COVID-19 Forecasting

1st Merlin Cherian
College of Computing and Informatics
Drexel University
mc3538@drexel.edu

2nd Mruga Shah
College of Computing and Informatics
Drexel University
mts339@drexel.edu

***Abstract*—More than a third of the world's population entered into lockdown with the spread of a novel coronavirus COVID-19. It became crucial to understand the impact of the spread of the virus which required forecasting of the trend of the confirmed cases as well as analyze the number of deaths and recoveries. This paper outlines the use of LSTM and RNN to forecast the cases of COVID-19 in the US. Underestimating the spread of this pandemic is far to risky. This paper makes use of existing ML libraries to forecast cases for 30 days into the future to help in planning and decision making.**

I. Background

COVID-19 is a new family of virus that emerged in China in late 2019. On March 11 2020, WHO declared it a pandemic 5.35M cases, 343K deaths. Since then, the number of people getting infected with the virus has been increasing exponentially. Only a few countries have been able to brought it under control. The way of life is changing and the future is uncertain. Institutions and governments need to plan for the coming months on how to reopen without causing further expansion of the pandemic.

Predicting the number of upcoming cases in the next 30 days would help the decision makers in preparing for the near future. Observing changes from interventions such as easing restrictions, will help them in planning for resources that will allow the situation to remain in control. In our project, we use the data collected since January 2020, to predict the values for 30 days in future. This data that is being collected by John Hopkins and made publicly available. We use the number of confirmed COVID-19 cases, recovered cases and deaths in the US in the last few months to predict those numbers in the upcoming month.

We use deep learning to train a model that will forecast the values in future. Specifically, we use the Long-Short Term Memory(LSTM) model of Recurrent Neural Networks to train the system. Both LSTM and RNN are explained in detail in section "Neural Network Architecture".

Through this project, we help in understanding the most likely outcomes(best/worst) and help the government and leaders make important decisions about the mobility of the people.

II. Related Work

Randerson develops an LSTM network in Python using Keras and other deep learning libraries to demonstrate time series-prediction on International Air Passengers problem. The networks maintain state memory across long sequences [8].

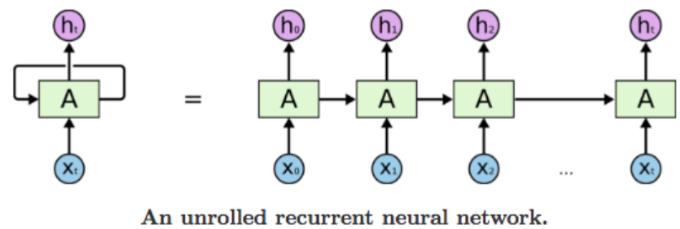

Fig. 1. Recurrent Neural Network

In their research work, Qiu, Wang and Zhou, forecast stock prices with long-short term memory neural network based on attention mechanism. Based on LSTM and an attention mechanism, a wavelet transform is used to de-noise historical stock data, extract and train its features, and establish the prediction model of a stock price [1].

III. Neural Network Architecture:

In this section we introduce in detail the concepts behind the model we use for forecasting. We also explain in detail the architecture that is used for our system.

RNNs: Recurrent Neural Network is a generalization of feed forward neural network that has an internal memory [5]. It remembers the past and it's decisions are influenced by what it has learnt from the past [4]. As a Neural Network classifier trains and learns to classify, similarly a Recurrent Neural Network trains but also remembers from prior inputs while predicting the output. RNNs can be applied to tasks such as unsegmented, connected handwriting recognition or speech recognition. Figure 1 shows a RNN network.

LSTMs: Long short-term memory is an artificial recurrent neural network architecture used in the field of deep learning. Unlike standard feed forward neural networks, LSTM has feedback connections. It can not only process single data points, but also entire sequences of data. They are a modified version of recurrent neural networks, which makes it easier to remember past data in memory.The vanishing gradient problem of RNN is resolved by LSTM. LSTM is well-suited to classify, process and predict time series data [5]. Figure 2 shows the LSTM model. In LSTMs, the top line running through the diagram is the cell state which runs down the entire chain of the cells and the information can very easily be passed through it unchanged. The LSTM does have the ability to

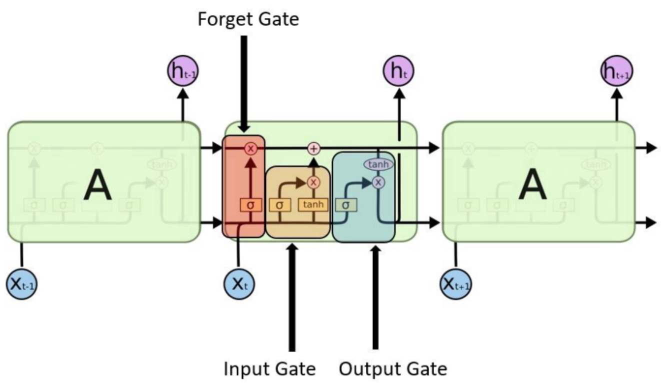

Fig. 2. LSTM Model

remove or add information to the cell state, carefully regulated by structures called gates. Gates are a way to optionally let information through. Additionally, there are three gates in an LSTM model: input, output and forget gate. They are composed out of a sigmoid neural net layer and a point wise multiplication operation. The last part is the transfer of information from one hidden layer to another.

In the following subsections we explain the architecture in minute detail by providing the code/values used while building the model.

A. Creating LSTM Model

There were a few packages used: Sequential for initializing neural network, LSTM to add a LSTM layer, Dropout layers to prevent over-fitting and dense to add a densly connected neural network layer.
The LSTM layers uses the following arguments: units to define the dimensionality of the output. In order to stack the LSTM layers so the subsequent LSTM layer has a 3D input return-sequences=True flag is used. The input_shape remains the same as the training data set.
Specifying Dropout(n) will mean that (n*100)% layers will be dropped. The Dense Unit specifies an output of 1 unit. The model is compiled using 'adam' optimizer and uses the 'mean_squared_error' loss function. The model is then fit to run for 100 epochs with a batch size of 1.

B. Training and Testing the model

Once the LSTM model is created, it is trained using scaled data for certain number of epochs/iterations (we used 100 iterations) using the model.fit() function from the Keras library. After the model is trained, the model is tested against 20% of data. To obtain the predictions against the trained model, model.predict() function is used. Once the predictions are obtained, the root mean squared error is calculated and the graph of the actual data and testing data is plotted to check whether the trained model is good enough to predict the values in future.

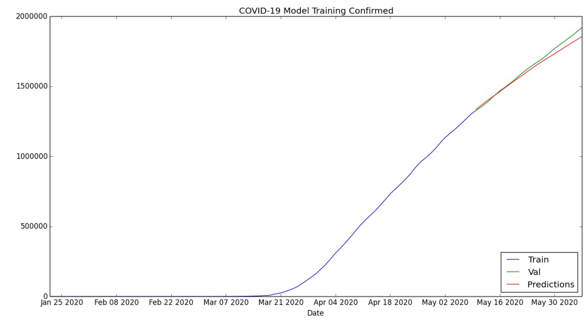

Fig. 3. Training and Testing the model for Confirmed cases

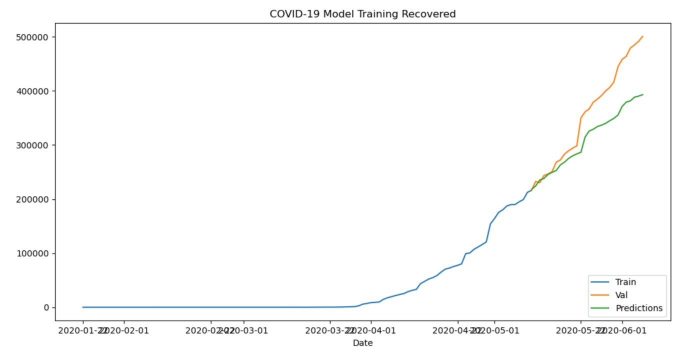

Fig. 4. Training and Testing the model for Recovered cases

C. Forecasting in the Future

Using the trained model values are forecasted in the future, for upcoming 30 days. Here, the Time Series Generator from Keras library is used [2]. Time Series Generator is used to automatically transform a uni-variate or multivariate time series data set into a supervised learning problem.

IV. Evaluation:

A. Data

The data is publicly available on the John Hopkins CSSEGIS GitHub page [3] in csv format. The files on the John Hopkins page are updated daily for 267 countries around the world. The data set is split into three groups: Confirmed, Deaths and Recovered. We use US data for our project.

B. Training the LSTM Model

Using the data, we were successfully able to train three different models for the three groups of data. The Figures 3, 4 and 5 show how accurately the models train for each of the data group. The blue and the orange lines on the graphs represent the actual data till June 6, 2020 while the green line represents the predictions made using the testing data for the month of May.

C. Predicting the Future

Using the data, we were successfully able to train our model and project the values into future. The Figures 6, 7 and 8

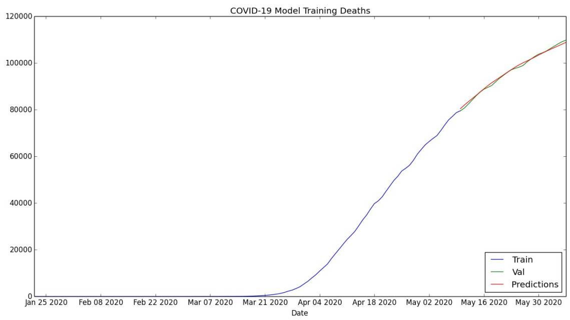

Fig. 5. Training and Testing the model for number of Deaths

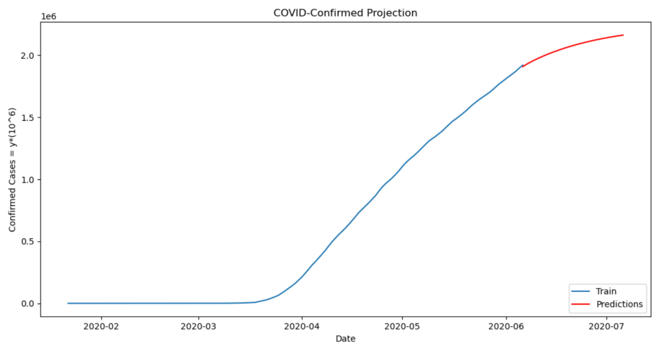

Fig. 6. Forecasting number of Confirmed cases in the next 30 days

show the predictions that our trained models make for the three groups of data. The blue line on the graph represents the complete data set (i.e. data from mid-January till June 6,2020) obtained from John Hopkins page while, the orange line represents the data that has been forecasted for next 30 days using the trained model.

V. Conclusions

Providing people with timely information is crucial during a pandemic. Although the growth rate is not as high as the initial weeks, the situation still looks dangerous in the US and reflects exponential growth. The projected confirmed cases are reaching $2.2 * 10^6$ and the deaths are reaching to a high of

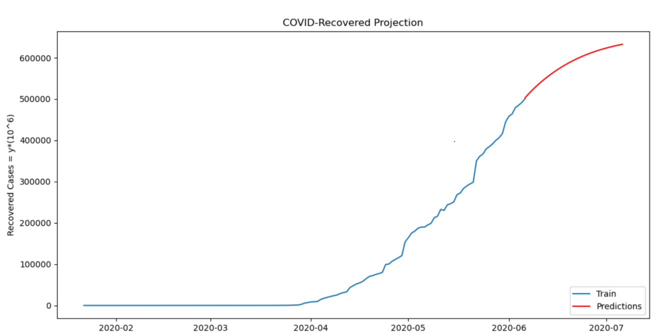

Fig. 7. Forecasting the number of Recoveries in the next 30 days

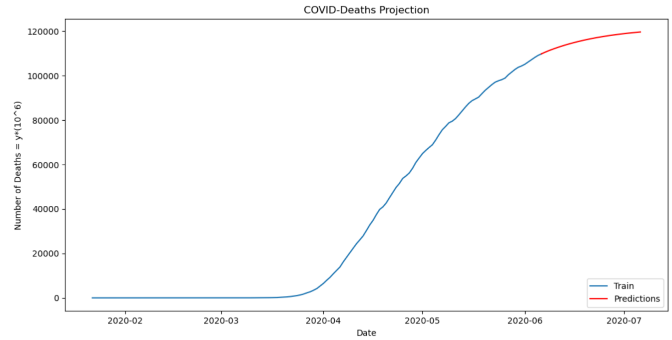

Fig. 8. Forecasting number of Deaths in the next 30 days

118024 in the next 30 days. If social distancing is relaxed, it would be difficult to contain the outbreak of the disease.

VI. Future Work/Extensions

The model can be extended to include the constraints such as government rules of lockdown and reopening to test for their effectiveness. We can also predict for multiple nations in real time and update the graphs daily. It is also possible to observe the differences of how the plots change according to the restrictions in different countries. To make the system more efficient, instead of LSTMs we could experiment with other models such as ResNet and Attention [6].

References

[1] Qiu, J., Wang, B. and Zhou, C., 2020. Forecasting Stock Prices With Long-Short Term Memory Neural Network Based On Attention Mechanism.

[2] Brownlee, J., 2020. How To Use The Timeseriesgenerator For Time Series Forecasting In Keras. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/how-to-use-the-timeseriesgenerator-for-time-series-forecasting-in-keras/.

[3] GitHub. 2020. Cssegisanddata/COVID-19. [online] Available at: https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series.

[4] Medium. 2020. Recurrent Neural Networks. [online] Available at: https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce.

[5] Medium. 2020. Understanding RNN And LSTM. [online] Available at: https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e.

[6] Medium. 2020. The Fall Of RNN / LSTM. [online] Available at: https://towardsdatascience.com/the-fall-of-rnn-lstm-2d1594c74ce0.

[7] Brownlee, J., 2020. Time Series Prediction With LSTM Recurrent Neural Networks In Python With Keras. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/.

[8] ”Stock Price Prediction Using Python Machine Learning” YouTube, Dec. 21, 2019 [Video file]. Available: https://www.youtube.com/watch?v=QIUxPv5PJOYt=2410s.