary

If a time series, has seasonal patterns, addition of seasonal terms makes it SARIMA.

To build an ARIMA model, we have to make the time series stationary because the linear regression model uses its own lags as predictors and works best when the predictors are not correlated and are independent of each other. The most common way for making stationary is to subtract the previous value from the current value. This subtraction can be needed more then one time depending on the complexity of the series.

The value of d represents minimum number of differences needed to make the series stationary. And d becomes zero when the series becomes stationary.

A pure AR model is one where Y_t depends only on its own lags. That is, Y_t is a function of the 'lags of Y_t

$$Y_t = \alpha + \beta_1 Y_(t-1) + \beta_2 Y_(t-2) + ... + \beta_p Y_(t-p) + \epsilon_1 \quad (1)$$

where, $Y_(t-1)$is the lag1 of the series,$\beta 1$ is the coefficient of lag1 that the model estimates and $\alpha$is the intercept term, also estimated by the model.

Likewise a pure Moving Average (MA) model is one where Y_t depends only on the lagged forecast errors.

$$Y_t = \alpha + \epsilon_t + \phi_1\epsilon_(t-1) + \phi_2\epsilon_(t-2) + ... + \phi_q\epsilon_(t-q) \quad (2)$$

where the error terms are the errors of the auto regressive models of the respective lags. The errors ϵ_t and $\epsilon_(t-1)$ are the errors from the following equations:

$$Y_t = \beta_1 Y_(t-1) + \beta_2 Y_(t-2) + ... + \beta_0 Y_0 + \epsilon_t \quad (3)$$

$$Y_(t-1) = \beta_1 Y_(t-2) + \beta_2 Y_(t-3) + ... + \beta_0 Y_0 + \epsilon_(t-1) \quad (4)$$

That was AR and MA models respectively.

The equation of ARIMA model becomes:

$$Y_t = \alpha + \beta_1 Y_(t-1) + \beta_2 Y_(t-2) + ... + \phi_1\epsilon_(t-1) + \phi_2\epsilon_(t-2) + ... \quad (5)$$

LSTM is a recurrent neural network which is capable of learning order dependence in sequence prediction problems. It is trained using Back-propagation through Time. We can use it to process, predict and classify on the basis of time series data.

LSTM networks have memory blocks instead of neurons which are connected through layers. Each block has components and a memory for recent sequences. A block also contains gates that manage it's state and output. Every gate within a block uses the sigmoid activation units to regulate the input sequences. It makes the change of state and addition of input information flowing through the block conditional.

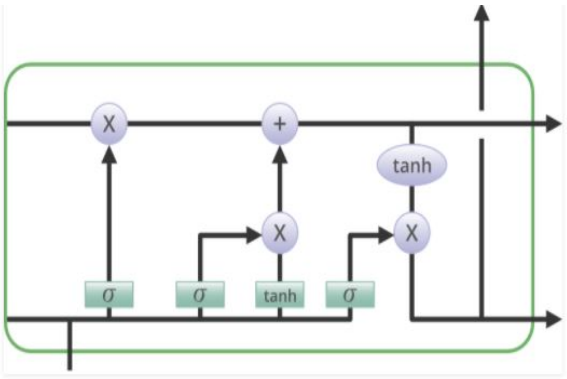

Fig. 1. Long short-term memory

There are three sorts of gates within a unit:

- Forget Gate: this gate conditionally decides unnecessary information to remove from the block.
- Input Gate: conditionally decides the necessity of the information from the input to update the memory state.
- Output Gate: conditionally decides the output depending on input and also the memory of the block. The gates have weights which are learned during the training procedure.

D. Implementation

Our dataset is time-series. It contains the data per day starting from 08 March, 2020. By plotting the dataset for different cases we get,

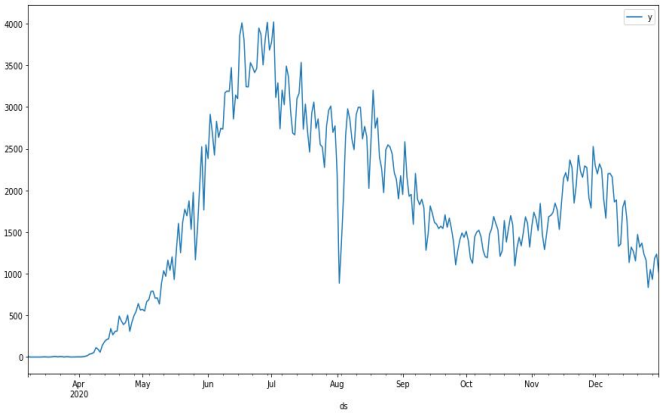

Fig. 2. Time-series plot for detected case

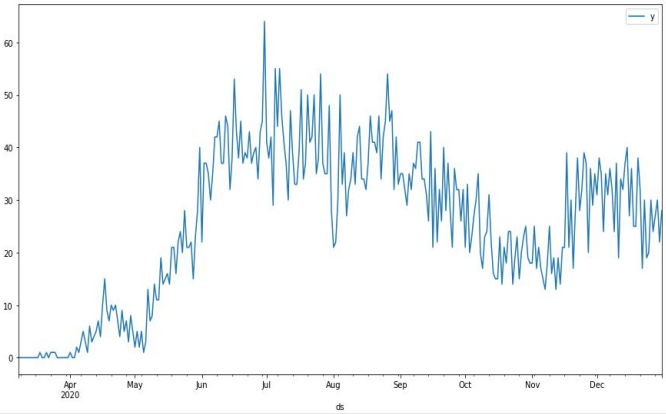

Fig. 3. Time-series plot for death case

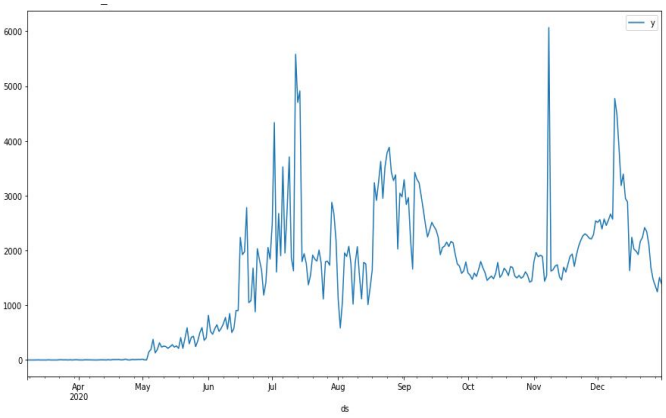

Fig. 4. Time-series plot for recovery case

A computer with Microsoft Windows 10 Pro was used for the experiment. It has the following specifications: Intel(R)

Core(TM) i5-5200U CPU @ 2.20GHz, 2201 Mhz, 2 Core(s), 4 Logical Processor(s), 4 GB of RAM, and 1 TB of hard disk. All of our codes were implemented in Google Colaboratory (”Colab” in short).

Necessary libraries for FB Prophet, ARIMA, SARIMAX, LSTM are defined. After some preprocessing (where it is necessary) methods for the respective algorithms are built.

For FB Prophet, cross validation is done for the cases (detected, death, recovery) considering the parameters (initial=’180 days’, period=’25 days’, horizon = ’50 days’). Then future date frames for 365 days are created respectively. Based on the future date frames, detected, death and recovery cases are detected where we got different components for the cases separately.

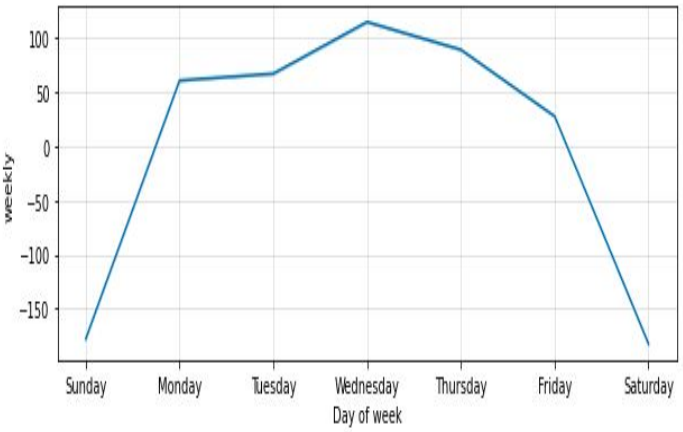

Fig. 7. Components for detected case (weekly)

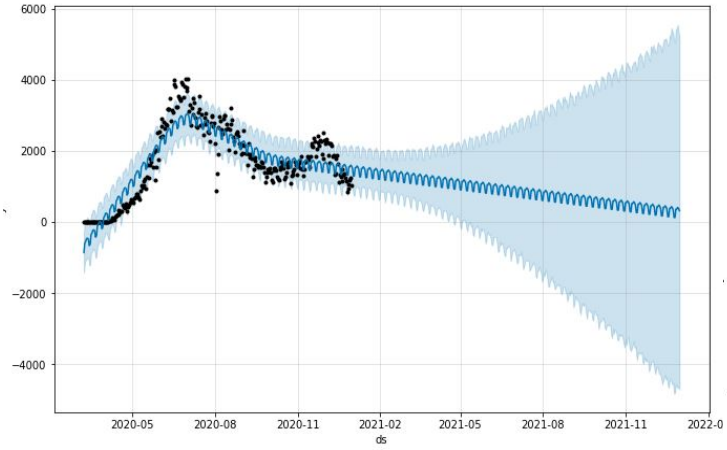

Fig. 5. Predicted Projection for detected case

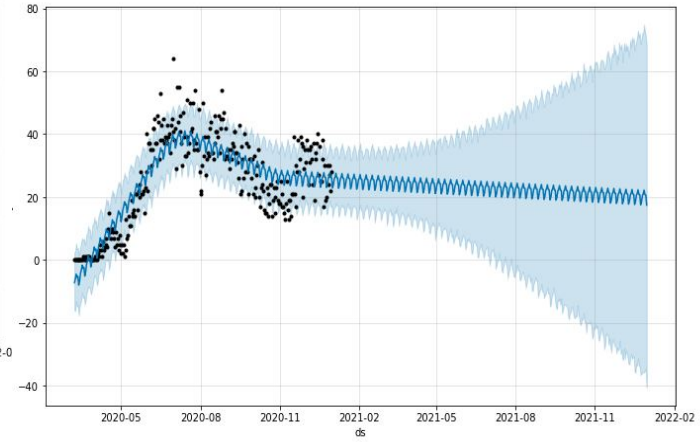

Fig. 8. Predicted Projection for death case

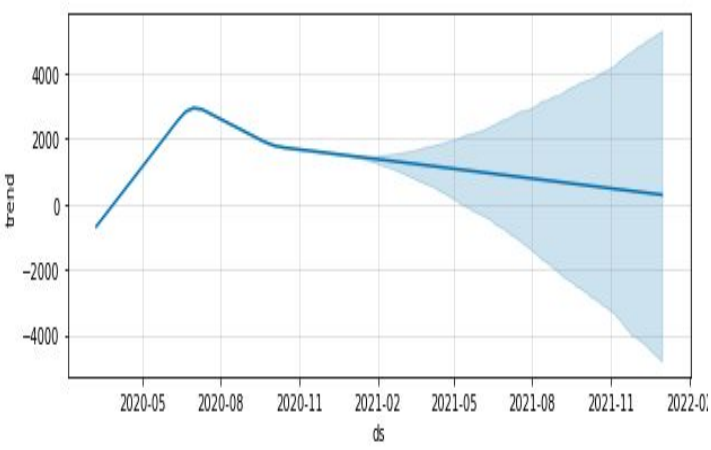

Fig. 6. Components for detected case (trend)

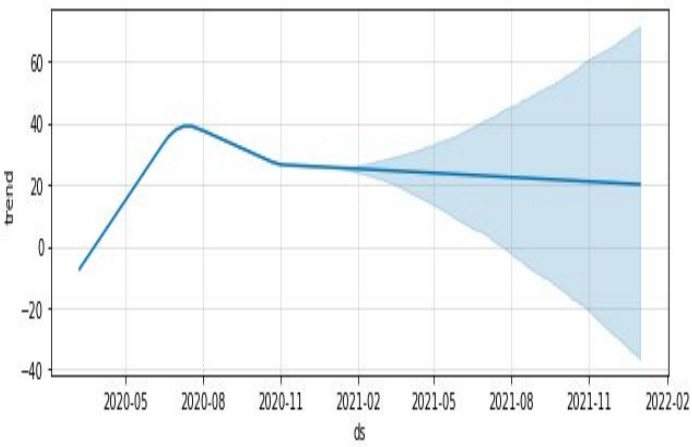

Fig. 9. Components for death case (trend)

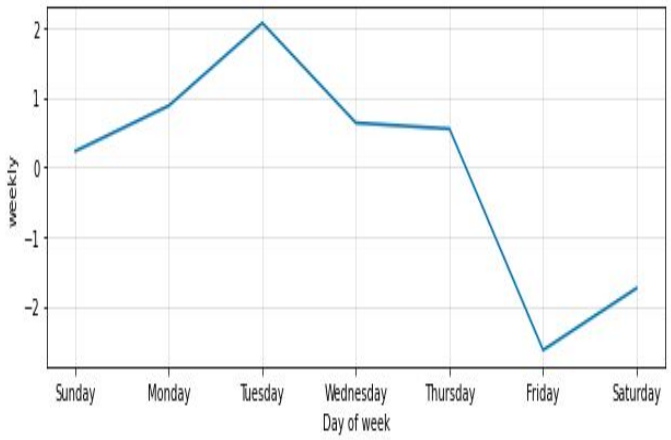

Fig. 10. Components for death case (weekly)

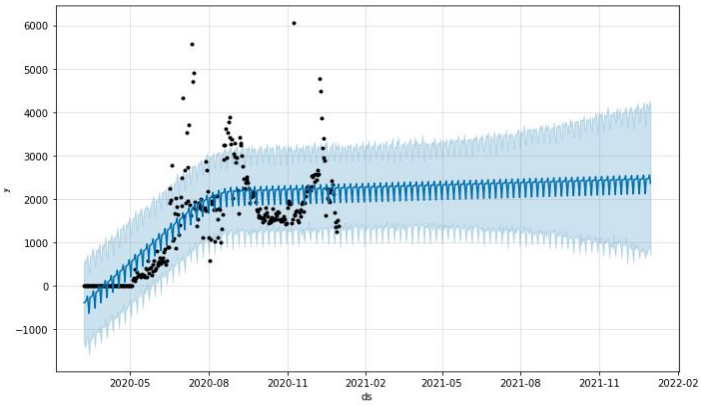

Fig. 11. Predicted Projection for recovery case

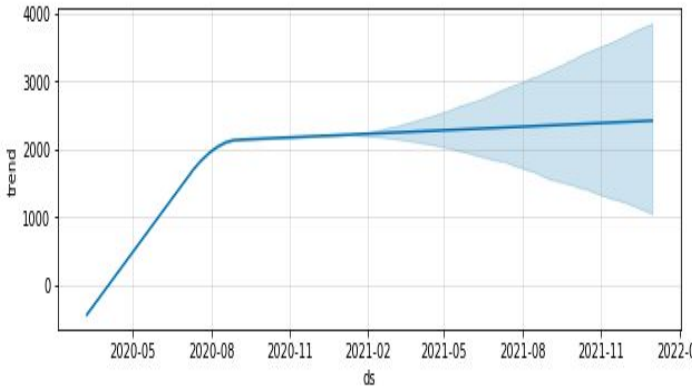

Fig. 12. Components for recovery case (trend)

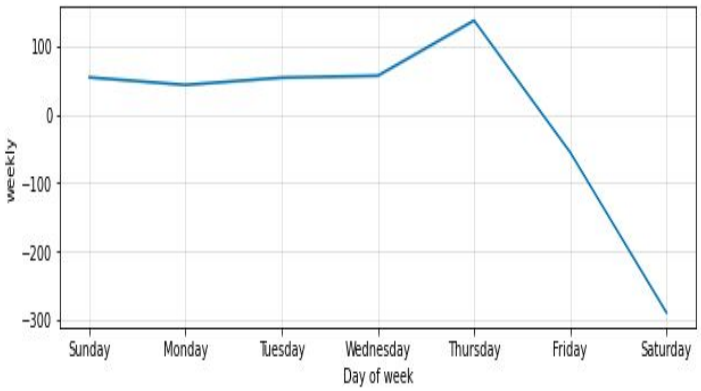

Fig. 13. Components for recovery case (weekly)

For ARIMA/SARIMAX, we first checked whether the time-series dataset was stationary or not. To check this, we followed two ways: Rolling Statistics (Plotting the rolling mean and rolling standard deviation) and Augmented Dickey-Fuller (ADF) Test. We concluded as non stationary for all the cases.

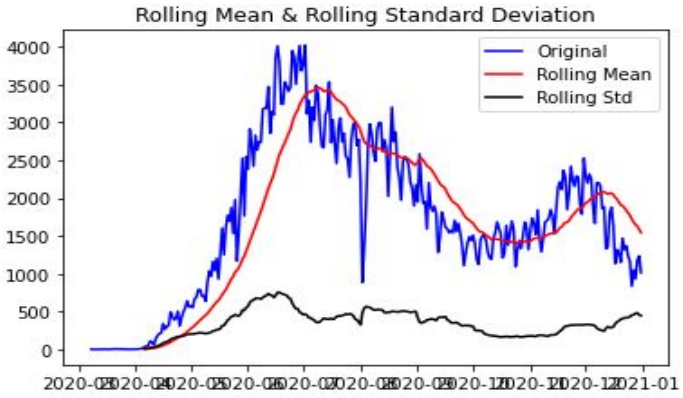

Fig. 14. Rolling Statistics for case 'Detected'

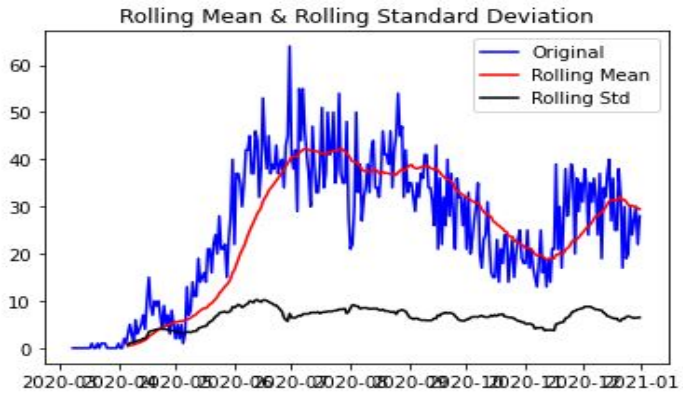

Fig. 15. Rolling Statistics for case 'Death'

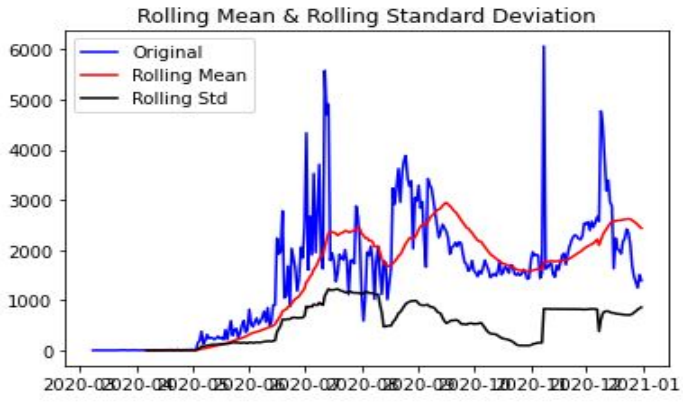

Fig. 16. Rolling Statistics for case 'Recovery'

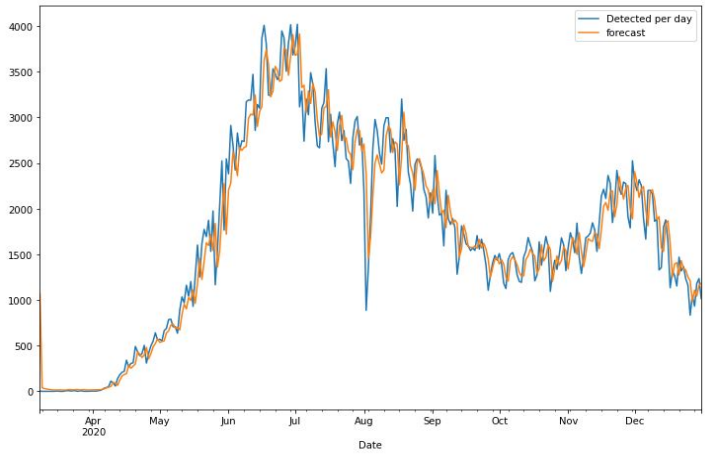

Fig. 17. Actual and forecast plot for case 'Detected' using ARIMA

ADF test result for case 'Detected':

ADF Test Statistic : -2.297858995339325

p-value : 0.17265941796374096

Lags Used : 16

Number of Observations Used : 282

Weak evidence against null hypothesis, time series has a unit root, indicating it is non-stationary.

ADF test result for case 'Death':

ADF Test Statistic : -1.9494303581925798

p-value : 0.30917641615394276

Lags Used : 13

Number of Observations Used : 285

Weak evidence against null hypothesis, time series has a unit root, indicating it is non-stationary.

ADF test result for case 'Recovery':

ADF Test Statistic : -1.9341781456038354

p-value : 0.31615733124816464

Lags Used : 9

Number of Observations Used : 289

Weak evidence against null hypothesis, time series has a unit root, indicating it is non-stationary.

Using difference, we convert these non-stationary to stationary. Then, '$auto_arima'$ function from the 'pmdarima' library helps us to identify the most optimal parameters for ARIMA models and SARIMAX models for all three cases and returned the fitted ARIMA models and SARIMAX models. Then based on these models, we predicted the results for our desired cases.

Plots for ARIMA,

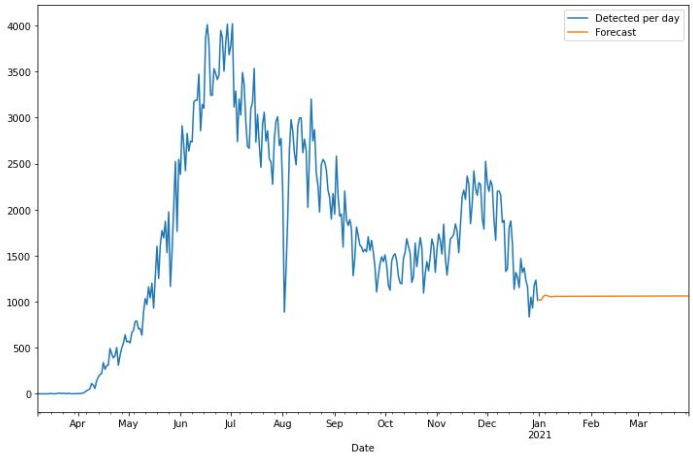

Fig. 18. Future Prediction for case 'Detected' using ARIMA

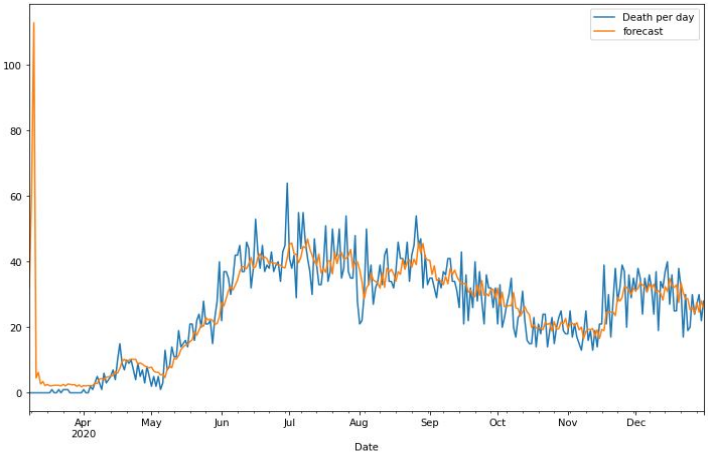

Fig. 19. Actual and forecast plot for case 'Death' using ARIMA

Plots for SARIMAX,

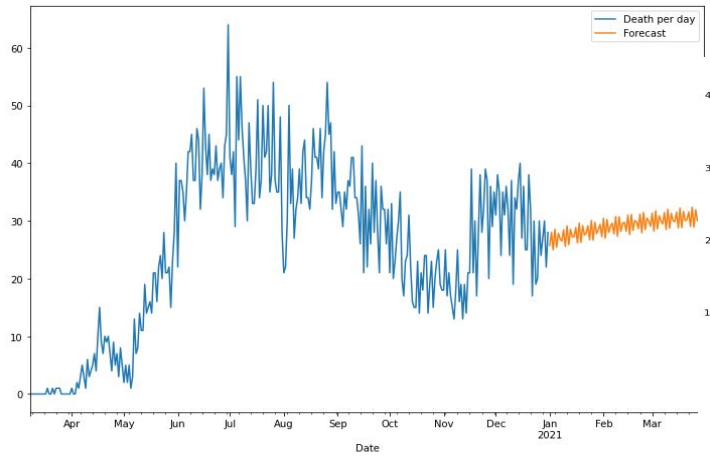

Fig. 20. Future Prediction for case 'Death' using ARIMA

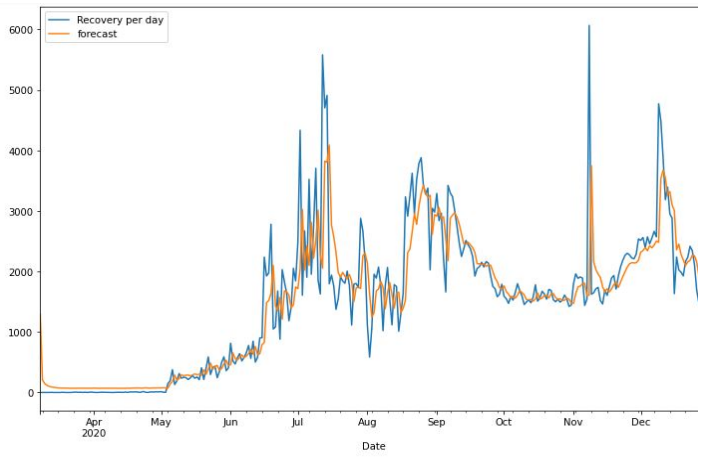

Fig. 21. Actual and forecast plot for case 'Recovery' using ARIMA

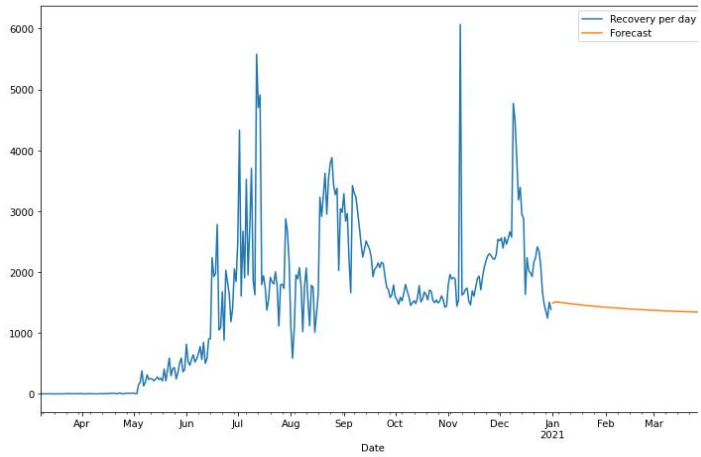

Fig. 22. Future Prediction for case 'Recovery' using ARIMA

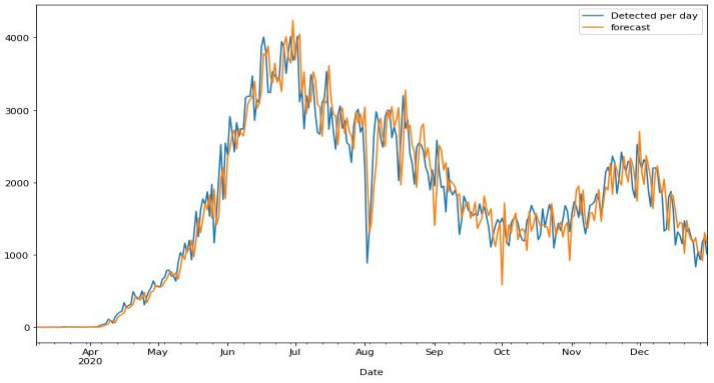

Fig. 23. Actual and forecast plot for case 'Detected' using SARIMAX

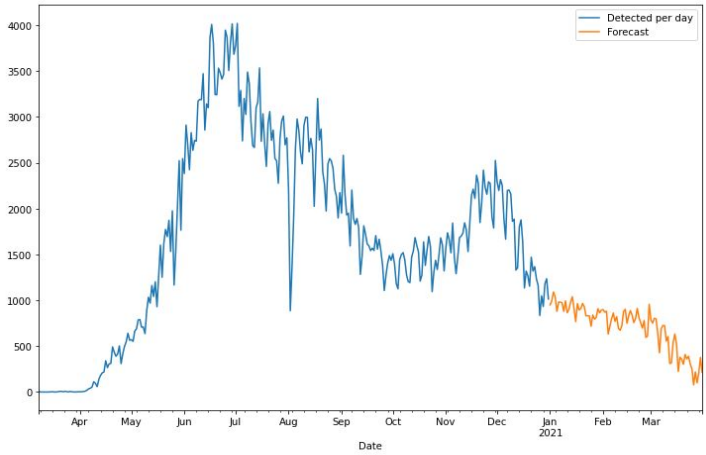

Fig. 24. Future Prediction for case 'Detected' using SARIMAX

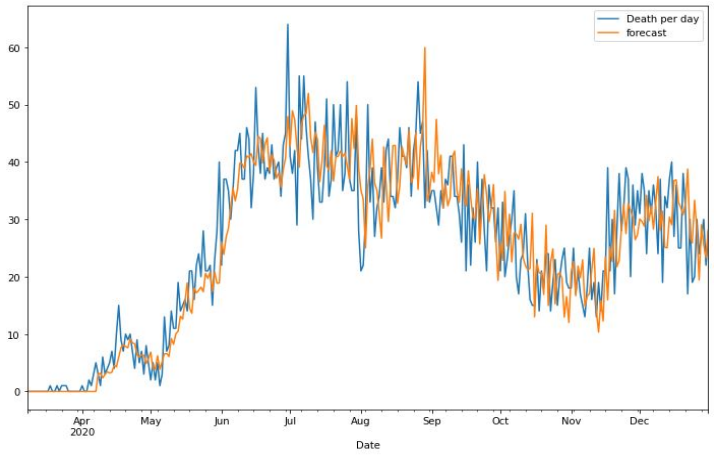

Fig. 25. Actual and forecast plot for case 'Death' using SARIMAX

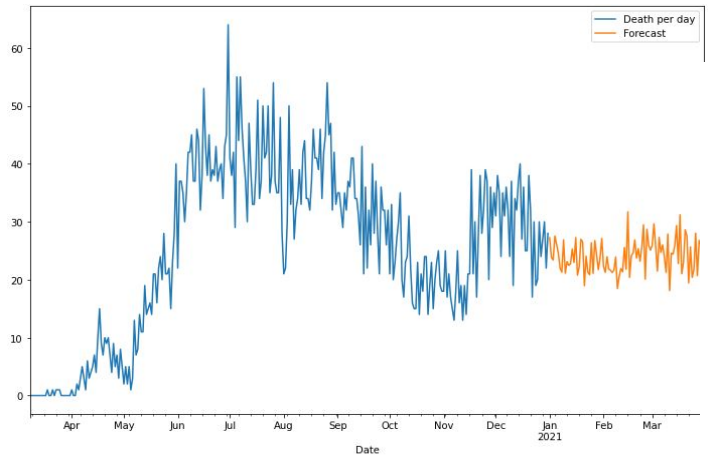

Fig. 26. Future Prediction for case 'Death' using SARIMAX

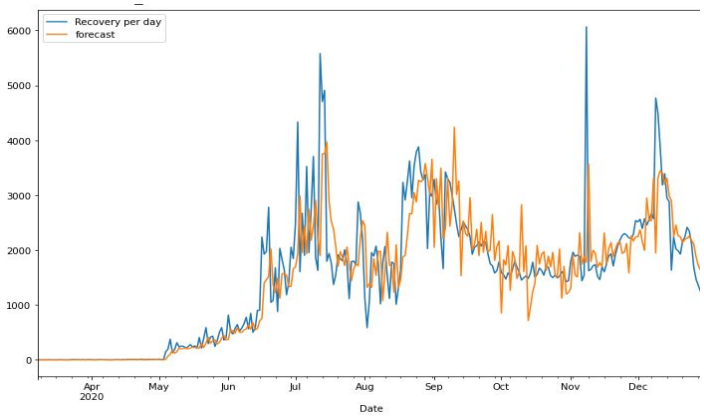

Fig. 27. Actual and forecast plot for case 'Recovery' using SARIMAX

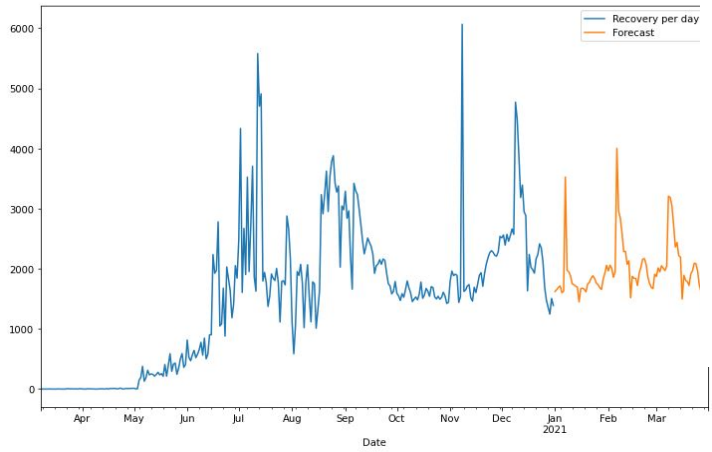

Fig. 28. Future Prediction for case 'Recovery' using SARIMAX

For LSTM, prediction plots for three cases are as following:

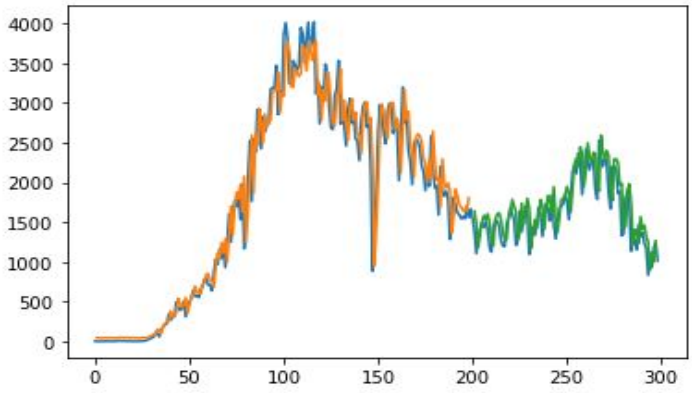

Fig. 29. Prediction for case 'Detected' using LSTM

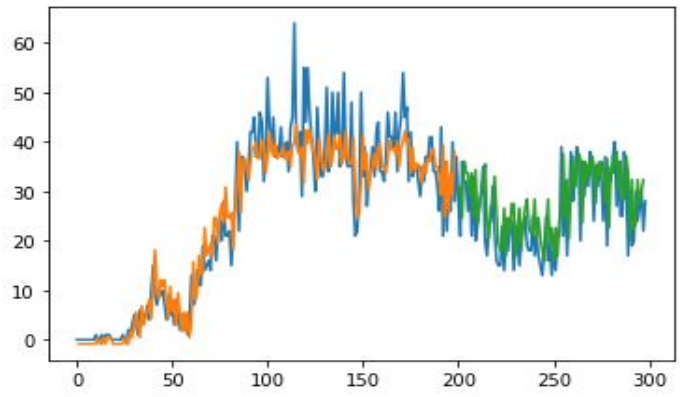

Fig. 30. Prediction for case 'Death' using LSTM

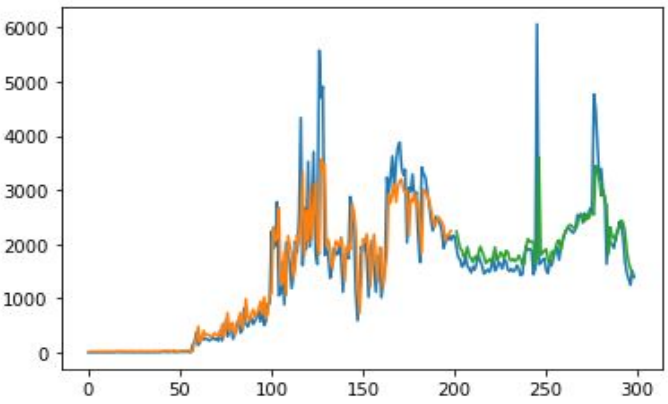

Fig. 31. Prediction for case 'Recovery' using LSTM

To assess the reliability of the proposed system, we adopted the evaluation metrics Root Mean Squared Error (RMSE) for each of the algorithms. RMSE is the most popular evaluation

metric used in regression problems. It follows an assumption that error are unbiased and follow a normal distribution.

This metric measures the differences between the actual X_i and the predicted $(\hat{X_i})$numbers of COVID-19 confirmations, recoveries, and deaths The main advantage of RMSE is that it penalises large prediction errors. RMSE was used to compare the prediction errors of the three prediction algorithms. It is defined as follows:

$$RMSE = \sqrt{\sum_{i=1}^{N}(X_i - \hat{X}_i)^2}$$

where, N is Total Number of Observations.

Result Discussion

This study provided a forecasting analysis of COVID-19 detected, recovery, and death in Bangladesh. The forecasting is done by applying four well-known different time series forecasting algorithms called FB Prophet, ARIMA, SARIMAX, LSTM. Based on the study results, the following conclusions were drawn:

- LSTM $(RMSE = 249.89)$ delivered the best performance for forecasting COVID-19 detected compared to FB Prophet $(RMSE = 991.057)$, ARIMA $(RMSE = 267.2488)$, SARIMAX $(RMSE = 307.5994)$.
- LSTM $(RMSE = 597.00)$ delivered the best performance for forecasting COVID-19 recovery compared to FB Prophet $(RMSE = 1246.54)$, ARIMA $(RMSE = 582.4039)$, SARIMAX $(RMSE = 639.2697)$.
- SARIMAX $(RMSE = 6.9687)$ delivered the best performance for forecasting COVID-19 death compared to FB Prophet $(RMSE = 13.8631)$, ARIMA $(RMSE = 9.7885)$, LSTM $(RMSE = 8.02)$.

We can conclude saying that

For case 'Detected', LSTM performs better, then ARIMA, then SARIMAX and lastly FB Propet.

For case 'Recovery', LSTM performs better, then SARIMAX, then FB Prophet and lastly ARIMA.

For case 'Death', SARIMAX performs better, then LSTM, then ARIMA and lastly FB Prophet.

Conclusion

This virus is spreading like a destruction of a giant and from the statistics it looks like it will take a while to stop its spreading. This research work is expected to shade light on Covid-19 prediction models for future researchers working with the machine learning techniques.

References

[1] Qamruzzaman, Md., "COVID-19 Impact on SMEs in Bangladesh: An Investigation of What They Are Experiencing and How They Are Managing?", SSRN, (July 17, 2020).

[2] Haque, A. (2020), "The COVID-19 pandemic and the public health challenges in Bangladesh: a commentary", Journal of Health Research, Vol. 34 No. 6, pp. 563-567

[3] Zaman, Mahruf Rahman, Arafat Khan, Asaduzzaman. (2020). "The COVID-19 Pandemic and the survival of Bangladesh: A mixed analysis."

[4] Begum, Momotaj Farid, Shaikh Barua, Swarup Alam, Mohammad. (2020). "COVID-19 and Bangladesh: Socio-Economic Analysis Towards the Future Correspondence." 10.20944/preprints202004.0458.v1.

[5] Haque, A K M Bahalul Pranto, Tahmid. (2020). COVID-19 (SARS-CoV-2) Outbreak in Bangladesh: Situation according to Recent Data Analysis using COVID-19DataSet for Bangladesh. 10.13140/RG.2.2.31876.76161/1.

[6] https://corona.gov.bd/press-release

[7] https://covidtracker.bsg.ox.ac.uk/

[8] https://www.worldometers.info/coronavirus/country/bangladesh/

[9] https://bd.ambafrance.org/-COVID-19-376-

[10] http://dashboard.dghs.gov.bd/webportal/pages/covid19.php

[11] https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-cases

[12] https://www.facebook.com/IedcrCOVID-19-Control-Room-104339737849330/

[13] http://unb.com.bd/m/search?search=covid

[14] https://www.dhakatribune.com/search?query=covid