A deep learning framework using CNN and stacked Bi-GRU for COVID-19 predictions in India

Sahil Ahuja · Nitin Arvind Shelke · Pawan Kumar Singh

Received: 22 December 2020 / Revised: 9 June 2021 / Accepted: 12 July 2021 / Published online: 23 July 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract
The novel coronavirus infection (COVID-19) first appeared in Wuhan, China, in December 2019. COVID-19 declared as a global pandemic by the WHO was the most rapidly spreading disease all across the world. India, the second most populated nation in the world, is still fighting it, when coronavirus reached the stage where community transmission takes place at an exponential rate. Therefore, it is crucial to examine the future trends of COVID-19 in India and anticipate how it will affect economic and social growth in a short run. In this paper, a new deep learning framework using CNN and stacked Bi-GRU has been developed for predicting and analyzing the COVID-19 cases in India. The proposed model can predict the next 30 days’ new positive cases, new death cases, recovery rate and containment and health index values with high accuracy. The proposed method is compared against Gaussian process regression (GPR) model on COVID-19 datasets. The experimental result shows that the proposed framework is highly reliable for COVID-19 prediction over the GPR model.

Keywords COVID-19 · RNN · Bi-GRU · CNN · Sequence learning · Containment and health index

1 Introduction
Towards the very end of 2019, the world entered into an unprecedented global crisis in the form of COVID-19, an infection caused by SARS-CoV-2, a new form of SARS coronavirus. Several pandemics have been documented in human history, with pandemic-related crises having had tremendous negative impacts on health, the environment and also causing political and social disruption. The coronavirus epidemic came to light on December 31, 2019, when China notified the World Health Organization about a series of cases of pneumonia of an unexplained origin in Wuhan City, Hubei Province [8]. Eventually, the epidemic spread to more regions in China and the rest of the world. It has now been deemed a pandemic by WHO. The virus was named SARS-CoV-2, and the disease is now renamed COVID-19. COVID-19 is an infectious disease caused by a recently discovered virus named coronavirus. The educational institutes are closed, companies and offices are limiting their staff and their working hours, and the government is restricting the mobility of people as much as possible. While these steps have helped in saving lives and have, therefore, provided a foundation for stronger medium-term growth, they have also led to some unprecedented economic losses in the short term. For the quantification of these short-term economic effects, whether they change across different kinds of containment measures is of supreme importance for many policymakers across the globe who are facing an agonizing short-term trade-off between normalizing economic activity and minimizing health risks. The containment and health index is a robust measure, which is based on eleven policy response indicators, including the closure of schools, closure of workplace, bans on travel, a policy adopted for testing and contact tracing, rescaled to a value from 0 to 100 (100 = strictest). We shall analyze its trend and attempt to forecast its behavior in the near future.

India is the second most populated country with a 1.3 billion population to cover, with average family income listed 112th out of 164 nations by the World Bank and 150th in global health care by the World Economic Forum. The first case has been reported on January 30, 2020. It has been more than 9 months since the first case, and the overall number of confirmed cases has reached a record of 7.86 M as of 24 October 2020. At the same time, 119 K people died [4]. Figure 1 shows the all-time daily COVID-19 confirm cases.
in India, whereas all-time daily death cases due to COVID-19 in India is shown in Fig. 2.

The data collected during the COVID-19 pandemic are an important source of information and can be seen as an opportunity for the researchers to mine useful information from the data. Several research works have been witnessed around the world to forecast and estimate COVID-19 cases. Adaptive neuro-fuzzy inference system (ANFIS)-based model has been proposed by Al-qaness et al. [5] using flower pollination algorithm (FPA) coupled with the salp swarm algorithm (SSA) to predict the next 10 days’ number of cases in China. To predict the number of COVID-19 cases in G-8 countries, Yonar et al. [18] used Box–Jenkins (ARIMA) and Brown/Holt linear, exponential smoothing approaches. Tomar et al. [15] used LSTM model to forecast total number of confirmed cases, daily positive cases, total recovered cases and total death cases in India for the next 30 days. The time-series model based on predictive error minimization (PEM)-based system has been proposed by Rafiq et al. [12] to predict the next 30 days’ number of cases and deaths in the affected states of India. Wang et al. [17] have proposed a hybrid model based on logistic regression and Fbprophet for predicting the COVID-19 trends in Brazil, Russia, India, Peru and Indonesia. Pathan et al. [11] have proposed long short-term memory (LSTM) model to predict the mutation rate of COVID-19. In Arora et al., the AI-based mathematical models proposed by Saba et al. [13] aimed to forecast the daily cases of COVID-19 in Egypt. They used ARIMA and NARANN techniques to design models with a 5-percent forecast error. To study and analyze the COVID-19 spread, a discrete-time stochastic model has been proposed by He et al. [6]. Wang et al. [16] have developed an improved LSTM model for predicting COVID-19 confirmed cases in Russia.

This paper intends to develop a deep learning framework based on the time-series data using convolutional neural network (CNN) and stacked bidirectional gated recurrent unit (Bi-GRU) for the prediction of COVID-19 medium-time period trend in India. To better understand the trend of COVID-19, smoothed time-series data including daily new confirmed cases, daily death cases, daily containment and health index and the daily recovery rate were passed separately into the preprocessing pipeline, which generated structured data considering the context equal to window size (here, 25 days). The processed data were then passed to a deep neural network with two convolutional layers to extract features and then to two stacked Bi-GRU layers for sequence learning. After training, the model achieved significantly less MSE, MAE and MAPE on the testing dataset and successfully projected forecast for the next 30 days with a single step at a time using a context of previously known readings of the last 25 days. Experimental findings show that under different conditions in different states across India, there seems to be a good match between actually recorded daily confirmed cases, death cases, recovered cases and containment and health index values with our predicted one. Our deep learning framework will allow public health practitioners and government to make the appropriate preparations to respond to possible changes in the COVID-19 pattern. The experiments are based on a compilation of confirmed cases of COVID-19 as of October 25, 2020.

The rest of this paper is structured as follows: Section 2 introduces the proposed methodology for COVID-19 forecasting. Section 3 describes the experimental results and discussion based on the COVID-19 epidemic data of India. The conclusion of this article is provided in Sect. 4.

2 Methodology

As discussed, there is a serious need to work on accurate forecasting of COVID-19 so that future enterprises could be planned accordingly without curbing the growth of all the sectors. An attempt to forecast the COVID-19 parameters using a novel deep learning architecture is discussed in this section.
2.1 COVID-19 parameters

To estimate the latent trend of the COVID-19 pandemic in India, data from multiple sources, [3], [1], [2] to reduce sampling bias and noise margin, have been considered on account of the following parameters:

- Daily new cases: To get an estimate on the trend of the count of people getting affected by the pandemic,
- Daily death cases: To get an estimate on the trend of the people losing their lives due to the COVID-19 pandemic,
- Recovery rate: It is defined by the equation,

\[
\text{Recover Rate} = \frac{\text{Total Recovery Cases}}{\text{Total Cases}} \times 100\% \quad (1)
\]

This parameter incorporates an important role in the immunity estimation of the general public and measures the magnitude of how much medical assistance is playing a role in limiting death cases due to COVID-19.
- Daily containment and health index: This is a measure based on eleven policy response indicators, including the closure of schools, closure of workplace, bans on travel, a policy adopted for testing and contact tracing, rescaled to a value from 0 to 100 (100 = strictest).

2.2 Data preprocessing

The data collected posses time-series nature. The pandemic had originated less than a year ago; therefore, the sample size of data points is confined to 250 timestamps. The dataset is normalized by the following definition,

\[
t_i = \frac{t_i}{\max(t)} + \epsilon \quad (2)
\]

where \( t = \{t_1, t_2, ..., t_i, ..., t_{n-1}, t_n\} \) denote the time-series data vector and \( i \) represents the timestamp. \( n \) denotes the total number of data points recorded. The value of \( \epsilon \) was taken to be 1500, for optimized model performance. This parameter has been tuned experimentally using randomized search approach for shrinking of the range of data values. This step plays an important role for achieving optimal performance of the model and for faster training. Figure 3 depicts the entire data preprocessing pipeline.

A 25-unit sampling window is considered for sampling the normalized dataset. This sampling window represents a context of 25 units, i.e., the model refers to the context of the last 25 days from the day of prediction before predicting a new value. This preserves the temporal correlation of the predicted value with its nearest actual values. The sampling window is slid over the entire dataset, sampling 25 values as the input vector and the next data point ahead in the dataset as the target value in each stride. The data of the four COVID variables discussed earlier were fed into this preprocessing pipeline to get a trainable structured dataset of size, \((n - length_{window}) \times length_{window}\) and the dataset of target of size, \((n - length_{window}) \times 1\). We chose \(length_{window}\) to be 25 for better model performance and faster training as too large a window size will introduce noise, while too small a window will lose essential correlations. A training set of the first 85% records in the sequence was considered, while the remaining records were used for testing the model’s generalized behavior to unseen data.

2.3 Model architecture

The preprocessed dataset is fed to the model pipeline as shown in Figs. 4 and 5. The model pipeline inputs a batch of sequence vectors with each sequence vector of size, \(1 \times length_{window}\). The optimal batch size considered for the training was found to be 32 after tuning. Each input sequence vector is reshaped into a matrix of size, \(\sqrt{length_{window}} \times \sqrt{length_{window}}\). A set of \(m\) filter windows each of size, \(k \times \sqrt{length_{window}}\), were convolved over the matrix. The utility of this step is to extract features from the context window itself, which already possesses sequential nature due to spatial correlation getting preserved in convolutional operation. Activation of a rectified linear unit (ReLU) is applied on the feature maps representing local reception. The utility of ReLU layers here is to impart nonlinearity and suppress negative activation from the previous layers maintaining computational efficiency during backpropagation.
To force the network to learn sequences of high-level features, a stacked Bi-GRU layer is used since it performs better than BiLSTMs and BiRNNs and preserves computational efficiency. The following equations represent the operation of the GRU cell. $z_t$ represents the hidden state at time, $t$, which concatenates previous hidden state, $h_{t-1}$ and input at the current timestep, $x_t$. It is accumulated by a dot product with weight matrices, $W_z$. Similarly, $r_t$ is found by a dot product of concatenation of hidden state and current input with the weight matrix, $W_r$. The accumulated contents are passed to the ReLU activation function, which suppresses irrelevant context information from the previous hidden state. The following equations represent the transformations of input tensors, previous hidden state $h_{t-1}$ and current input $x_t$ to produce the output, which also represents the new cell state achieved, $h_t$.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$  \hspace{1cm} (3)

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$  \hspace{1cm} (4)

$$\hat{h}_t = \tanh(W \cdot [r_t \cdot h_{t-1}, x_t])$$  \hspace{1cm} (5)

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \hat{h}_t$$  \hspace{1cm} (6)

In addition to prevent the problem of vanishing gradient, Bi-GRU learns important information from the previous cell state along with current input bidirectionally and drops obsolete context information in a better way than RNNs. Even LSTMs solve the vanishing gradient problem, it involves more computation complexity. GRUs tend to give the same performance with improvement to some extent and with lesser computational overheads. In the last stage of the model pipeline, the vector, $I$ representing sequential and feature encodings from the Bi-GRU layer, is accumulated, and activation of ReLU is applied. The activation from the final layer is the single-step forecast representing the forecast of the next day ahead of the context window. The algorithm 1 shows the pseudo-code of the proposed framework.

### 3 Experimental result and discussion

All the simulations and experiments were carried out on Google Colab with open-source libraries such as NumPy, Pandas, Matplotlib, TensorFlow and Keras. The experimental configuration is based on the operating environment having Intel(R) Core (TM) i5-10300H CPU with processor speed 4.5 GHz, 16 GB RAM, 1 TB SSD, 8 GB NVIDIA GeForce GTX 1650 graphics processor under 64-bit Windows 10 Operating system. Due to the novel COVID situation, the dataset volume is limited for mining deeper information. The proposed methodology is implemented on a dataset pertaining to India. A sufficiently good amount of performance has been observed. Also, a forecast for the next 30 days has been found.
Algorithm 1 Pseudo-code of the proposed framework

Require: DataCOVID, Split Ratio, len_window, m, ε, k, latent_dim
1: NormData ← DataCOVID / (max(DataCOVID) + ε)
2: D ← SampleWithContextWindow(Norm Data, len_window)
3: X, Y ← D
4: Get Training and Testing Data using Split Ratio on X, Y
5: Convert each $x \in XTrain$ into a square matrix and $T ← YTrain$
6: Apply m Conv1D filters to each square matrix to get m feature vectors using Conv1D(k)
7: Initialize $W_z$, $W_r$, $h_0$ of appropriate dimensions using Bi-GRU(latent_dim)
8: for $t ← 1$ to $m$ do
9: Compute $z_t ← \sigma(W_z \cdot [h_{t-1}, x_t])$
10: Compute $r_t ← \sigma(W_r \cdot [h_{t-1}, x_t])$
11: Compute $\hat{Y}_t ← \tanh(W \cdot [r_t h_{t-1}, x_t])$
12: Compute $h_t ← (1 - z_t) * h_{t-1} + z_t * \hat{Y}_t$
13: end for
14: Initialize $WW_z$, $WW_r$, $hh_0$ of appropriate dimensions using Bi-GRU(latent_dim)
15: for $t ← 1$ to $m$ do
16: Compute $zz_t ← \sigma(WW_z \cdot [hh_{t-1}, Y_t])$
17: Compute $rr_t ← \sigma(WW_r \cdot [hh_{t-1}, Y_t])$
18: Compute $\hat{Y}_t ← \tanh(WW \cdot [rr_t hh_{t-1}, Y_t])$
19: Compute $hh_t ← (1 - zz_t) * hh_{t-1} + zz_t * \hat{Y}_t$
20: end for
21: $I ← YY_m$
22: Initialize W of appropriate dimensions using Dense(1, linearActivation)
23: $T ← \text{ReLU}(W \cdot I)$
24: Error ← MSE($T$, $T$)
25: Backpropagation(Error, AdamOptimizer) ⊇ //Weight Updation
26: $Y_{pred} = \text{PretrainedModel}(XTest)$ ⊇ //Using trained model weights, feed XTest to get predictions.
27: Compare $Y_{pred}$ with $YTest$.
28: Return Trained model.

3.1 Dataset description

The COVID-19 parameters considered for the experimental result pertain to India in particular. To estimate the latent trends of the COVID-19 pandemic in India, we considered data, [3], [1], [2], having the following schema:

- Daily new cases: This time-series measure represents the count of new COVID-19 cases reported daily. A total of 250 data points have been considered, which were smoothened and refined by truncating redundant initial zeros for creating an improved generalized model. It is implicitly assumed that the number of new cases before data collection was zero, and therefore, the zeros in the head were truncated. The maximum value in this data was 93198.571 after smoothing, which is an important parameter for normalization, as discussed in the methodology.

- Daily death cases due to COVID-19: This parameter estimates the trend of people losing their lives due to the pandemic. A total of 250 data points have been consid-
ered, which were then smoothened, and initial zeros were truncated as they were redundant. It is implicitly assumed that the number of death cases before data collection was zero, and so the zeros in the head were removed. The maximum value in this data was 1168 after smoothing, which is used for normalization as discussed in the methodology.

- Recovery rate: This measure plays an important role in estimating the immunity of the people in general and estimating the effectiveness of medical campaigns in limiting death cases due to COVID-19. A total of 250 data points have been considered, of which initial zeros were truncated as they were redundant. It is implicitly assumed that the recovery rate before data collection was zero, and so the zeros in the head were ignored. The maximum value in this data was 6663608, which is used for normalization, as discussed in the methodology.

- Daily containment and health index: This time-series measure is based on eleven policy response indicators, including the closure of schools, closure of workplace, bans on travel, a policy adopted for testing and contact tracing, which is rescaled to a value from 0 to 100 (100 = strictest). A total of 250 data points have been considered, of which initial zeros were truncated as they were redundant. It is implicitly assumed that the index values before data collection were zero, and so the zeros in the head are removed. The maximum value in this data was 100, which is used for data normalization, as discussed in the methodology.

3.2 Forecast result with the proposed framework

In this section, the notable results of the randomized search hyperparameter tuning process are discussed. The batch size for training the model was taken to be 8, 16 and 32, respectively. Initially, the loss function chosen was MAE but later switched to MSE since lower MSE proved to have a better MAE as well. The optimizer considered for the training of the forecast model was Adam since it incorporates the benefits of RMSProp with the added advantage of momentum and acceleration. Figure 6 represents the details of the layers used in the proposed model.

The two important hyperparameters were $\epsilon$ and the number of epochs. A forecast for the next 30 time-steps is obtained based on a window of the previous 25 days with a single-point prediction at a single time instance, and then, using this prediction as input and sliding the window, we produce the next forecast. Repeating this process produces a forecast for the next 30 days. Figure 7a represents the actual and learnt trend of new cases of COVID-19 in India, Fig. 7b represents the actual and learnt trend of deaths cases in India due to COVID-19, Fig. 7c represents the actual and learnt trend of containment and health index and Fig. 7d rep-
Fig. 6 Proposed model layer details

Fig. 7 Time-series predictions of COVID-19 for India with $\epsilon = 0$ and $\text{epochs} = 100$ a New cases b New deaths c Containment and health index d Recovery rate

Fig. 8 Time-series predictions of COVID-19 for India with $\epsilon = 1500$ and $\text{epochs} = 500$ a New cases b New deaths c Containment and health index d Recovery rate

Fig. 9 Time-series predictions of COVID-19 for India using GPR model a new cases b new deaths c containment and health index d recovery rate

3.3 Forecast results with GPR model

GPR is a nonparametric regression model that is widely used in predictive analysis [9]. To predict the trends for COVID-19 in India for the next 30 days, the GPR model is tested on the datasets as mentioned above. Figure 9a shows the actual smoothed new cases COVID-19 data for India and prediction from the GPR model. Figure 9b shows the actual smoothed new deaths COVID-19 data for India and prediction from the GPR model. Figure 9c shows the actual recovery rate COVID-19 data for India and prediction from the GPR model. Figure 9d shows the actual containment and health index COVID-19 data for India and prediction from the GPR model.
### 3.4 Performance evaluation

The performance metrics, such as mean-squared error (MSE), mean absolute error (MAE) and mean absolute percentage error (MAPE), are taken into consideration to check the performance of the proposed model. The loss is computed using the MSE, and it is defined as,

\[
MSE = \left( \frac{1}{N} \right) \sum_{i=1}^{N} (y_i - \hat{y}_i)^2
\]  

(7)

MAE is defined as,

\[
MAE = \left( \frac{1}{N} \right) \sum_{i=1}^{N} |y_i - \hat{y}_i|
\]  

(8)

MAPE is defined as

\[
MAPE = \left( \frac{100\%}{N} \right) \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right|
\]  

(9)

Table 1 shows the metrics of the model on the testing dataset. Since the dataset has been scaled down, causing the metrics values to also shrink due to which the costs have approached close to zero, but still the performance can be evaluated with these data accurately. Considering the MAPE values from the table, we observe that there has been a significant improvement with an increase in the normalization parameter, \(\epsilon\) and number of epochs. The metrics mentioned in the table and the projected forecast are consistent enough and can be visualized easily. The trend of the actual dataset and the forecast is promising and therefore gives a sense of confidence in the model’s quality.

### 3.5 Performance comparison

In this part, the proposed model is compared with the GPR model for COVID-19 predictive analysis. The comparison of the proposed model (with 700 epochs) with the GPR model is shown in Table 2. The proposed model is found better than the GPR model, which can be verified by the value of accuracy measures (MSE, MAP and MAPE). For new death cases smoothed dataset and containment and health index dataset, the proposed model is better with respect to MSE and MAE, while it is close enough with regard to MAPE. For the new cases smoothed dataset, the GPR model is a little better but close enough with respect to MSE, MAE and MAPE. For the recovery rate dataset, the proposed model is better with respect to MAPE, while GPR is a little better but close enough with respect to MSE and MAE.

### 4 Conclusion

COVID-19 has almost interrupted the world because of the destructive nature of the pandemic and spreading rapidly across the globe. This paper contributes to the research work on COVID-19, which can forecast the epidemic trends of COVID-19 (next 30 days) in India using a deep learning framework. The forecast projected by the CNN and Bi-GRU combo model proposed in this work shows a rise in the new positive cases, death cases and containment index, and the recovery rate. Also, a little possibility of stabilization or flattening in the trend can be observed. The predicted result can help the government authorities to manage and prepare for the future. The simulation results show that the pro-

---

**Table 1** Performance evaluation of the proposed model for COVID-19 predictions

| Data  | \(\epsilon\) | EP | MSE     | MAE     | MAPE  |
|-------|--------------|----|---------|---------|-------|
| NCS   | 0            | 100| 0.0060  | 0.0678  | 7.5288|
| NDS   | 0            | 100| 0.0028  | 0.0469  | 5.3654|
| RR    | 0            | 100| 0.0032  | 0.0477  | 5.6868|
| CHI   | 0            | 100| 3.1102e-07| 5.4378e-04| 1.0746|
| NCS   | 1500         | 500| 0.0017  | 0.0378  | 4.1077|
| NDS   | 1500         | 500| 1.4257  | 0.9303  | 1.2071|
| RR    | 1500         | 500| 3.6037e-05| 0.0047  | 1.1468|
| CHI   | 1500         | 500| 7.0876e-05| 0.0076  | 0.9764|
| NCS   | 1500         | 700| 1.45E-05| 0.0025  | 60.80 |
| NDS   | 1500         | 700| 4.05E-05| 0.0037  | 3.4828|
| RR    | 1500         | 700| 2.37E-06| 0.0012  | 2577.7946|
| CHI   | 1500         | 700| 1.741E-06| 8.91E-04| 2.6483|

\(\epsilon\): epsilon; \(EP\): epochs; \(NCS\): new cases smoothed; \(NDS\): new deaths smoothed; \(RR\): recovery rate; \(CHI\): containment & health index

**Table 2** Comparison of the proposed model (epochs=700) for COVID-19 predictions against GPR model

| Data  | Metric  | GPR  | Proposed model |
|-------|---------|------|----------------|
| NCS   | MSE     | 6.45E-06| 1.45E-05         |
|       | MAE     | 0.00157| 0.0025          |
|       | MAPE    | 58.77  | 60.80          |
| NDS   | MSE     | 7.00E-05| 4.05E-05       |
|       | MAE     | 0.00373| 0.0037         |
|       | MAPE    | 3.139  | 3.4828       |
| RR    | MSE     | 2.19E-07| 2.37E-06       |
|       | MAE     | 0.00027| 0.0012        |
|       | MAPE    | 9566.469| 2577.7946    |
| CHI   | MSE     | 1.748E-06| 1.741E-06   |
|       | MAE     | 0.00058| 8.91E-04     |
|       | MAPE    | 0.4018  | 2.6483       |

\(NCS\): new cases smoothed; \(NDS\): new deaths smoothed; \(RR\): recovery rate; \(CHI\): containment & health index
posed framework outperforms the GPR model with better accuracy for COVID-19 prediction. In the future, we plan to predict the COVID-19 trends for other countries across the world. It opens possibilities for the application of various state-of-the-art time-series and sequence learning models like bidirectional encoder representations from transformers (BERT) and generative pretrained transformer (GPT) to be used for such use-cases.

References

1. Coronavirus government response tracker. available [online]: https://www.bsg.ox.ac.uk/research/research-projects/coronavirus-government-response-tracker. (Accessed 10 Oct 2020)
2. Covid-19: Government response stringency index. available [online]: https://ourworldindata.org/grapher/covid-stringency-index. (Accessed 10 Oct 2020)
3. Humanitarian data exchange. available [online]: https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-cases. (Accessed 10 Oct 2020)
4. Ministry of home a situation report: Government of india 2020. available [online]: https://www.mohfw.gov.in/. (Accessed 25 Oct 2020)
5. Al-Qaness, M.A., Ewees, A.A., Fan, H., Abd El Aziz, M.: Optimization method for forecasting confirmed cases of covid-19 in china. J. Clin. Med. 9(3), 674 (2020)
6. He, S., Tang, S., Rong, L.: A discrete stochastic model of the covid-19 outbreak: Forecast and control. Math. Biosci. Eng 17, 2792–2804 (2020)
7. Kapoor, A., Ben, X., Liu, L., Perozzi, B., Barnes, M., Blais, M., O’Banion, S.: Examining covid-19 forecasting using spatio-temporal graph neural networks. arXiv preprint arXiv:2007.03113 (2020)
8. Li, H., Liu, L., Zhang, D., Xu, J., Dai, H., Tang, N., Su, X., Cao, B.: SARS-CoV-2 and viral sepsis: observations and hypotheses. Lancet 375, 1517–1520 (2020)
9. Liu, H., Ong, Y.S., Shen, X., Cai, J.: When Gaussian process meets big data: a review of scalable GPs. IEEE Trans. Neural Netw. Learn. Syst. 31(11), 4405–4423 (2020)
10. Melin, P., Monica, J.C., Sanchez, D., Castillo, O.: Multiple ensemble neural network models with fuzzy response aggregation for predicting covid-19 time series: the case of mexico. In: Health-care, vol. 8, p. 181. Multidisciplinary Digital Publishing Institute (2020)
11. Pathan, R.K., Biswas, M., Khandaker, M.U.: Time series prediction of covid-19 by mutation rate analysis using recurrent neural network-based lstm model. Chaos, Solitons & Fractals p. 110018 (2020)
12. Rafiq, D., Suhail, S.A., Bazaz, M.A.: Evaluation and prediction of COVID-19 in India: a case study of worst hit states. Chaos, Solitons Fract. 139, 110014 (2020)
13. Saba, A.I., Elsheikh, A.H.: Forecasting the prevalence of COVID-19 outbreak in Egypt using nonlinear autoregressive artificial neural networks. Process Saf. Environ. Protect. 141, 1–8 (2020)
14. Said, A.B., Erradi, A., Aly, H., Mohamed, A.: Predicting COVID-19 cases using bidirectional LSTM on multivariate time series. arXiv preprint arXiv:2009.12325 (2020)
15. Tomar, A., Gupta, N.: Prediction for the spread of COVID-19 in india and effectiveness of preventive measures. Sci. Total Environ. 728, 138762 (2020)
16. Wang, P., Zheng, X., Ai, G., Liu, D., Zhu, B.: Time series prediction for the epidemic trends of COVID-19 using the improved LSTM deep learning method: Case studies in russia, peru and iran. Chaos, Solitons Fract. 140, 110214 (2020)
17. Wang, P., Zheng, X., Li, J., Zhu, B.: Prediction of epidemic trends in COVID-19 with logistic model and machine learning technics. Chaos, Solitons Fract. 139, 110058 (2020)
18. Yonar, H., Yonar, A., Tekindal, M.A., Tekindal, M.: Modeling and forecasting for the number of cases of the COVID-19 pandemic with the curve estimation models, the box-jenkins and exponential smoothing methods. EJMO 4(2), 160–165 (2020)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.