COVID-19 Time Series Forecasting of Daily Cases, Deaths Caused and Recovered Cases using Long Short Term Memory Networks

Suraj Bodapati
Employee, JP Morgan Chase & Co.
Hyderabad,India
surajbodapati97@gmail.com

Harika Bandarupally
Student, Computer Science and Engineering
Chaitanya Bharathi Institue of Technology
Hyderabad, India
bandarupallyharika@gmail.com

M Trupthi
Assistant Professor, Information Technology
Chaitanya Bharathi Institue of Technology
Hyderabad, India
trupthijan@gmail.com

***Abstract*— Novel Coronavirus (COVID-19) outbreak that emerged originally in Wuhan, the Hubei province of China has put the entire human race at risk. This virus was declared as Pandemic on 11th March 2020. Considering the massive growth rate in the number of cases and highly contagious nature of the virus, machine learning prediction models and algorithms are essential to predict the number of cases in the coming days. This could help in reducing the stress on health care systems and administrations by helping them plan better. In this paper the datasets used are obtained from the John Hopkins University's publicly available datasets to develop a state-of-the-art forecasting model of COVID-19 outbreak. We have incorporated data-driven estimations and time series analysis to predict the trends in coming days such as the number of cases confirmed positive, number of deaths caused by the virus and number of people recovered from the novel coronavirus. To achieve the estimations, we have used the Deep learning model long-short-term memory network (LSTM).**

Keywords— Deep learning, Artificial Neural Networks, Long-Short-Term Memory (LSTMs), Pandemic, COVID-19, Corona-virus.

I. Introduction

The World has been affected by a highly contagious virus called the Corona virus or SARS-COV-2. This virus originated in the wet markets of Wuhan, Hubei province of China during December 2019. This virus quickly spread to more than 160+ countries within a span of 3 months causing over 400,000 deaths with more than 8.9 million people affected globally[7]. This virus has caused very distressing times across all the countries and significant disruptions in global economies. Several intervening measures have been taken by the affected countries such as quarantining people to stop the spread of the virus.

Coronavirus being a contagious and infectious disease like the flu with certain growth patterns, such patterns are noted to be non-linear and dynamic in nature. Data is Dynamic in nature as the cases might differ based on the seasons, populations etc. [2]. Thus a deep learning model based on long short term memory networks using Pytorch framework can be used to predict the data accurately.

Deep learning power in the field of Artificial Intelligence can be established by recurrent neural networks (RNNs) and LSTMs. These models are one of the best dynamic models that are used to generate sequences in multiple domains such as recognizing speech and music, emotional tone prediction for a piece of text (sentiment-classification of text), caption generation and machine translations [3]. There are different methods to achieve the task for time-series analysis, Machine learning algorithms like Linear and Logistical Regressions, SVM etc., are at the center of these applications [6]. While these tools are great in examining observations and reaching to conclusions, they come with some serious limitations. In most cases the data is skewed and relativistic. Considering this a robust new method using deep learning models are inevitable to gain time series forecasting results with higher accuracy.

II. Concepts

A. Artificial Neural Networks (ANN)

ANNs are programmed to try and simulate a human brain by modelling the neural structure on a smaller scale [3]. ANN consists of interconnected web of nodes joined by edges known as neurons. The main function ANN is to perform progressively complex calculations on a set of inputs, then use the output to solve a problem [2]. ANNs are used for lots of different applications. An ANN typically consists of 3 layers namely input, hidden and output layers. Neural net can be seen as a result of spinning classifiers composed in a layered web; this is because every node in the hidden layer and output layer has their own classifier.

B. Recurrent Neural Networks (RNN)

Recurrent neural networks (RNN) find their best usage when the patterns in data vary with time. This deep learning model is a simple structured model with a built-in feedback loop that allows it to act as a forecasting-engine [15]. In the feed forward neural network signals have unidirectional movement from input to output one layer at a time, In RNN the layer's output is added to the next input and fed back into the same layer. Contrary to feed-forward neural nets, an RNN can accept a sequence of values as input and produces a sequence of values as output, the capability to operate in sequence unfolds RNN to a wide variety of applications [13]. It is possible to obtain a capable net of more complex outputs by stacking RNNs one on top of another [20].

C. Long Short-Term Memory Networks (LSTMs)

Characteristically, an RNN is a very challenging neural net to train [20]. Since RNNs make use of back propagation, they run into the problem of vanishing-gradient. Unfortunately, the vanishing-gradient is exponentially worse for an RNN. The reason being that, each time step is the equivalent to an entire layer in a feed-forward neural network. So, training an RNN for a 100-time step is similar to training a 100 layer feed-forward neural net. This results in exponentially small gradients and information decay through time. These problems can be solved using Long-Short-Term-Memory networks (LSTMs). LSTM are modules of RNN that can learn the long-term dependencies. By placing the LSTM modules inside an RNN, long-term dependency challenges can be avoided.

D. Time series data (TS)

Time series data refers to the data that is collected over a regular time period and captures a series of data points captured at regular intervals of time where every data point is equally spaced over time [4]. Trend, seasonality and error are the important components of a time-series data. Forecasting upcoming patterns and trends based on historical dataset containing temporal features is known as Time Series prediction [5]. Data with temporal components will be the best suited data to forecast the novel coronavirus transmission [6]. A time-series data pattern can be noticed when a certain trend recurrences at regular time periods like confirmed cases, deaths, recovered cases etc. In many real-time situations, either seasonality or trend is absent. After finding the nature of time series data, different forecasting methods must be applied. The two categories of time-series data are non-stationary data and stationary data. A stationary series is independent of the time components such as seasonality, trends etc. Constant mean and variances are observed with respect to time. A non-stationary depends on the seasonality effects and trends in it and varies with respect to time. Statistical properties like mean, variance and standard deviation also changes with respect to time. Compared to non-stationary TS, stationary TS data is easier to analyze and provides good forecasting results [4].

E. Pytorch

Pytorch is a high-quality deep learning library with plenty of extensions and a support of a large community. Pytorch offers GPU support, the option to set up a deep net by configuring its hyper parameters. Once configured the deep net can be called from the routines of our programs. This library provides a powerful vectorized implementation of the math behind deep learning; In addition there are many libraries that extend Pytorch functionality for various applications.

III. SELECTION OF DEEP LEARNING MODEL

Time series forecasting is a challenging problem when working with noisy dynamic data. Deep learning models offer a lot of promise when working with time series data [8]. In this section different deep learning models are compared and why LSTMs are the better choice when compared to other models used for time series forecasting can be understood.

A. Models

- *Traditional Time Series Forecasting methods like ARIMA:* Traditional models like ARIMA require complete data, incomplete and noisy data as in this case cannot be applied to this model. Model works best for univariate and linear-relationships. Traditional models do not work well for long-term [13].
- *Multi-layer-Perceptron's(MLP) for Time-series :* MLP's are best suited for problems having meaningful mappings unlike the problem statement presented in this paper. Static mapping functions and fixed inputs and outputs are required.
- *Convolution Neural Networks (CNNs):* CNNs are often used for image classification problems. CNNs though can extract import features from input sequence and enjoys benefits missing from traditional models and MLPs, it cannot learn from temporal dependency. CNNs are slow and fickle to train of time series data. The model also over fits easily [14].
- *Recurrent Neural Networks (RNNs):* RNNs suffer from long-range dependencies because of the vanishing gradient problem and exploding. Gradient vanishing in a Recurrent Neural Network (RNN) refers to the challenges where the long-term component's gradient norm decreases exponentially quickly to 0, hindering the ability of the model from learning long-term temporal correlations. The opposite event to this can be referred as gradient exploding. LSTM has been introduced to address the issue [14].
- *Any Hybrid models:* In this paper hasn't pursued hybrid models like CNN-LSTM etc., as a simple LSTM network structure serves the purpose and doesn't risk over-fitting.

To address and mitigate the limitations of the above mentioned models, LSTMs have been chosen [4]. The benefits of using LSTMs are they establish temporal connections, define and maintain an internal memory cell state during the course of the entire life cycle of this model. In addition they are simple, well-understood, approximate non-linear functions, robust to noise, can handle multi-step forecasts and multivariate inputs. The LSTM is designed to estimate the movement of Covid-19s spread with consideration of uncertainties [7].

IV. METHODOLOGY OF TIME SERIES PREDICTION OF COVID-19 DATA USING LSTM

A. Data Exploration

The data in the paper contains different categories of time series data namely total no. of cases, deaths and no. of people who have recovered from novel corona virus. For Each category of data on exploring the .csv file structure the Province/State, Country/ Region/ Latitude, Longitude and cumulative number of virus cases, deaths due to the virus and recovered patients can be found as shown in Fig.3 and Fig.4 respectively.

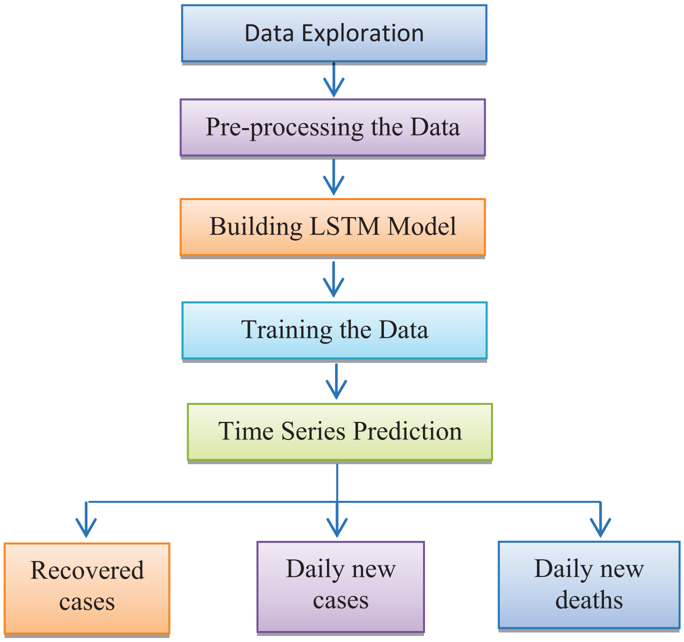
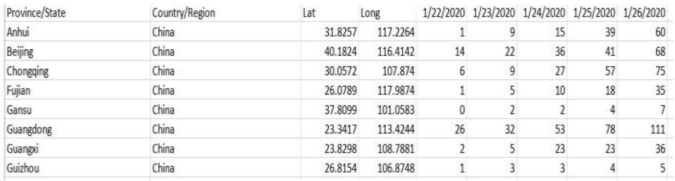
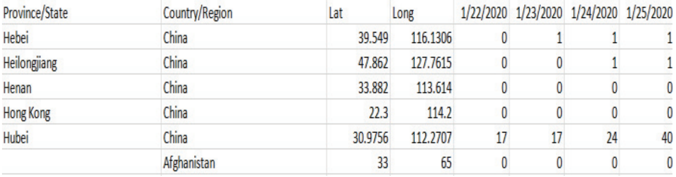


Fig. 1. Represents the flowchart of the methodology used.



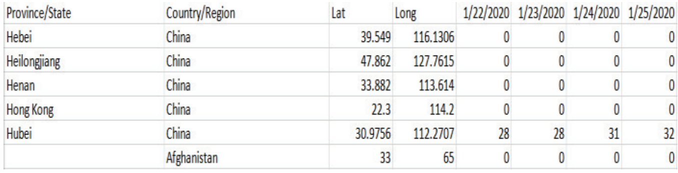
Province/State	Country/Region	Lat	Long	1/22/2020	1/23/2020	1/24/2020	1/25/2020	1/26/2020
Anhui	China	31.8257	117.2264	1	9	15	39	60
Beijing	China	40.1824	116.4142	14	22	36	41	68
Chongqing	China	30.0572	107.874	6	9	27	57	75
Fujian	China	26.0789	117.9874	1	5	10	18	35
Gansu	China	37.8099	101.0583	0	2	2	4	7
Guangdong	China	23.3417	113.4244	26	32	53	78	111
Guangxi	China	23.8298	108.7881	2	5	23	23	36
Guizhou	China	26.8154	106.8748	1	3	3	4	5

Fig. 2. Represents the three data sets for total number of cases.



Province/State	Country/Region	Lat	Long	1/22/2020	1/23/2020	1/24/2020	1/25/2020
Hebei	China	39.549	116.1306	0	1	1	1
Heilongjiang	China	47.862	127.7615	0	0	1	1
Henan	China	33.882	113.614	0	0	0	0
Hong Kong	China	22.3	114.2	0	0	0	0
Hubei	China	30.9756	112.2707	17	17	24	40
	Afghanistan	33	65	0	0	0	0

Fig. 3. Represents the three data sets for total number of deaths.



Province/State	Country/Region	Lat	Long	1/22/2020	1/23/2020	1/24/2020	1/25/2020
Hebei	China	39.549	116.1306	0	0	0	0
Heilongjiang	China	47.862	127.7615	0	0	0	0
Henan	China	33.882	113.614	0	0	0	0
Hong Kong	China	22.3	114.2	0	0	0	0
Hubei	China	30.9756	112.2707	28	28	31	32
	Afghanistan	33	65	0	0	0	0

Fig. 4. Represents the three data sets for total number of people recovered from the virus respectively.

Since the Province/State, Country/ Region/ Latitude and Longitude are not required, operations can be performed to remove these columns. Next check for missing values in the data and sum all the rows to get cumulative daily cases. Then proceed to convert the date column into date time structure using Pandas. In the next step, plot the data. The results of cumulative daily cases can be seen in Fig. 5.

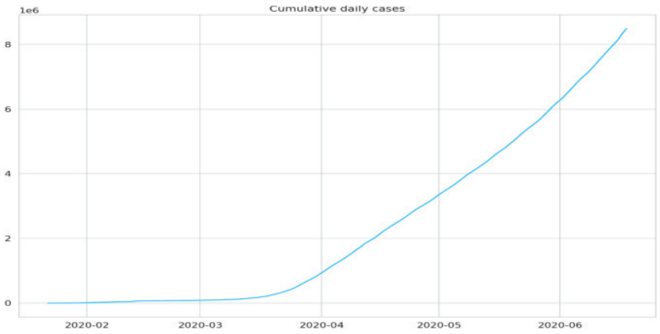


Fig. 5. Plot of cumulative daily cases over the months.

Next the accumulation is undone by subtracting the present value from the preceding value and saving the sequence's first value. This results in an increase of cases on daily bases. Plot of the daily cases can be found in Fig. 6.

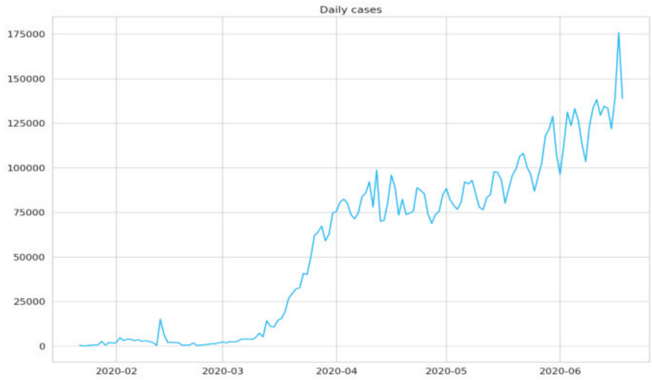


Fig. 6. Plot of cumulative daily cases over the months.

B. *Preprocessing the Data*

Split the data into training and testing data. In this paper a 80-20 split is used to separate training and testing data. Training data consists of 104 numbers of days.

Next scale the training data between zero to one in order to improve the performance and training speed. In order to achieve this MinMaxScaler from scikit-learn is used. The big sequence of daily cases is converted to smaller ones. Every training sample comprises a sequence of five data points of history and a label for the real value that the model is required to predict. Now, create the actual sequences for the time sequence data to feed to our LSTM model, which work by inputting a sequence of numbers or vectors and output a number, like required in this model. This is similar to regression to classify sequences.

C. *Building the model*

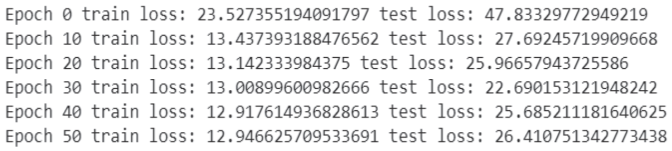
The difficulty of the model is encapsulated into a class that belongs to *torch.nn.Module.* The model presented in this paper consists of three main methods namely:

- *Constructor method*: To create the layers and initialize all helper the data.
- *Rest hidden state method*: A stateless LSTM is used, which requires the state to be reset after every example.

- *Forward*: To get the sequences, pass all the sequences through LSTM layer, at once. The output from the last time step is passed through linear layer to obtain the prediction.

D. *Method for Training the model with limited data*

A builder function is used to train the model. Observe that the hidden-state of the model is being reset at the beginning of every epoch. Batches of data are not used in this model; the model is exposed to all the examples at once. An instance of the model is created and trained. The train and test loss of the model is as shown in Fig. 7 and Fig. 8.



```
Epoch 0 train loss: 23.527355194091797 test loss: 47.83329772949219
Epoch 10 train loss: 13.437393188476562 test loss: 27.69245719909668
Epoch 20 train loss: 13.142333984375 test loss: 25.96657943725586
Epoch 30 train loss: 13.00899600982666 test loss: 22.690153121948242
Epoch 40 train loss: 12.917614936828613 test loss: 25.685211181640625
Epoch 50 train loss: 12.946625709533691 test loss: 26.410751342773438
```

Fig. 7. Represents the train and test loss with respect to epochs.

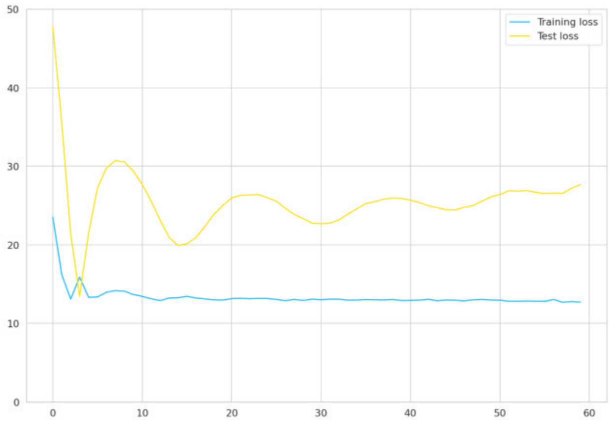


Fig. 8. Plot train and test loss.

E. *Predicting daily case*

At this stage based on how this model is trained, only a single day in the future can be predicted. A straight forward approach is employed to conquer this limitation. The next future days' output is predicted by inputting the previously predicted values. Fig. 9 shows that the predictions seem to be around the same ballpark.

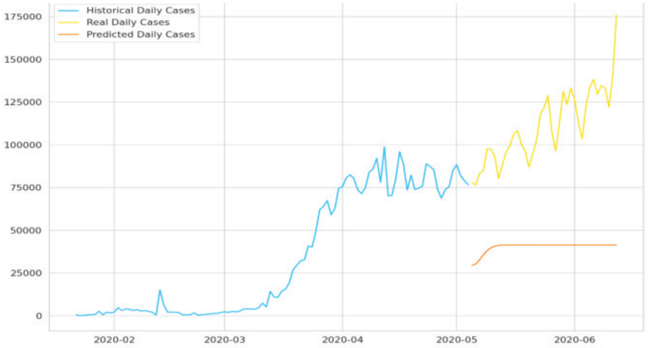


Fig. 9. Plot of Historical Daily cases, Real Daily cases and Predicted daily cases.

F. *Predicting future cases*

All the available data is utilized in training the model. The pre-processing and training steps are the same as explained above. The fully trained model is used to predict confirmed positive virus cases for next 12 coming days. In order to create charts containing historical and predicted cases the date index of the data frame is extended. Fig. 10 shows the predicted future cases.

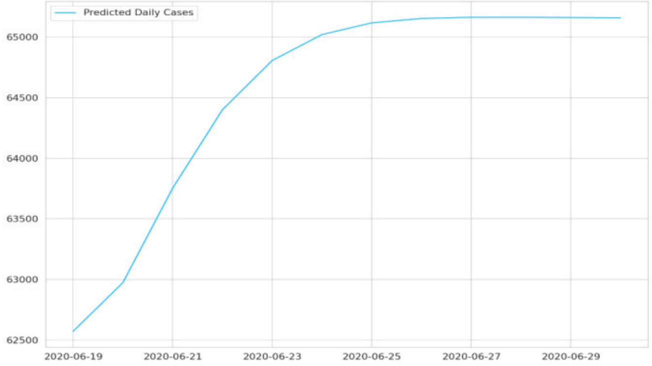


Fig. 10. Plot of predicted future cases for next 12 days.

All the data is used to plot the results as shown in Fig. 11.

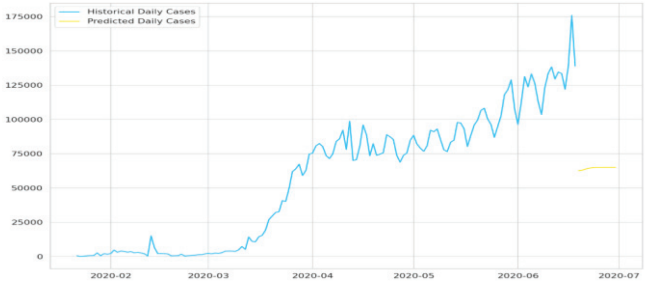
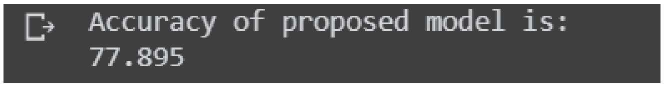


Fig. 11. Plot of historical daily cases and predicted daily cases over the months.

The accuracy of the model proposed is 77.89% since the model is exposed to limited data, as the data increases and varies trends and patterns of the virus are exposed to the model and trained on, the accuracy of the model will be improved.



```
Accuracy of proposed model is:
77.895
```

Fig. 12. Accuracy of the proposed model.

G. *Predicting Daily Recovery and Deaths*

As mentioned in this paper earlier, our model can also be used to predict the number of recoveries as well as the number of deaths. Refer Fig. 13 and Fig. 14 for results.

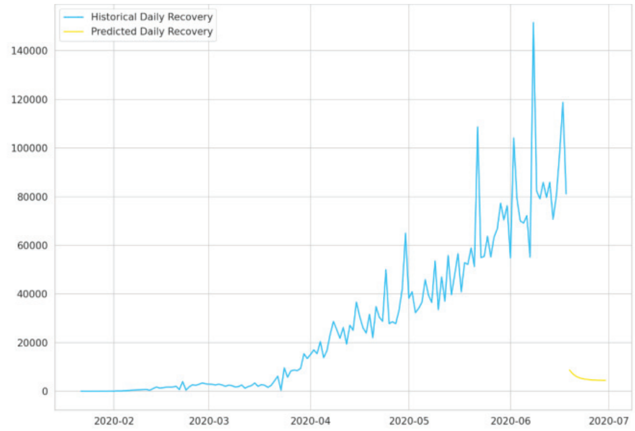


Fig.13. Plot of historical and predicted daily recovered cases.

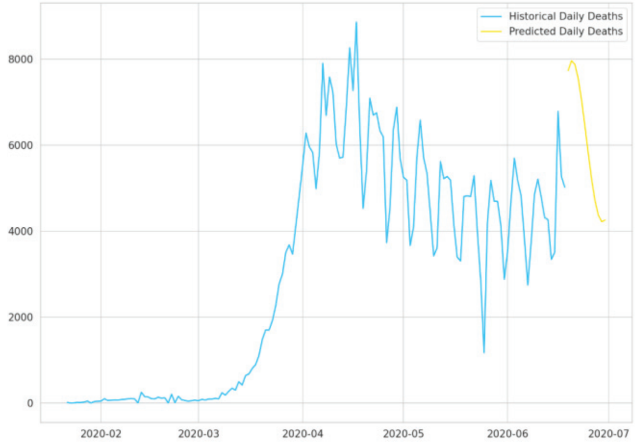
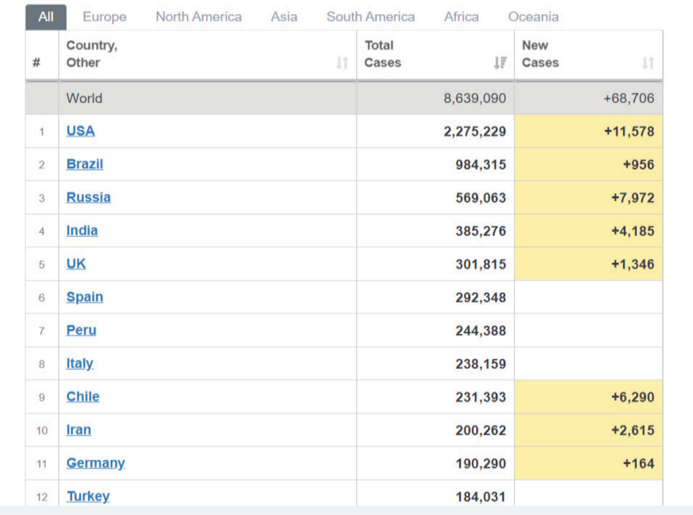


Fig. 14. Plot of historical and predicted daily deaths.

H. *Comparison with real world data*

It can be inferred from Fig. 10. That the number of cases predicted for 19 June 2020 is >62,500.According to “Worldometers.info” (refer Fig. 15), the original new cases on 19 June 2020 is 68,706 which implies that the values predicted by our model are approximately equal to the original values.



#	Country, Other	Total Cases	New Cases
	World	8,639,090	+68,706
1	USA	2,275,229	+11,578
2	Brazil	984,315	+956
3	Russia	569,063	+7,972
4	India	385,276	+4,185
5	UK	301,815	+1,346
6	Spain	292,348	
7	Peru	244,388	
8	Italy	238,159	
9	Chile	231,393	+6,290
10	Iran	200,262	+2,615
11	Germany	190,290	+164
12	Turkey	184,031	

Fig. 15. COVID-19 stats on 19-06-20 from Wordometers.info.

V. CONCLUSION

Predicting COVID-19 cases has immense significance in the present dire scenario. In this work the growth patterns of the disease have been analyzed, data-driven estimations have been incorporated. Deep learning model based on RNN, LSTMs and time series analysis have been used to predict the trends in coming days such as the no. of confirmed positive viral cases, no. of deaths caused by the virus and number of people recovered from the novel corona virus. Encouraging experimental results have been obtained on the dataset used.

VI. FUTURE SCOPE

The problem of predicting Covid-19 related data such as future cases, recovered cases and deaths is difficult, since we are amidst an outbreak [11]. The future trends and patterns may vary widely based on myriad external conditions ([12], [15]) like quarantine measures, new behavior of the virus strain, population of a country [14] etc., as the dataset becomes larger and we have more data to train our model, we can improve the accuracy. The same model can be used to predict any future pandemics that are similar in nature to SARS COVID-19. This model can be integrated with an application that streams live data from government sites to view real time graphs of COVID-19 related data. Hope that everything will recover and get back to normal soon.

REFERENCES

[1] B. Eddy Patuwo, and Michael Y. Hu. ”Forecasting with artificial neural networks:: The state of the art.” International journal of forecasting 14.1 (1998): 35-62.

[2] Jay S Sevak, Aerika D. Kapadia, Jaiminkumar B. Chavda, Arpita Shah, Mrugendrasinh Rahevar. "Survey on semantic image segmentation techniques", 2017 International Conference on Intelligent Sustainable Systems (ICISS), 2017

[3] "Proceedings of the Third International Conference on Computational Intelligence and Informatics", Springer Science and Business Media LLC, 2020

[4] Vinay Kumar Reddy Chimmula, Lei Zhang. "Time Series Forecasting of COVID-19 transmission in Canada Using LSTM Networks", Chaos, Solitons & Fractals, 2020

[5] Hochreiter, Sepp, and J¨urgen Schmidhuber. ”Long short-term memory.” Neural computation 9.8 (1997): 1735-1780.

[6] L¨angkvist, Martin, Lars Karlsson, and Amy Loutfi. ”A review of unsupervisedfeature learning and deep learning for time-series modeling.” Pattern Recognition Letters 42 (2014): 11-24.

[7] Taieb, Souhaib Ben, et al. ”A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition.” Expert systems with applications 39.8 (2012): 7067-7083.

[8] Yang, Zifeng, et al. ”Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions.” Journal of Thoracic Disease 12.3 (2020): 165.

[9] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. ”Deep learning.” nature 521.7553 (2015): 436-444.

[10] https://www.worldometers.info/coronavirus/

[11] Peng, Liangrong, et al. ”Epidemic analysis of COVID-19 in China by dynamical modeling.” arXiv preprint arXiv:2002.06563 (2020).

[12] Roda, Weston C., et al. ”Why is it difficult to accurately predict the COVID-19 epidemic?.” Infectious Disease Modelling (2020).

[13] Hui He, Ran Hu, Ying Zhang, Runhai Jiao, Honglu Zhu. "Chapter 18 Hourly Day-Ahead Power Forecasting for PV Plant Based on Bidirectional LSTM", Springer Science and Business Media LLC, 2019

[14] Benvenuto, Domenico, et al. ”Application of the ARIMA model on the COVID-2019 epidemic dataset.” Data in brief (2020): 105340.

[15] Tayaba Abbasi, King Hann Lim, Ke San Yam. "Predictive maintenance of Oil and Gas Equipment using Recurrent Neural Network", IOP Conference Series:Materials Science and Engineering, 2019.

[16] Yudistira, Novanto, et al. ”UV light influences covid-19 activity through big data: trade offs between northern subtropical, tropical, and southern subtropical countries.” medRxiv (2020).

[17] Paterlini, M. ”’Closing borders is ridiculous’: the epidemiologist behind Sweden’s controversial coronavirus strategy.” Nature (2020).

[18] Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio. ”On the difficulty of training recurrent neural networks.” International conference on machine learning. 2013.

[19] "Web, Artificial Intelligence and Network Applications", Springer Science and Business Media LLC, 2020

[20] Bayer, Justin Simon. Learning Sequence Representations. Diss. Technische Universit¨at Munchen, 2015