Covid 19 Prediction

Rutik Sanjay Sangle

016007589
San Jose State University
rutiksanjay.sangle@sjsu.edu

Venkata Sai Sri Batchu

016118557
San Jose State University
venkatasaisri.batchu@sjsu.edu

Suhas Byrapuneni

016118596
San Jose State University
suhas.byrapuneni@sjsu.edu

Nhat Trinh

011227645
San Jose State University
nhat.trinh@sjsu.edu

***Abstract*—The coronavirus (COVID-19) epidemic has caused millions of infections and lakhs of fatalities worldwide. Identification of potential infection cases and the rate of virus propagation is crucial for early healthcare service planning to prevent fatalities. Standard models have demonstrated limited accuracy for long-term prediction due to a high amount of uncertainty and a lack of crucial data. In this paper, we are going to deal with a large dataset that includes information regarding every country in the world that has reported COVID deaths for predicting the next outbreak.**

Keywords—COVID-19, LSTM, ARIMA, XGBoost, Machine Learning

I. Introduction

Since it first appeared in China in December 2019, the novel coronavirus (COVID-19) has infected millions of people globally. When people weren't aware of the virus, COVID-19 began to spread from one person to another; over time, it expanded slowly to nearly every country and eventually became a pandemic. The WHO and other health organizations identify coronaviruses as a group of viruses whose symptoms vary from the common cold to more serious illnesses. The WHO chose the name COVID-19 in order to avoid stigmatizing associations with the virus's origins in terms of communities, geography, or animals.

Numerous investigations are carried out right away to comprehend the properties of this coronavirus to understand the outbreak. It is asserted that direct contact and respiratory droplets are the two main methods of SARS-CoV-2 transmission from person to person.

Modern machine learning techniques for epidemic prediction modeling highlight two significant areas of research that need to be addressed by machine learning. To start, epidemic time series prediction has advanced, and SIR and SEIR model improvements have followed. Machine learning can undoubtedly help in light of the shortcomings of the SIR and SEIR that are currently in use. This study advances COVID-19 time series prediction.

In this project, we are trying to use three different models and find out which model is best for time-series prediction. We are using LSTM (Long Short-Term Memory), XGBoost (Extreme Gradient Boosting), and ARIMA (Autoregressive Integrated Moving Average) models for our Covid-19 dataset.

Our first model is LSTM which is an artificial neural network that has feedback connections. The LSTM has both “long-term memory” and “short-term memory” in which the connection weights and biases in the network change once per batch of training. The architecture goal is to establish a short-term memory for RNN that can last thousands of timesteps.

Secondly, we have XGBoost which provides a parallel tree boosting and the leading machine learning model for regression and many other problems.

Our last model is ARIMA which stands for Autoregressive Integrated Moving Average. This model works with time series data to either understand the data or predict the future direction based on the past value. This model simply assumes that the future will resemble the past and gives a prediction that might be false in some unconditional environments.

II. Methods

A. LSTM

Long short-term memory networks or LSTM are widely used for time-series forecasting applications. LSTM models can be used for univariate time series forecasting where only a single series of observations and a model are needed to learn from the previous data and predict the future data in the series [1]. LSTM is a part of Recurrent Neural Networks (RNN). LSTM can be used for short-term as well as long-term predictions [2]. The figure below shows the architecture of the LSTM model:

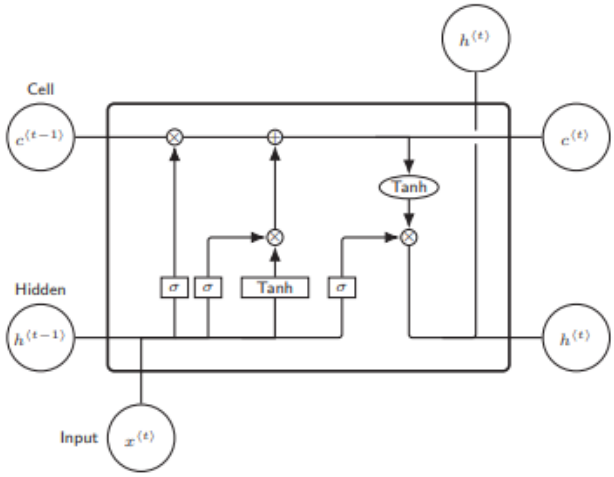

Figure 1: LSTM Architecture [2]

The LSTM network has some gates that track the sequence information. Mainly, there are three gates, the input gate i_t, the forget gate f_t, and the output gate o_t. The role of the input gate is used to select the data that is going to be stored for the next state. The role of the forget gate is to select the data that is not going to be stored. The output gate is used to choose the information that has to be sent to the output. The symbol c is used to represent the current state while $c\tilde{}$ is used to represent the next state.

For the calculation of i_t, f_t, o_t, and $c\tilde{}_t$, the following equations are used:

$$f_t = \sigma\,(w_f \times [h_t{-}1, x_t] + b_f)$$
$$i_t = \sigma\,(w_i \times [h_t{-}1, x_t] + b_i)$$
$$o_t = \sigma\,(w_o x_t + w_o)$$
$$c\tilde{}_t = \tan h\,(w_c \times [h_t{-}1, x_t] + b_c)$$

where, W represents the weight matrix, b_i, b_f, b_o, and $b_{c\text{-}}$ represent the bias vectors. x_t is the input, h_t -1 is the calculated LSTM output and σ is the sigmoid activation function. [2]

Calculation of the state of the cell is done by:

$$c_t = f_t \cdot c_t{-}1 + i_t \cdot c\tilde{}_t$$

where the · represents the Hadamard product.

For the calculation of the LSTM network output h_t,

$$h_t = o_t \times \tan h(c_t)$$

Then finally, the output of LSTM $y\tilde{}_t$ is given by

$$y\tilde{}_t = w_y h_t$$

In this project, the LSTM model is trained using approximately 80% data and tested using 20% data. For the input data, the cumulative cases for the country India is used.
The LSTM model needs the data in a very specific manner. The observations have to be transformed into multiple series where a few observations are given as input and one observation is given as output. In this model, we have decided to use three time steps for input and the fourth time step as the output for the input. In this manner, the LSTM model learns the pattern and makes future predictions.

For example,

Data: 10, 20, 30, 40, 50, 60, 70

After Modification:

Input	Output
X: 10 20 30	Y: 40
X: 20 30 40	Y: 50
X: 30 40 50	Y: 60

After this pre-processing of data, the data is passed to the model for learning. After testing the model, we were able to achieve very good results which were plotted with the help of line plots.

B. ARIMA

"ARIMA is an acronym that stands for Autoregressive Integrated Moving Average. It is a class of models that captures a suite of different standard temporal structures in time series data" [3]. The name of the model describes its working.

AR: Autoregression – using dependent relationship and lagged observations [4]

I: Integrated – a difference of raw observations to make time-series stationary [4]

MA: Moving Average – using dependency between observation and residual error [4]

For this model, no specific pre-processing was required. We just have to pass the cumulative cases to the model. The following equation can be used to formulate the ARIMA model:

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \ldots + a_p y_{t-p} + b_1 e_{t-1} + \ldots + b_1 e_{t-q} + e_t$$

In this, y_t represents data, y_{t-1} y_{t-2} are data at past time instants. $e_{t-1,}$ e_{t-2}… are errors at past time instants and e_t stands for present error. ARMA assumes that the error is Gaussian distributed. The coefficients, $a_1, a_2 \ldots a_p$ are the AR coefficients, and $b_1, b_2, \ldots b_q$ are the MA model coefficients. The model can be validated using AIC (Akakine Information Criterion) [4].

C. XGBoost

"XGBoost is an efficient implementation of gradient boosting for classification and regression problems. It is both fast and efficient, performing well, if not the best, on a wide range of predictive modeling tasks and is a favorite among data science competition winners, such as those on Kaggle" [5]. For XGBoost, the data has to be transformed into a supervised learning problem. Along with that, walk-forward validation is used for better evaluations. The formula of the objective loss function of the XGBoost model is shown below:[6]

$$\overline{Object}(t) = \sum_{i=1}^{n} l(y_i, \hat{y}_i^t) + \Omega(f_t) + C$$

$$\hat{y}_i^t = \hat{y}_i^{t-1} + f_t(x_i)$$

$$\Omega(f_t) = \gamma T_t + \frac{1}{2}\lambda \parallel w \parallel^2$$

where X is a constant, x_i is the input vector, T_t gives the number of leaves in the tree, γ and λ are hyperparameters. y_i is the real value of sales. l(.) is the square loss function and f_t(.) is a regression tree. Along with this, according to Taylor's formula, the object function is approximately expressed as:

$$Object(t) \approx \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{t-1}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + C$$

D. *LSTM for Future values prediction*

We decided to use the LSTM model for the prediction of the future value. By making simple modifications to the existing LSTM model, we were able to predict the future values for the year 2023.

E. *XGBoost for Future Values prediction*

Along with LSTM, we also used XGBoost for the prediction of the future value. The same changes were made as LSTM to predict future values.

III. Comparisons

This section will briefly describe the results of all the models used in the project and a comparison will be done to see how each model performed.

A. *LSTM*

The below figure shows the comparison of the actual and the predicted values given by the LSTM model. The testing was done on a test set of around 230 values.

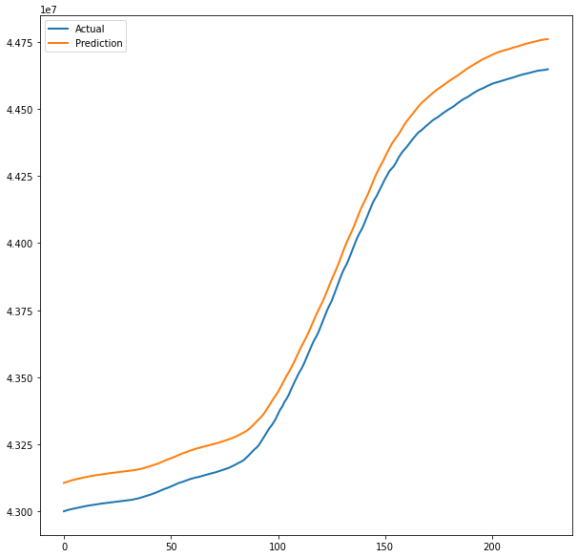

Figure 2: LSTM Result

For the LSTM model, the RMSE (Root Mean Squared Error) is 104923. This is the highest as compared to all other models.

B. *ARIMA*

The following figure shows the line plots of the actual and predicted values for the same test set of 230 values. As compared to the LSTM model, the ARIMA model is showing better performance. The difference between the lines in the plots has decreased. From this, we can say that the ARIMA model is predicting better.

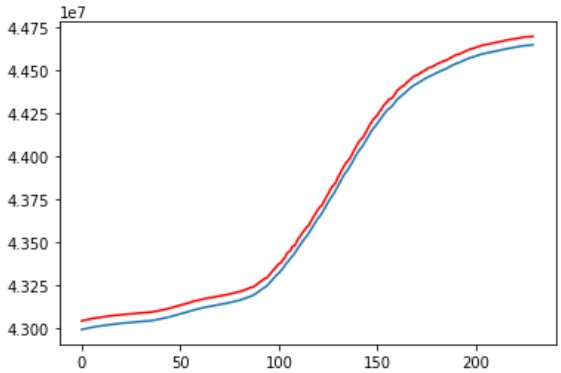

Figure 3: ARIMA Result

For the LSTM model, the RMSE (Root Mean Squared Error) is 49985. This is comparatively low than the LSTM model.

C. *XGBoost*

The below figure shows the plot for the expected and predicted values for the same set of 230 values. In this plot, the lines of predicted and actual values is overlapping. From this, we can say that the performance of XGBoost is better than LSTM and ARIMA.

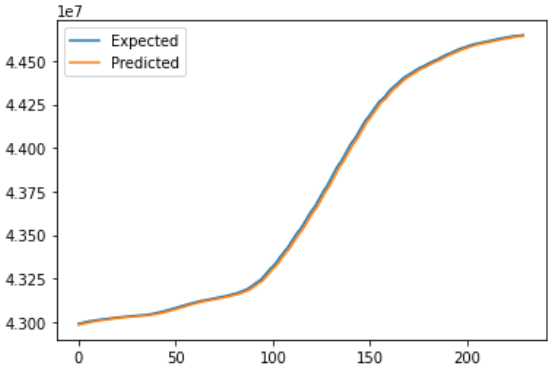

Figure 4: XGBoost Result

For the LSTM model, the RMSE (Root Mean Squared Error) is 13702. This is the lowest of all the models.

D. *Comparison of RMSE of all models*

Method	RMSE
LSTM	104923
ARIMA	49985
XGBOOST	13702

Considering the RMSE values of all the models, we decided to move forward with the LSTM and XGBoost models for future predictions.

E. *LSTM Model for Future Values Prediction*

The following image shows the line plots showing the past values (blue line) and the prediction of the future values (yellow line). The values are predicted for the year 2023. As per the predictions, we can see a slight incline in the values which shows that there is less chance of a next COVID wave or spike.

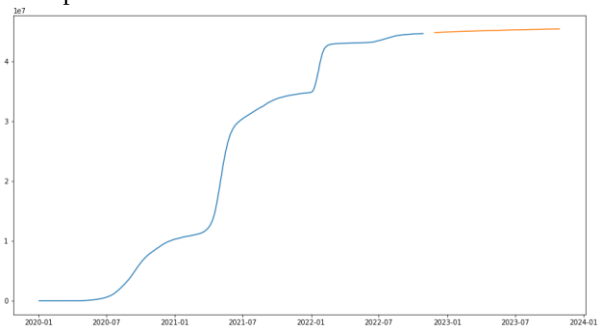

Figure 5: LSTM with future values prediction

F. *XGBoost for Future Values Prediction*

The below image shows the results of the XGBoost model with future values prediction. As this model was the best-performing model of all three, we can rely more on this result. As per the predictions, we can see that there is a decrease in cumulative cases. So, we can conclude that there is very less chance of a next COVID wave or spike.

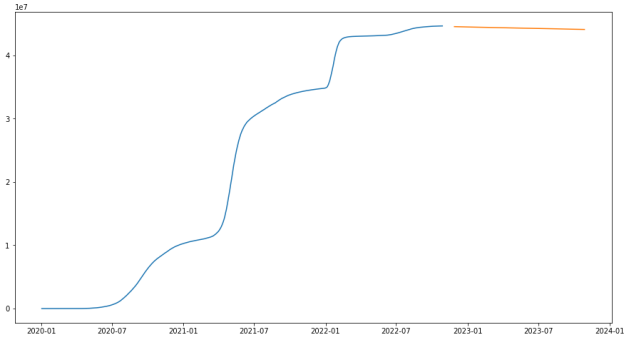

Figure 6: XGBoost with future values prediction

CONCLUSION

The world has witnessed many COVID cases and some COVID waves or spikes where there was a sudden rise in the number of cases. This project is focused on building machine learning models that can predict the next wave or spike if there will be any. Different models have performed differently for the same set of data. Using models such as LSTM, ARIMA, and XGBoost, we have concluded that there is very less chance of a next COVID wave or spike. The future work for this project includes the prediction of such pandemics using similar data along with multiple external factors such as weather, location, age, BMI, etc.

REFERENCES

[1] J Brownlee “How to Develop LSTM Models for Time Series Forecasting” machinelearningmastery.com https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/ (accessed on 01 Dec 2022)

[2] K. Moharm, M. Eltahan, and E. Elsaadany, "Wind Speed Forecast using LSTM and Bi-LSTM Algorithms over Gabal El-Zayt Wind Farm," *2020 International Conference on Smart Grids and Energy Systems (SGES)*, 2020, pp. 922-927, doi: 10.1109/SGES51519.2020.00169.

[3] J Brownlee “How to Create an ARIMA Model for Time Series Forecasting in Python” machinelearningmastery.com https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/ (accessed on 01 Dec 2022)

[4] C. Narendra Babu and B. Eswara Reddy, "Predictive data mining on Average Global Temperature using variants of ARIMA models," *IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012)*, 2012, pp. 256-260.

[5] J Brownlee “How to Use XGBoost for Time Series Forecasting” machinelearningmastery.com https://machinelearningmastery.com/xgboost-for-time-series-forecasting/ (accessed on 01 Dec 2022)

[6] Y. Niu, "Walmart Sales Forecasting using XGBoost algorithm and Feature engineering," *2020 International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*, 2020, pp. 458-461, doi: 10.1109/ICBASE51474.2020.00103.

[7] A Baranovskij “Time-Series Prediction Beyond Test Data” towardsdatascience.com https://towardsdatascience.com/time-series-prediction-beyond-test-data-3f4625019fd9 (accessed on 01 Dec 2022)