r-LSTM: Time Series Forecasting for COVID-19 Confirmed Cases with LSTM-based Framework

Mohammad Masum
Analytics and Data Science Institute
Kennesaw State University
Kennesaw, USA
mmasum@students.kennesaw.edu

Hossain Shahriar
Department of Information Technology
Kennesaw State University
Marietta, USA
hshahria@kennesaw.edu

Hisham M. Haddad
Department of Computer Science Kennesaw State University
Marietta, USA
hhaddad@kennesaw.edu

Md. Shafiul Alam
Analytics and Data Science Institute
Kennesaw State University
Kennesaw, USA
malam6@students.kennesaw.

2020 IEEE International Conference on Big Data (Big Data) | 978-1-7281-6251-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/BigData50022.2020.9378276

***Abstract*--The coronavirus disease 2019 (COVID-19) caused a pandemic outbreak with affecting 213 nations worldwide. Global policymakers are imposing many measures to slow and reduce the rapid growth of the infections. On the other hand, the healthcare system is encountering significant challenges for a massive number of COVID-19 confirmed or suspected individuals seeking treatment. Therefore, estimating the number of confirmed cases is necessary to provide valuable insights into the growth of the outbreak and facilitate policy making process. In this study, we apply ARIMA models as well as LSTM-based recurrent neural network to forecast the daily cumulative confirmed cases. The LSTM architecture generates more precise forecasting by leveraging both short- and long-term temporal dependencies from the pandemic time series data. Due to the stochastic nature in optimization and random initialization of weights in neural network, the LSTM based model produce less reproducible outcome. In this paper, we propose a reproducible-LSTM (r-LSTM) framework that produces a reproducible and robust results leveraging z-score outlier detection method. We performed five round of nested cross validation to show the consistency in evaluating model performance. The experimental results demonstrate that r-LSTM outperformed the ARIMA model producing minimum MAPE, RMSE, and MAE.**

***Keywords*--ARIMA Model, LSTM, Time Series Forecasting, COVID-19 pandemic, coronavirus**

I. INTRODUCTION

The ongoing 2019 novel coronavirus (2019-nCoV), named as COVID-19, epidemic was first identified amid an outbreak of respiratory illness cases in Wuhan, China and later rapidly spread throughout around the globe [7]. It affects respiratory distress (like influenza) with symptoms such as cold, cough, fever, and breathing issue in gradually severe cases [12]. The novel coronavirus outbreak declared as a global pandemic on 11 March 2020 by the WHO due to the growth rate and scale of transmission of the virus [4] The US also declared the epidemic as a public health emergency on 01 February 2020 [5]. The COVID-19 has spread to more than 200 nations worldwide. As of 05 June 2020, the epidemic has resulted in 6,844,797 confirmed cases with 398,146 reported deaths globally [6]. One of the most affected countries is the US where 1,902,632 confirmed cases with 109,359 confirmed deaths reported as of June 05, 2020. Eventually, the virus posed a great danger to the health and safety of people across the world. Global policymakers are imposing many measures to slow and reduce the rapid growth of infections. On the other hand, the healthcare system is encountering significant challenges like testing and caring for a massive number of confirmed or COVID-19 suspected individuals seeking treatment. Additionally, with the exponential growth of COVID-19 patients, the hospitals are facing difficulties in ensuring essential supplies like ventilators, personal protective equipment, and test kits. Therefore, estimating the number of confirmed cases provides valuable insights into the growth of the outbreak and facilitates policy making process. In this study, we applied ARIMA models as well as LSTM-based recurrent neural networks to forecast the daily cumulative confirmed cases in the US.

Time series forecasting is a well-known challenge for infectious disease. Many researchers have already attempted for COVID-19 pandemic time series forecasting where most of the studies have applied statistical or machine learning methods [8, 9, 10, 11]. ARIMA along with other methods like single exponential, double exponential, moving average, and S-curve models were employed for forecasting daily new cases of COVID-19 in India by using data from

22 January 2020 to 13 April 2020 [8]. Experimental results of the study showed that ARIMA (2,2,2) outperformed other methods with a minimum mean squared percentage error (MAPE) of 4.1. Exponential smoothing models were implemented to capture a variety of trend and seasonal forecasting patterns with limited number of training data [9]. The research emphasized on real time cumulative daily cases (from January 22, 2020 until March 11, 2020) in the US and produced ten-days-ahead forecasts along with updating forecasts for every ten days. Another analysis, focusing on 10 Brazilian states COVID-19 daily cumulative cases, utilized several models like ARIMA, cubist regression (CUBIST), random forest (RF), support vector machine (SVR), and stacking-ensemble learning for short term forecasting like one-, three-, and six-days ahead forecasting [10]. The study showed that the SVR ranked best by evaluating the models' performance with mean absolute error (MAE) and symmetric MAPE. Autoregressive time series models based on the two–piece scale mixture normal distribution that can avoid assumption of symmetric distribution of the error terms was applied to cumulative confirmed and recovered cases in the world utilizing data from February 02, 2002 to April 30, 2020 [11]. Moving average, weighted moving average, and single exponential smoothing methods were applied to several countries COVID-19 confirmed, death, and recovered time series data. [13]. There is limited number of published research articles that forecast the COVID-19 confirmed cases in the US.

The rest of the paper is organized as follows: Section 2 describes the methodologies (ARIMA and proposed r-LSTM) that are implemented in this paper. The experimental setting and results are explained in Section 3. Finally, Section 4 concludes the paper.

II. Methodology

We applied ARIMA model and LSTM based framework to COVID-19 pandemic cumulative daily time series data for forecasting. The ARIMA is well-known statistical method for time series analysis while LSTM, a special variant of RNN, is a state-of-the-art technique that has been successfully applied on time series analysis and virus prediction [15, 18].

LSTM is an extended version of recurrent neural network (RNN) architecture that can learn long term dependencies. The building block of LSTM architecture is memory block that consists of memory cell to preserve information of preceding time step with self-recurrent connections. Fig. 1 displays an internal architecture of a memory cell. The cell consists of three controlling gates: input gate, forget gate, and output gate [17, 18]. The forget gate decides what information should be preserve or removed from the memory cell using a sigmoid layer, while update of values is controlled by the input gates that leverages a *tanh* layer and a sigmoid layer [19]. The sigmoid function determines which value should be updated and the tanh layer generates potential values that can be added to the memory cell. The output gate utilizes sigmoid function to decide memory contribution to the cell output and then a *tanh* activation is applied to capture non-linearities of the values. Finally, the output value is multiplied with the output of a sigmoid layer. Equations (1-5) describes the full mechanisms of a LSTM model for an input x_t at time *t* where f_t, i_t, o_t, and $c_t$are the forget, input, output gates, and internal memory cell state at time t, respectively. h_t, and $h_{t-1}$are the values of hidden layer of the LSTM memory cell at time step t, and $t-1$, correspondingly. $\otimes$, and σ denote elementwise multiplication, and sigmoid activation function, respectively.

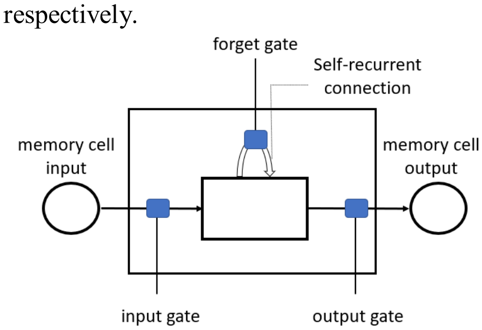

Figure 1: Internal architecture of a memory cell of LSTM

$$f_t = \sigma\,(W_{f_h}[h_{t-1}], W_{f_x}[x_t], b_f) \quad (1)$$

$$i_t = \sigma\,(W_{i_h}[h_{t-1}], W_{i_x}[x_t], b_i) \quad (2)$$

$$c_t = f_t \times c_{t-1} + i_t \otimes \, tanh(W_{c_h}[h_{t-1}], W_{c_x}[x_t], b_c) \quad (3)$$

$$o_t = \sigma\,(W_{o_h}[h_{t-1}], W_{o_x}[x_t], b_o) \quad (4)$$

$$h_t = o_t \otimes tanh(c_t) \quad (5)$$

LSTM architecture extensively takes advantage of the gates that sophisticatedly adjust the values of memory cells and provide an internal dynamic in a cooperative way [21]. Considering this property of LSTM, in general, it shows a superior ability to learn nonlinear statistical and temporal dependencies of real-world time series data [19].

The output of neural networks varies due to the stochastic nature in optimization and random initialization of weights. Hence, to generate more reproducible and robust results, we propose a simple but effective framework, named r-LSTM, that generate reproducible results. For reproducible results, the r-LSTM initially executes experiments n number of times and subsequently utilizes the summary statistics of repetitions. We assumed that there are outliers in the distribution of n repetitions. A z-score method to detect outliers in the distribution and remove any output outside of two standard deviations from the mean. Finally, the mean is calculated for each unit of the forecasts. Fig. 2 shows the architecture of r-LSTM framework.

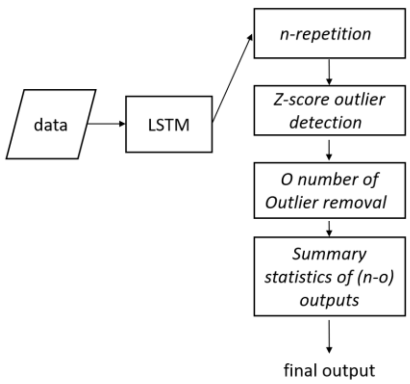

Figure 2: Architecture of the r-LSTM framework

III. Experiments and Results

A. Dataset specification

The daily cumulative confirmed cases of the COVID-19 pandemic in the USA. from 22nd January to 25th May were collected from the official website of John Hopkins University [22]. Fig. 3 shows the patterns of daily cumulative confirmed cases from COVID-19, where the first confirmed case was reported on 22nd January and 1,662,302 number of cumulative confirmed cases were reported on 25th May.

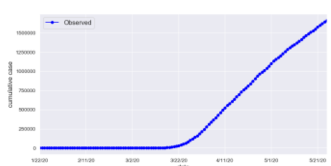

Figure 3: COVID-19 daily cumulative confirmed cases in the US from 22nd January to 25th May

The conventional cross-validation (CV) technique, widely used in applied machine learning, does not consider temporal dependency between observations, and subsequently utilizes values from the future to forecast the past. Hence, the traditional CV is not appropriate for time series data. In this study, we applied a 5-fold nested CV that preserves the order of the observations. Fig. 4 exhibits the nested five rounds of datasets where the blue, green, and red boxes include the number of forecast rounds, training, and test data, respectively. The training data are used to construct models, while test data used for forecasting. For instance, for the first round of forecast: models are constructed on data from Jan 22 to Apr 05 and then forecast subsequent 10 days (from Apr 06 to Apr 15). Maintaining consistency of results, we performed forecasts for 10 days ahead of the last day of training data for all five rounds. Analysis of forecasting process and experimental results for each of the five-round forecasts are comprehensively discussed in later sections.

The COVID-19 pandemic dataset does not contain any information of patients. Thus, no formal ethical review or prior informed consent was required.

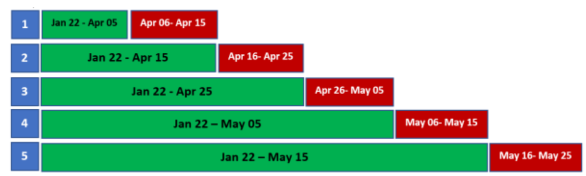

Figure 4: Nested five rounds of COVID-19 time series data

B. Evaluation metrics

Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) are widely used metrics when comparing different forecasting methods applied to a single time series data. Equations (6), (7), and (8) are the mathematical definitions of MAPE, RMSE, and MAE, respectively where y_i and $\hat{y}_i$ are the i-th observed value and forecasted value respectively and N is the number of test data points.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \quad (6)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \quad (7)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \quad (8)$$

C. ARIMA experimental setting

Table 1 demonstrates a summary of the developing procedure of the final ARIMA model for each round of the forecasts. The column 'Transformation' shows the required power transformation, and the "Differencing" column shows required differencing of the original series for transforming the data into a stationary process. For instance, the series for the first round was converted into a stationary process by transforming ($x_i' = x_i^{\frac{1}{3}}$) at first and then carrying out both the first and second differences of the series. The Augmented-Dickey Fuller test was performed on the transformed series to validate stationarity. The test provides significant p-values (less than 0.001) for each round of the process, confirming that the series finally transformed into stationarity process. Once stationarity is achieved, then different potential models are applied, and eventually, the best model, that produced minimum MAPE, is selected for each of the rounds.

Table 1:The process of conversion of stationarity for each round

Round	Transformation	Differencing
1st	$x_i' = x_i^{\frac{1}{3}}$	1st & 2nd
2nd	$x_i' = x_i^{\frac{1}{4}}$	1st & 2nd
3rd	$x_i' = x_i^{\frac{1}{4}}$	1st, 2nd, & 3rd
4th	$x_i' = x_i^{\frac{1}{4}}$	1st, 2nd, & 3rd
5th	$x_i' = x_i^{\frac{1}{4}}$	1st & 2nd

D. *r-LSTM experimental setting*

We applied the min-max normalization technique to the time series data to scale the original data to a fixed range of 0 and 1. We applied the r-LSTM network on each of the five-round of normalized datasets with default Keras initialization weights where "glorot uniform" is the initializer for kernel weights matrix and orthogonal initializer is used as the recurrent initializer. The bias vector is initialized with all zeros. "Sigmoid" activation function is used in the recurrent step. We used 'ReLu' activation function in the hidden layer. Adam optimization algorithm and mean squared error were used as optimizer and loss function, respectively. 150 LSTM units were used in the hidden layer. The learning rate remains fixed at 0.001. All the discussed hyper-parameters were optimized using a grid search procedure. All the experiments were performed with the identical setup to maintain consistency in the model evaluation process.

In the process of r-LSTM, we ran all experiments 30 times since it follows the central limit theorem of the sampling distribution. Consequently, we performed a z-score method to detect and remove outliers from the forecasting list. Finally, mean of the remaining experiments was calculated for each day of forecasts.

The r-LSTM network with optimized hyper-parameters was applied to each of the five-round of forecasts with time series generator class in Keras. In the training phase, the generator uses n number of lag observations as input with a fixed batch size of 1 as output that allows generating a 1-step ahead forecast after the last available date in the training set. We opted to employ recursive multi-step time series forecasting criteria that involve recursively applying the model for one step ahead forecasts until reaching the desired n-step forecast horizon. In this process, the prior time step forecast was fed to the model as an input to forecast the sequential time step.

The experiments are carried out on a Windows 10 Intel(R) Core (TM) i7-8565U CPU 1.80 GHz with 16.0 GB RAM and NVIDIA GeForce MX250 2GB GDDR5. We implemented our experiment on Keras framework in Python 3.7 version.

E. *Results*

We applied ARIMA on the stationary process for each of the rounds of forecasts. The diagnostics of residual correlation of the best-fitted model are presented in the fig.5. The figures represent ACF, PACF, IACF, white noise probability of the residual correlation.

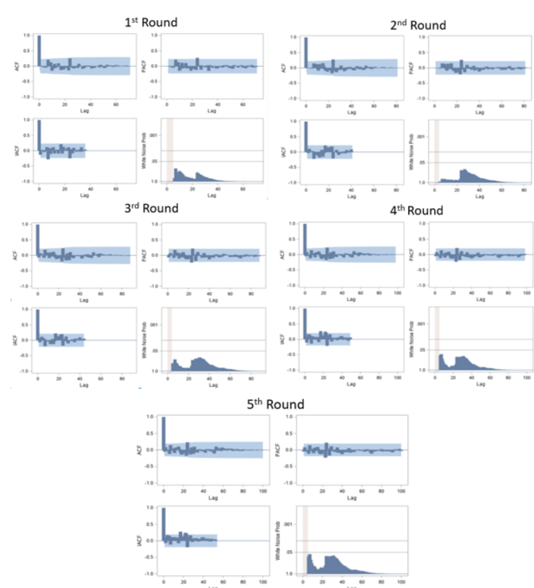

Figure 5: ACF, PACF, IACF, and White noise probability plots for each round

Table 2: Forecasting accuracy using ARIMA method for each round

Round	Best model	MAPE	RMSE ($\times 10^4$)	MAE ($\times 10^4$)
1st	ARIMA (2,3)	14.1	10.2	7.98
2nd	ARIMA (2,2)	3.36	3.93	2.93
3rd	ARIMA (0,5)	0.57	0.70	0.61
4th	ARIMA (0,5)	0.51	0.70	0.71
5th	ARIMA (0,5)	0.81	1.62	1.30

The best models, for each round of forecasts, satisfy all diagnostics criteria. As an illustration, the ACF and PACF for first-round forecast show that there is no significant correlation of residuals series. The IACF does not reveal any instability and the white noise probability shows significance. Table 2 displays the selected best ARIMA model, and forecasting accuracy for each of the rounds. For example, for the 1st round of forecasts, the ARIMA (2,3) model was selected as the best model based on the lowest MAPE.

We applied the r-LSTM framework with *n* number of lag observations time steps from 1 to 15 for each round of forecasts and then selected the time steps that produce minimum MAPE. Hence, in total, we performed $15 \times 5 \times 30 = 2250$ experiments. Table 3 displays the optimal lag observations and forecasting accuracy using r-LSTM method for all five rounds. The minimum MAPE, RMSE, and MAE for 1st-, 2nd-, 3rd-, 4th-, and 5th- round of forecasts were attained by observing 2, 3, 2, 9, and 14 of lag days, respectively. Best forecasting accuracy (MAPE score: 0.12; RMSE: 0.23×10^4, MAE: 0.19×10^4) was achieved for 5th round of forecast where the number of training data was maximum comparing to other rounds. On the other hand, with less training size, the first round performed poorly comparing to other forecasts.

Table 3: Forecasting accuracy using r-LSTM method for each round

	Lag days	MAPE	RMSE ($\times 10^4$)	MAE ($\times 10^4$)
1st round	2	5.04	5.23	3.40
2nd round	3	2.12	2.57	1.84
3rd round	2	0.45	0.58	0.49
4th round	9	1.08	2.01	1.50
5th round	14	0.12	0.23	0.19

Fig. 6 shows the trend of actual values (blue), ARIMA forecasted values (green) and r-LSTM forecasted values (red color). For first round: it shows that the r-LSTM framework forecasts are initial 5 days and then the difference between actual confirmed cases and forecasted cases starts increasing. On the other hand, the ARIMA model shows comparative poor forecasts from the day four. The second round of

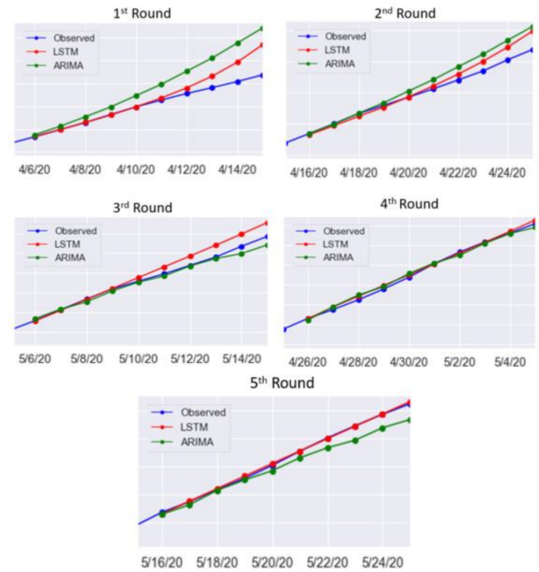

Figure 6: Comparative analysis for ARIMA and r-LSTM models' performance

Table 4:Average performance of ARIMA and r-LSTM models

	MAPE	RMSE ($\times 10^4$)	MAE ($\times 10^4$)
ARIMA	3.87 ± 5.2	3.42 ± 3.5	2.70 ± 2.7
r-LSTM	**1.76 ± 1.7**	**2.12 ± 1.7**	**1.48 ± 1.1**

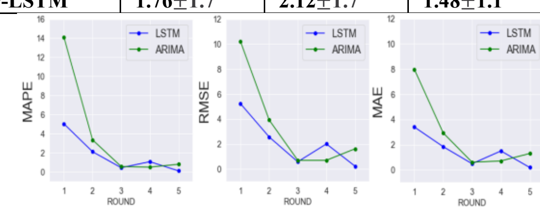

Figure 7: Forecasting accuracies of ARIMA & LSTM against different rounds

forecast follows similar pattern as of first rounds. Both r-LSTM and ARIMA model perform well with less forecasting error and follow similar pattern for 3rd round. For fourth round: the forecasts of r-LSTM show an approximately linear pattern and starts over forecasting after fifth day, while the ARIMA provides approximately accurate forecasts until day eight and then under forecasts for 9th, and 10th days. The r-LSTM produces nearly identical forecasts for all ten days of the fifth round, while the ARIMA starts under forecasts of daily cumulative confirmed cases after 5th day.

From Table 2 & 3, r-LSTM outperformed ARIMA model for 1st-, 2nd -, 3rd -, and 5th – round of forecasts while ARIMA surpass forecast accuracy for 4th round in terms of MAPE, RMSE, and MAE scores. The LSTM framework shows significantly improved accuracy (MAPE score: 5.04) for 1st-round of forecasting, while ARIMA generates 14.1 MAPE score. Forecasting accuracy shows upward trend (by minimizing MAPE, RMSE, and MAE) of improvement with more training data for both ARIMA and LSTM models as shown in Fig. 7. The r-LSTM outperformed the ARIMA models on average of five rounds of forecast. Table 4 shows the average performance of ARIMA and LSTM models. The mean of LSTM scores for MAPE, RMSE, and MAE are lower than the ARIMA models' and at the same time, the standard deviations for ARIMA model much higher than the LSTM models.

IV. Conclusion

The novel coronavirus epidemic was first identified amid an outbreak of respiratory illness cases in Wuhan, China and later rapidly spread around the globe. The infectious disease has caused 6,844,797 confirmed cases and 398,146 confirmed deaths by June 05, 2020, fundamentally affecting the USA along with other countries. Due to the exponential spread and transmission rate, the virus poses a great danger to the health and safety of people across the globe. Therefore, development of accurate forecasting models is necessary to provide valuable insights into the growth of the outbreak and facilitate policy making process. In this study, we present a reproducible r-LSTM framework to forecast daily cumulative confirmed cases in the US. Nested five round forecasting method was applied to demonstrate the consistency of the models' performance. The experimental results suggest the superiority of r-LSTM over ARIMA model.

References

[1] Estrada, E. (2020). Topological Analysis of SARS CoV-2 Main Protease. bioRxiv.

[2] Ortea, I., & Bock, J. O. (2020). Re-analysis of SARS-CoV-2 infected host cell proteomics time-course data by impact pathway analysis and network analysis. A potential link with inflammatory response. BioRxiv.

[3] Liu, F., Lu, Y., & Cai, M. (2020). A Hybrid Method With Adaptive Sub-Series Clustering and Attention-Based Stacked Residual LSTMs for Multivariate Time Series Forecasting. IEEE Access, 8, 62423-62438.

[4] Elmousalami, H. H., & Hassanien, A. E. (2020). Day level forecasting for Coronavirus Disease (COVID-19) spread: analysis, modeling and recommendations. arXiv preprint arXiv:2003.07778.

[5] Sohrabi, C., Alsafi, Z., O'Neill, N., Khan, M., Kerwan, A., Al-Jabir, A., ... & Agha, R. (2020). World Health Organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19). International Journal of Surgery.

[6] https://www.cdc.gov/nchs/nvss/vsrr/covid19/index.htm

[7] https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/how-covid-spreads.html?CDC_AA_refVal=https%3A%2F%2Fwww.cdc.gov%2Fcoronavirus%2F2019-ncov%2Fabout%2Findex.html

[8] Tandon, H., Ranjan, P., Chakraborty, T., & Suhag, V. (2020). Coronavirus (COVID-19): ARIMA based time-series analysis to forecast near future. arXiv preprint arXiv:2004.07859.

[9] Petropoulos, F., & Makridakis, S. (2020). Forecasting the novel coronavirus COVID-19. PloS one, 15(3), e0231236.

[10] Ribeiro, M. H. D. M., da Silva, R. G., Mariani, V. C., & dos Santos Coelho, L. (2020). Short-term forecasting COVID-19 cumulative confirmed cases: Perspectives for Brazil. Chaos, Solitons & Fractals, 109853.

[11] Maleki, M., Mahmoudi, M. R., Wraith, D., & Pho, K. H. (2020). Time series modelling to forecast the confirmed and recovered cases of COVID-19. Travel Medicine and Infectious Disease, 101742.

[12] Sujath, R., Chatterjee, J. M., & Hassanien, A. E. (2020). A machine learning forecasting model for COVID-19 pandemic in India. Stochastic Environmental Research and Risk Assessment, 1.

[13] Elmousalami, H. H., & Hassanien, A. E. (2020). Day level forecasting for Coronavirus Disease (COVID-19) spread: analysis, modeling and recommendations. arXiv preprint arXiv:2003.07778.

[14] Newbold, P. (1983). ARIMA model building and the time series analysis approach to forecasting. J. Forecast. 2, 23–35. doi: 10.1002/for.39800

[15] Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., & Jenssen, R. (2017). An overview and comparative analysis of recurrent neural networks for short term load forecasting. arXiv preprint arXiv:1705.04378.

[16] Chimmula, V. K. R., & Zhang, L. (2020). Time series forecasting of covid-19 transmission in canada using lstm networks. Chaos, Solitons & Fractals, 109864.

[17] Choi, J. Y., & Lee, B. (2018). Combining LSTM network ensemble via adaptive weighting for improved time series forecasting. Mathematical Problems in Engineering, 2018.

[18] Liu, L., Han, M., Zhou, Y., & Wang, Y. (2018, June). Lstm recurrent neural networks for influenza trends prediction. In International Symposium on Bioinformatics Research and Applications (pp. 259-264). Springer, Cham.

[19] Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019). A Comparative Analysis of Forecasting Financial Time Series Using ARIMA, LSTM, and BiLSTM. arXiv preprint arXiv:1911.09512.

[20] Chakraborty, T., & Ghosh, I. (2020). Real-time forecasts and risk assessment of novel coronavirus (COVID-19) cases: A data-driven analysis. Chaos, Solitons & Fractals, 109850.

[21] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015, pp. 4041–4049, Chile, December 2015.

[22] https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-case