

# COVID-19 Growth Curve Forecasting for India Using Deep Learning Techniques



V. Vanitha and P. Kumaran

## Introduction

The global spread of the COVID-19 pandemic has resulted in considerable number of losses of life. It has been regarded as the world's largest economic and health disaster since World War II. According to the World Health Organization (WHO), the SARS-CoV-2 virus has infected approximately 200 million individuals globally. The virus has been shown to spread between persons via respiratory channels during human movement, enhancing its transmissibility and making the whole population vulnerable. COVID-19 confirmed cases numbered 4,799,266 on May 17th, while 316,520 persons died as a result of the pandemic at global level. Due to the lack of vaccine and drugs at the initial outbreak, different countries have taken varied approaches to contain the outbreak. The most usual responses include strict lockdown, partial lockdown, the closure of all educational institutions, and the cancellation of all sorts of aircraft. Because of the link between human movement and viral transmissibility, governments throughout the world have implemented restrictions, such as mandatory face mask, social distancing, and shutting public transit and restaurants, to avoid crowds. Although the implementation of such regulations has slowed the spread, the emergence of lethal mutations continued to put public health at risk.

---

V. Vanitha (✉)

Sri Ramachandra Faculty of Engineering and Technology, Sri Ramachandra Institute of Higher Education and Research, Porur, Chennai, Tamil Nadu, India  
e-mail: [vanikkdi@gmail.com](mailto:vanikkdi@gmail.com)

P. Kumaran

Department of Computer Science and Engineering, National Institute of Technology Karaikal, Karaikal, Puducherry, India  
e-mail: [kumaran.p@nitpy.ac.in](mailto:kumaran.p@nitpy.ac.in)

© The Author(s), under exclusive license to Springer Nature  
Switzerland AG 2023

G. R. Kanagachidambaresan et al. (eds.), *System Design for Epidemics Using Machine Learning and Deep Learning*, Signals and Communication Technology, [https://doi.org/10.1007/978-3-031-19752-9\\_18](https://doi.org/10.1007/978-3-031-19752-9_18)

305

Medical supplies are frequently in low supply due to rising patient numbers, placing a strain on healthcare systems and personnel in many nations. Thus, one of the most important factors to contain and control the spread is understanding the nature of the spread and accurately projecting the patterns. Reliability in forecasting COVID-19 spread trends can aid in the prediction of pandemic outbreaks and boost government readiness to combat the pandemic. Furthermore, precise forecasting can offer feedback on whether the implemented strategy helps reduce the burden on the country's healthcare system.

Such a tumultuous environment of epidemic breakouts sparked numerous broad questions where there is a definite answer: Would coronavirus endure until a vaccine is discovered, or will it be eradicated after a set length of time? How long does it take a medical expert to develop the correct drug or vaccine? What is the estimated number of individuals who will be affected by this epidemic? What is the likelihood of death or recovery among the afflicted patients? Is it different in different age groups and different parts of the world? If that's the case, what may be the reasons? How effective is the lockdown approach at reducing the spread? What are the negative consequences of lockdown, and how long can various countries afford it?

In the last decade, machine learning (ML) has established itself as a distinct academic subject by tackling a slew of extremely complicated and sophisticated real-world challenges. There is extant research that attempts to forecast death daily using the traditional and deep learning methods, like long short-term memory (LSTM) and its variants. The mean squared error (MSE) and absolute error (MAE) score are commonly used to evaluate the prediction capabilities of the models. Recurrent neural network, a type of deep learning, is used in this study to anticipate the pandemic trend for India by forecasting the number of new cases. The reason for choosing India is due to the fact that it is one of the top 10 severely afflicted countries in the world, according to healthcare professionals. Furthermore, the LSTM model built beats several previously published models; therefore, the work utilizes it to anticipate COVID-19 instances a week in advance.

The rest of the paper is organized as follows: section “[Literature review](#)” discusses related research in this field, section “[Materials and methods](#)” describes the dataset and details the proposed system, section “[Results and discussion](#)” discusses the experimental results, and section “[Conclusion](#)” concludes the study.

## Literature Review

Machine learning models are widely employed to understand the COVID-19 pandemic from various medical perspectives, including understanding the impact of antibodies [1], chest X-rays and chest CT images [2, 3], mutations [4], and forecasting pandemic trends.

The authors of this study [5] focused on predicting the number of COVID-19 cases that will be confirmed, recovered, or died in 60 days in the 16 high-impact nations. They used a seasonal auto-regressive integrated moving average (SARIMA) and an auto-regressive integrated moving average (ARIMA) models. According to their study, the SARIMA model is more realistic than the ARIMA model. Da Silva et al. [6] compared the univariate ARIMA and a proposed hybrid model that examine the number of illnesses in the top 27 afflicted cities in Brazil. Their experiments demonstrated that the ensemble model outperformed the single model by 26.73%.

Researchers have indicated a strong desire to learn more about India's rapid expansion. Swaraj et al. [7] built a model for predicting the COVID-19 epidemic in India that used ARIMA and a nonlinear auto-regressive neural network (NAR). When compared to the single ARIMA model, the hybrid model exhibits a considerable reduction in evaluation metrics. Wadhwa et al. [8] forecast the number of active cases across India 3 months ahead using the linear regression (LR) model. Khan et al. [9] implemented various machine learning models to determine when will the number of cases in India stop growing and to examine policy restrictions. According to their findings, the GPR model surpasses the other models with an accuracy of 95 percent. Using daily fresh confirmed cases in Russia, Peru, and Iran, Wang et al. [9] created an LSTM model to estimate pandemic trends for 150 days. Bayesian model was used on publicly available global data to assess the impact of lockdowns on COVID-19 transmission for five nations with high covid incidence (India, Brazil, Russia, the USA, and the UK). It has been established that if the lockdowns are lifted, the outbreak tempo in Brazil, India, and Russia would considerably rise.

In [10], an auto-regression model was used to predict confirmed and recovered COVID-19 cases in Jakarta. With an MPAAE value of less than 20%, the results suggest that this technique delivers adequate forecasting accuracy. When compared to traditional approaches, such as ARIMA, exponential smoothing, BATS, and Prophet, this methodology performed better for pandemic prediction. However, the prediction quality of the Poisson auto-regression technique still has to be improved to achieve good prediction performance. ARIMA, MLP, LSTM, and feedforward neural network (FNN) are four regression models used to forecast COVID-19 spread in [11]. The LSTM model was shown to have the highest forecast accuracy in this investigation.

In [12], a few machine learning models, including susceptible-infected-recovered, linear regression, polynomial regression, and SVR and LSTM, are examined in projecting COVID-19 cases in Saudi Arabia and Bahrain. When utilizing confirmed COVID-19 cases data from Saudi Arabia, the results show that SVR offers the greatest predicting, whereas LR surpasses the other models with Bahrain verified cases data.

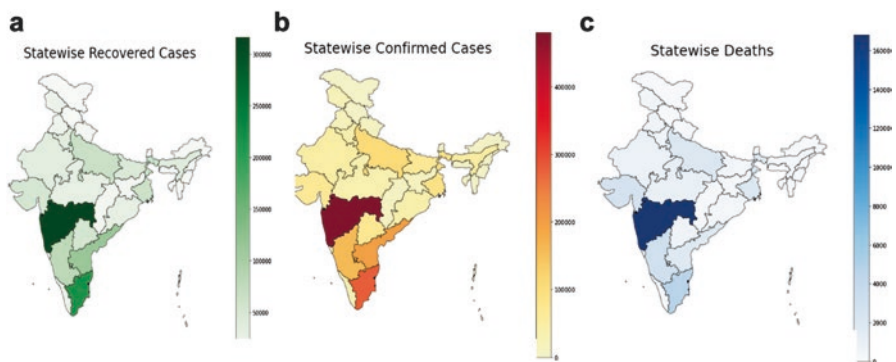
## Materials and Methods

### *Description of Dataset*

The data for this study was taken from the government of India official website <https://www.mohfw.gov.in/>. The dataset contains information about the newly confirmed COVID-19 cases, cured cases, and deaths for each day for each state. The confirmed cases, cured cases, new cases, and death are updated by the Ministry of Health and Family Welfare (MoHFW), India, on a regular basis. The website provides state-wise statics of all aforementioned parameters. In the dataset, daily COVID-19 statistics are available for 560 days from January 30, 2020, till August 11, 2021. It contains 18,110 corona records observed for different states at different days. This dataset has been used to analyze the state-wise trend. The data from August 12, 2021 till date of this article was fetched from coronavirus research center of John Hopkins University available at GitHub site and are updated daily. The records are split into 65:35 for training and validation; records of 450 days are used for training, and remaining records are utilized for validation. A time step of seven is considered as the spread of covid is significant from 1 week to another week. COVID-19 statistics plots from data taken from MoHFW are shown in Fig. 1. Figure 2 depicts the total confirmed, recovered, active cases and deaths for each state. Figure 3 shows the top ten states with the highest confirmed cases. Figure 1a–c displays the heatmap plot of confirmed, recovered, and deaths for each state in India.

### *Forecasting COVID-19 with Recurrent Neural Network*

The analysis of underlying patterns in time series data has seen as key way to solve a series of forecasting problems, like stock market forecasting, traffic planning and management, and weather prediction. In healthcare applications, time series



**Fig. 1** Plots from dataset. (a) State wise recovered cases, (b) state wise confirmed cases, (c) state wise deaths

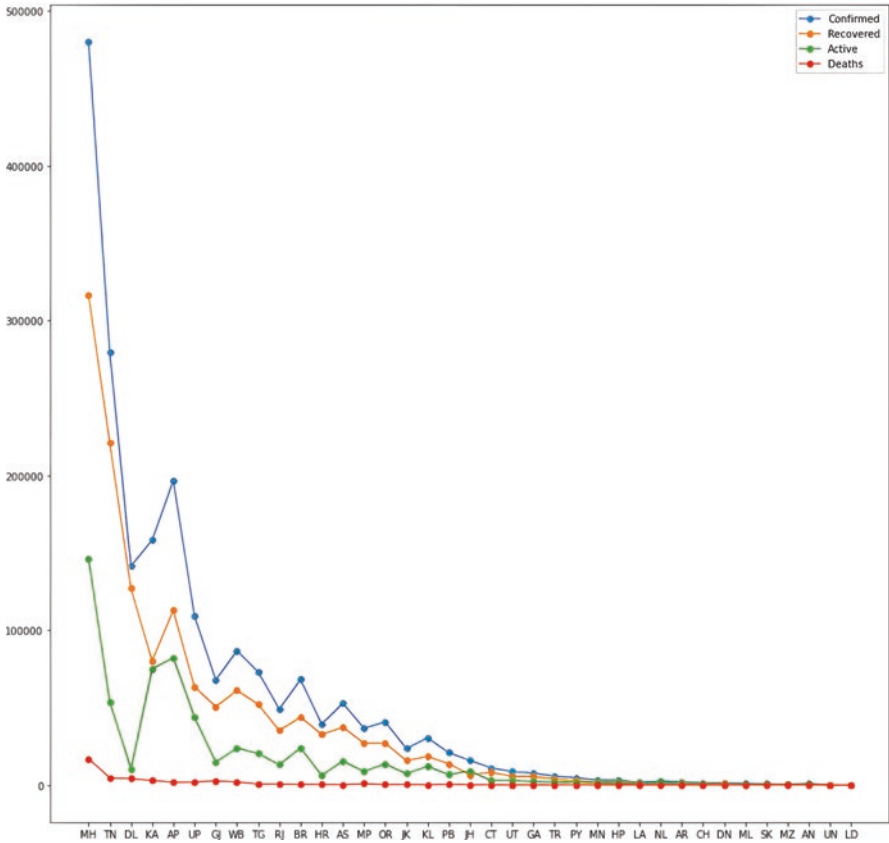


Fig. 2 Total infected cases till August 11, 2021

forecasting model is used to predict the spread of disease, estimate survival and mortality rate, and evaluate the possible risk caused by disease over time.

For short-term forecasting, conventional time series models, e.g., ARIMA and exponential smoothing, are appropriate. Long-term forecasting involves unearthing the underlying trends of the data and the effect of the association among the related parameters to provide estimates for future [13]. As they demand tremendous computations, conventional techniques were limited in their ability in terms of high-dimensional data and complex nature of functions [14].

Currently, deep learning models have been widely employed in forecasting problems [15], owing to its nature to learn the mapping of the input-output pair and support multiple inputs and outputs. Specifically, recurrent neural networks (RNNs) pose the ability to handle the sequence dependency that exists between inputs. However, for any standard RNN, weights on the hidden layers and output layers would either decay or explode. To tackle this gradient problem, long short-term memory (LSTM) has been designed and have been employed successfully in various domains [16].

	State/UnionTerritory	Confirmed	Deaths	Cured	Active
0	Maharashtra	1928603	49463	1824934	54206
1	Karnataka	918544	12081	894834	11629
2	Andhra Pradesh	881948	7104	871588	3256
3	Tamil Nadu	817077	12109	796353	8615
4	Kerala	755718	3042	687104	65572
5	Delhi	624795	10523	608434	5838
6	Uttar Pradesh	584966	8352	562459	14155
7	West Bengal	550893	9683	528829	12381
8	Odisha	329306	1871	325103	2332
9	Rajasthan	307554	2689	295030	9835

Fig. 3 Top ten states with respect to active cases (till August 11, 2021)

```

ADF Statistic: 0.627317
p-value: 0.988270
Critical Values:
    1%: -3.438
    5%: -2.865
    10%: -2.569
Failed to Reject Ho - Time Series is Non-Stationary
    
```

Fig. 4 ADF statistics for the COVID-19 dataset

### *ADF (Augmented Dickey-Fuller) Test*

The time series forecasting model a stationary time series data for better prediction. So, as the preliminary step, we checked the nature of the dataset used in the study using the augmented Dickey-Fuller (ADF) test. The results of the test are interpreted based on the p-values. The ADF test was performed on the covid dataset and found to be nonstationary as the p-value is over 5% as shown in Fig. 4.

In order to make the dataset stationary, lag 1 difference was performed on the dataset. The ADF statistics after lag difference is shown in Fig. 5.

ADF Statistic: -4.160818  
 p-value: 0.000767  
 Critical Values:  
     1%: -3.438  
     5%: -2.865  
     10%: -2.569  
**Reject Ho - Time Series is Stationary**

Fig. 5 ADF statistics for the COVID-19 dataset after performing lag difference technique

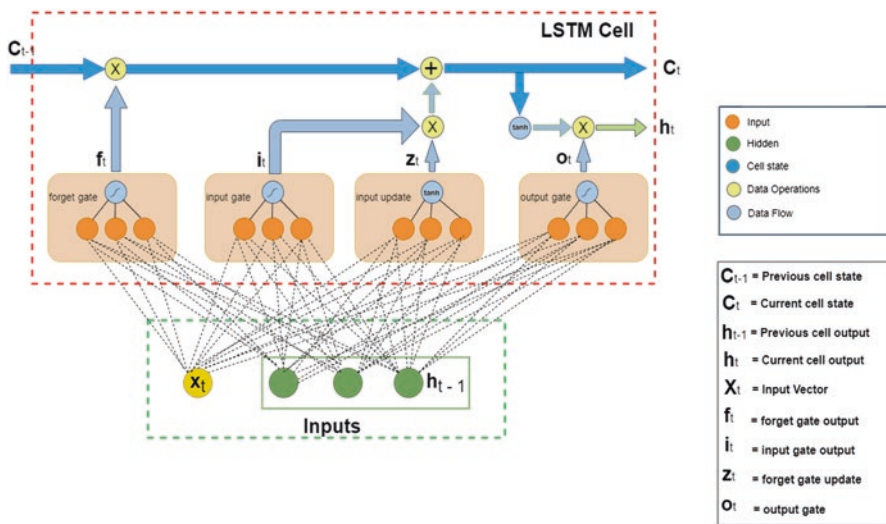


Fig. 6 LSTM cell

### LSTM

RNN is the key deep learning technique on time series data to extract temporal correlations hidden in the data [17]. It has one to many hidden states distributed in the temporal way and can forecast the future with good accuracy than traditional methods [18–20]. The major disadvantage with this method is its inability to overcome vanishing gradient problem [21]. To address this shortcoming, LSTM was developed, which regularizes the gradient flow [16]. Long short-term memory is a recent variant of recurrent neural network to resolve exploding and vanishing gradient problems. LSTMs are capable of learning long-range dependencies hidden in the data through memory cells (LSTM cells). The dissection of LSTM cell is shown in Fig. 6.

These dependencies and temporal correlation of the input are captured in the LSTM cell through the series of gates, viz., forget gate, input gate, and output gate,

along with the sigmoid and tangent activation function. The computation at each gate in LSTM cell is shown in the equations below [22].

$$\text{Input gate } i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}C_{t-1} + b_i) \tag{1}$$

$$\text{Forget gate } f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}C_{t-1} + b_f) \tag{2}$$

$$\text{Output gate } o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}C_t + b_o) \tag{3}$$

$$\text{Cellstate } c_t = \sigma(f_t c_{t-1} + i_t \tan h(W_{xc}x_t + W_{hc}h_{t-1} + b_c)) \tag{4}$$

$$\text{Hidden state } h_t = o_t \tan h(c_t) \tag{5}$$

where  $\sigma$  represents sigmoid function and  $\tan h$  is tangent function. In this paper, variants of LSTM are implemented and are discussed in the following sections.

### Stacked LSTM

In stacked LSTM, multiple LSTM layers stacked together as depicted in Fig. 7. Each intermediate LSTM output layer provides a sequence of outputs which is fed to the next LSTM layer. Also, it provides output for each time step rather than a one

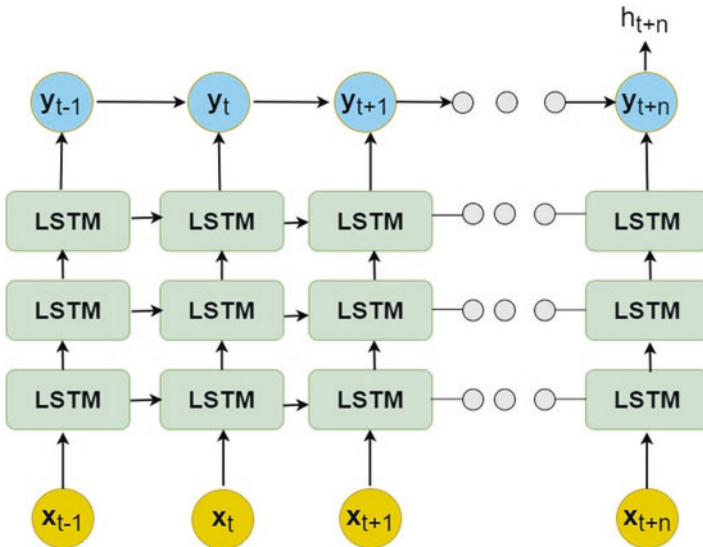


Fig. 7 Stacked LSTM



output for all input time steps. The computation at each stage is given in Eqs. (6), (7), (8), (9) and (10) [22].

$$\text{Input gate } i_t^L = \sigma(W_{ih}^L h_{t-1}^L + W_{ix}^L h_t^{L-1} + b_i^L) \tag{6}$$

$$\text{Forget gate } f_t^L = \sigma(W_{fh}^L h_{t-1}^L + W_{fx}^L h_t^{L-1} + b_f^L) \tag{7}$$

$$\text{Output gate } o_t^L = \sigma(W_{oh}^L h_{t-1}^L + W_{ox}^L h_t^{L-1} + b_o^L) \tag{8}$$

$$\text{Cell-state } c_t^L = (f_t^L W_{c_{t-1}}^L + i_t^L c_{t-1}^{L-1}) \tag{9}$$

$$\text{Hidden state } h_t^L = o_t^L \tan h(c_t^L) \tag{10}$$

### Bidirectional LSTM

Unlike LSTM, which can process inputs only in the forward direction, bidirectional LSTM uses information from both directions (from future to past and from past to future) as shown in Fig. 8.

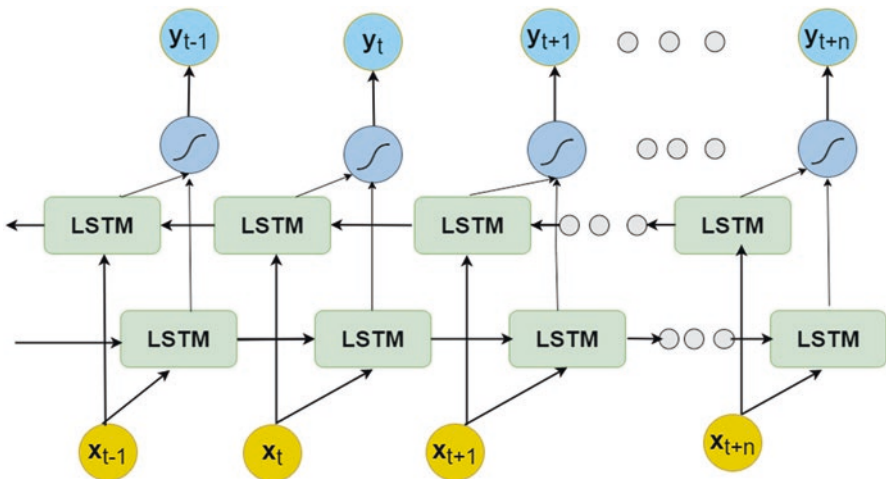


Fig. 8 Bidirectional LSTM

The computation at each stage for producing output is given below [22].

$$\text{Input gate } i_t^{\leftarrow L} = \sigma \left( W_{\leftarrow ih}^L h_{t-1}^L + W_{\leftarrow ix}^L h_t^{L-1} + b_{\leftarrow i}^L \right) \quad (11)$$

$$\text{Forget gate } f_t^{\leftarrow L} = \sigma \left( W_{\leftarrow fh}^L h_{t-1}^L + W_{\leftarrow fx}^L h_t^{L-1} + b_{\leftarrow f}^L \right) \quad (12)$$

$$\text{Output gate } o_t^{\leftarrow L} = \sigma \left( W_{\leftarrow oh}^L h_{t-1}^L + W_{\leftarrow ox}^L h_t^{L-1} + b_{\leftarrow o}^L \right) \quad (13)$$

$$\text{Cell-state } c_t^{\leftarrow L} = (f_t^{\leftarrow L} W c_{t-1}^{\leftarrow L} + i_t^{\leftarrow L} c_t^{\leftarrow L-1}) \quad (14)$$

$$\text{Hidden state } h_t^{\leftarrow L} = o_t^{\leftarrow L} \tan h \left( c_t^{\leftarrow L} \right) \quad (15)$$

The output of the network is the cumulative outputs from both directions and is given by

$$\text{Output } y_t = W_{hy}^{\leftarrow} h_t^{\leftarrow} + W_{hy}^{\rightarrow} h_t^{\rightarrow} + b_y \quad (16)$$

## *The Proposed Models*

### **LSTM Model**

Three LSTM models are built for this study and experimented on the dataset. The first model based on stacked LSTM is shown in Fig. 9. It has an input layer, two LSTM hidden layers, a fully connected layer, and an output layer. The input time sequence is set to 7, considering the significance of a week of COVID-19 data. Both the first and second LSTM hidden layers have 150 units and a rectified linear unit (ReLU) activation function. The fully connected layer is designed with 64 neurons, and the final output layer has a dense layer with 1 neuron. The proposed second LSTM model is similar to the first model with an additional dropout layer after the first hidden LSTM layer (dropout probability 0.5).

The hyperparameters set for both models are summarized in Table 1.

### **Bidirectional LSTM Model**

A bidirectional LSTM model with architecture as shown in Fig. 10 was implemented. It has an input layer, two bidirectional LSTM hidden layers, a fully connected layer, and an output layer. The input time sequence is set to 7 as that of stacked LSTM model. Both the first and second LSTM hidden layers are an LSTM layer with 300 units and a rectified linear unit (ReLU) activation function. The fully connected layer is designed with 150 neurons, and the final output layer had a dense layer with 1 neuron.

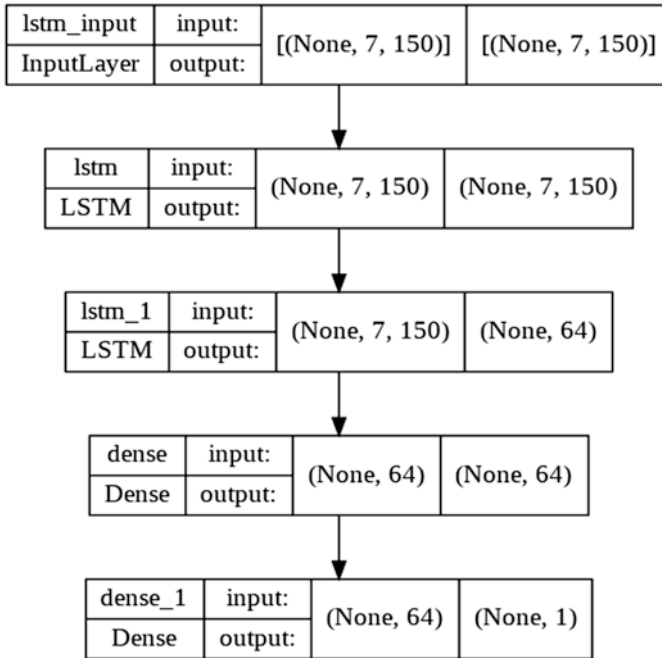


Fig. 9 Proposed stacked LSTM model

Table 1 Hyperparameters of the proposed model

Parameters	Value
Time step	7
Activation function	ReLU
Batch size	32
Epochs	150
Optimizer	Adam
Loss function	Squared error loss
No. of fully connected layers	1 (64 neurons)
Dropout probability (stacked LSTM+ dropout model)	0.5

## Results and Discussion

In this section, the performance of three proposed models on Indian covid dataset is discussed. Three variants of LSTM models, namely, stacked LSTM, stacked LSTM + dropout, and bidirectional LSTM, are built and experimented on the dataset. Each model has been trained using the same dataset and evaluated by the same validation dataset. The forecasting for all proposed models is based on the attribute confirmed

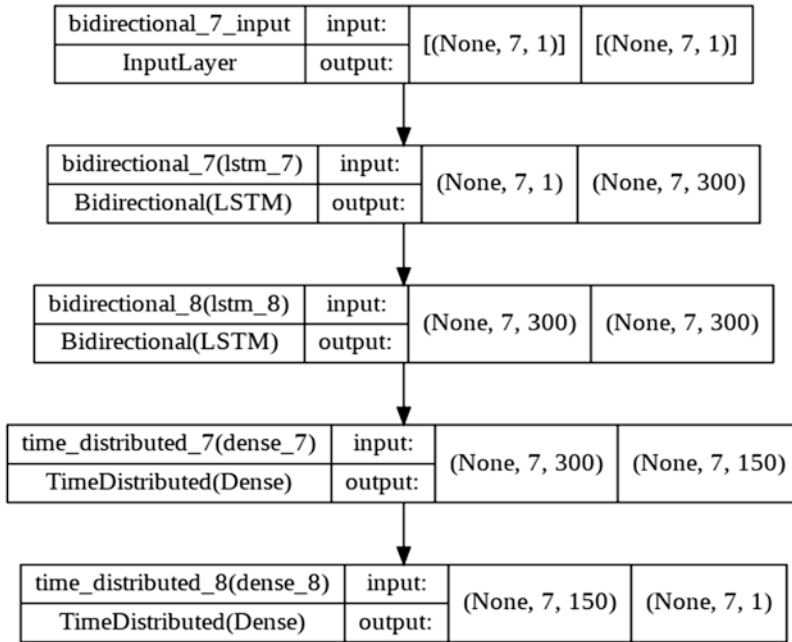


Fig. 10 Proposed bidirectional LSTM model

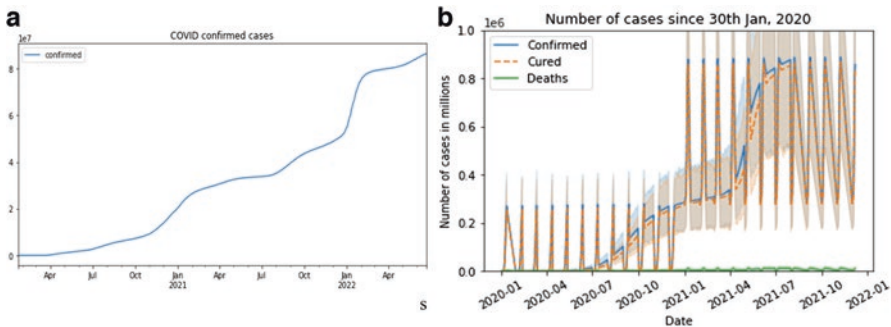


Fig. 11 Confirmed cases (a) for 1 year period, (b) for 2 years period

cases in the dataset. The confirmed covid cases plotted for 1 year period (2021–2022) and 2 years period (2020–2022) are shown in Fig. 11a, b, respectively.

From Fig. 11a, rise is event at two periods: January 2021 peaking in March 2021 and January 2022 peaking in February 2022. These two peaks indicate the second and third wave in India, respectively. The second wave in India started in January 2021, peaked in March 2021, and declined in June 2021. The third wave driven by the Omicron variant started in January 2022, peaked in February 2022, and declined

in March 2022. The start of the first wave (January 2020) and its trend for 2 years period of 2020–2022 can be interpreted in Fig. 11b.

The trend of active, cured, and death for top three states, namely Maharashtra, Karnataka, and Tamil Nādu for the period of 2 years from 2020 to 2022 are shown in Fig. 12. These three states top the list of severely affected states. Though the number of active, cured, and death cases vary for each state, they exhibit more or less same trend throughout the 2-year period.

The proposed three models, namely, stacked LSTM model, LSTM with dropout layer, and bidirectional model, are built on training dataset and checked against the validation dataset. All the experiments are conducted using a 16GB graphics processing unit and Keras framework with TensorFlow back end.

The proposed models are trained on the dataset for different epoch sizes of 50,100,150 and 500. The training loss and validation loss for three models at epoch size = 150 are given in Figs. 13a, b and 14, respectively. The loss plot curve can be used to interpret the performance of the model whether are underfit, overfit, or perfectly fit the data. Underfitting models have high bias, meaning that training loss will not decrease with increase in data. It indicates that the model is not able to learn from the training data. On the other hand, overfitting indicates high variance. The model can perform well on the training data, but poor on the unseen data. It means that model cannot generalize well.

The training-validation loss plot of stacked LSTM revealed that both training loss and validation loss are high with smaller training samples. As the samples are increased, both the losses came down. More to that, both the losses follow the same path, and distance between them is less. It indicated that the proposed stacked LSTM model shows good fit on data and can generalize well on the unseen data.

The training-validation loss plot of bidirectional LSTM revealed that validation loss is very high than training loss and shoots up at several batches of datapoints. This indicates that the proposed bisectinal model has overfitting problem and not able to generalize on new data.

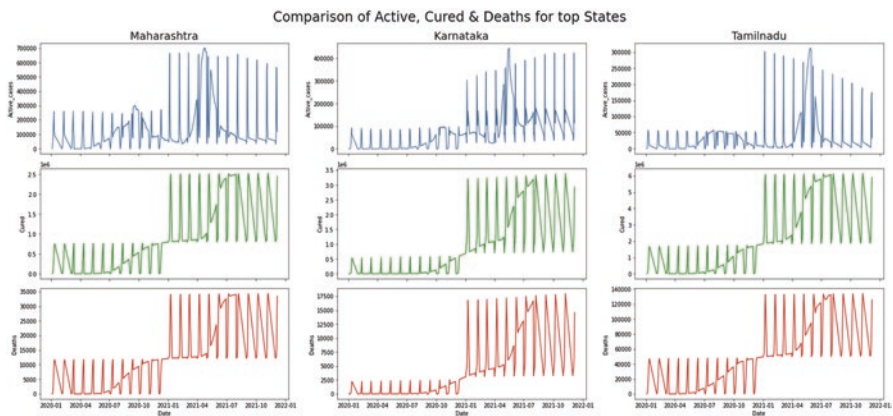
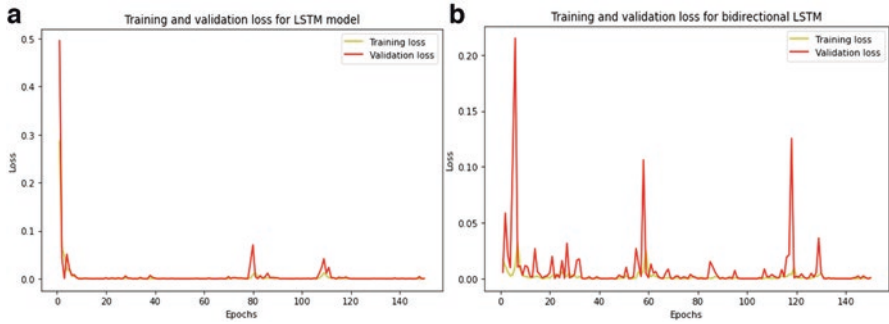
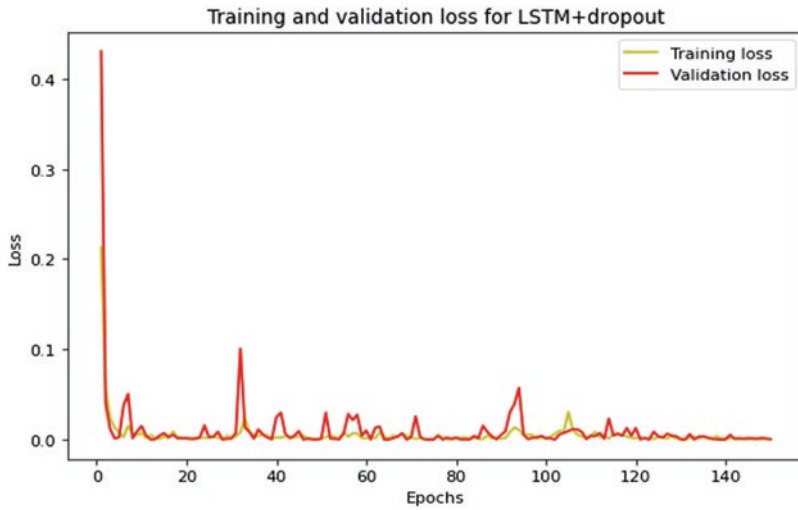


Fig. 12 Trend of active, cured, and death in top three states



**Fig. 13** (a). Plot of training/validation loss for LSTM, (b) training/validation loss for bidirectional LSTM



**Fig. 14** Training/validation loss for LSTM + dropout

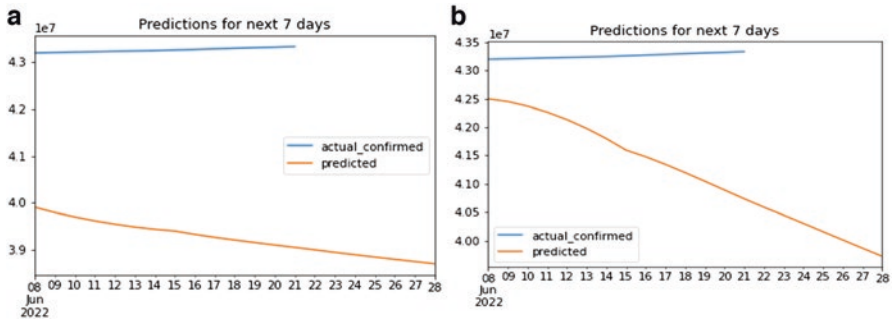
The training-validation loss plot of stacked LSTM + dropout model has shown a similar behavior of the first model, but with the validation loss greater than the first model.

Error measures of the proposed models are calculated using the metrics RMSE and MAPE and are tabulated in Table 2.

As the LSTM and bidirectional LSTM models have shown better, these two models are used for forecasting. These models forecast the confirm cases for 7 days from June 21, 2022, to June 28, 2022, and are shown in Fig. 15a, b, respectively.

**Table 2** Performance measure

Model	RMSE	MAPE
LSTM	0.1504	0.1077
LSTM+ dropout	95.20	0.1568
Bidirectional LSTM	131.24	0.2924



**Fig. 15** Forecasting using (a) LSTM model, (b) LSTM + dropout model

## Conclusion

This study proposed three LSTM variant models to forecast confirmed cases of COVID-19 in India. The data was collected through the government of India website and Johns Hopkins University. The necessary preprocessing techniques on data were carried out and was normalized. The data was split into training and testing dataset. The first model is LSTM model with input layer, two hidden layers, a dense layer, and an output layer. In the second model, dropout layer was added to the first model. The third model is bidirectional LSTM model. The performance of the proposed models has been evaluated using MAPE, and RMSE, on test dataset. The findings revealed that the proposed stacked LSTM outperforms other models and is best suited for Indian covid dataset.

## References

1. R. Magar, P. Yadav, A. Barati Farimani, Potential neutralizing antibodies discovered for novel corona virus using machine learning. *Sci. Rep.* **11**, 5261 (2021). <https://doi.org/10.1038/s41598-021-84637-4>
2. N. Zhu, D. Zhang, W. Wang, A novel coronavirus from patients with pneumonia in china, 2019, 9.22.21, *NEJM*, 2020 [WWW Document]. URL <https://www.nejm.org/doi/full/10.1056/nejmoa2001017>

3. M. Toğaçar, B. Ergen, Z. Cömert, COVID-19 detection using deep learning models to exploit social mimic optimization and structured chest X-ray images using fuzzy color and stacking approaches. *Comput. Biol. Med.* **121**, 103805 (2020). <https://doi.org/10.1016/j.combiomed.2020.103805>
4. B. Mullick, R. Magar, A. Jhunjhunwala, A. Barati Farimani, Understanding mutation hotspots for the SARS-CoV-2 spike protein using Shannon entropy and K-means clustering. *Comput. Biol. Med.* **138**, 104915 (2021). <https://doi.org/10.1016/j.combiomed.2021.104915>
5. K. Arun Kumar et al., Forecasting the dynamics of cumulative covid-19 cases (confirmed, recovered and deaths) for top-16 countries using statistical machine learning models: Auto-regressive integrated moving average (arima) and seasonal auto-regressive integrated moving average (sarima). *Appl. Soft Comput.* **103**, 107161 (2021)
6. T.T. da Silva, R. Francisquini, M.C.V. Nascimento, Meteorological and human mobility data on predicting COVID-19 cases by a novel hybrid decomposition method with anomaly detection analysis: a case study in the capitals of Brazil. *Expert Syst. Appl.* **182**, 115190 (2021). <https://doi.org/10.1016/j.eswa.2021.115190>
7. A. Swaraj, K. Verma, A. Kaur, G. Singh, A. Kumar, L. Melo de Sales, Implementation of stacking based ARIMA model for prediction of Covid-19 cases in India. *J. Biomed. Inf.* **121**, 103887 (2021). <https://doi.org/10.1016/j.jbi.2021.103887>
8. P. Wadhwa, Aishwarya, A. Tripathi, P. Singh, M. Diwakar, N. Kumar, Predicting the time period of extension of lockdown due to increase in rate of COVID-19 cases in India using machine learning. *Mater. Today Proc.*, International Conference on Newer Trends and Innovation in Mechanical Engineering: Materials Science **37**, 2617–2622 (2021). <https://doi.org/10.1016/j.matpr.2020.08.509>
9. P. Wang, X. Zheng, G. Ai, D. Liu, B. Zhu, 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 Solit. Fractals* **140**, 110214 (2020). <https://doi.org/10.1016/j.chaos.2020.110214>
10. B.I. Nasution, Y. Nugraha, J.I. Kangrawan, A.L. Suherman, Forecasting of covid-19 cases in Jakarta using Poisson autoregression, in *2021 9th International Conference on Information and Communication Technology (ICoICT)*, (IEEE, Piscataway, 2021), pp. 594–599
11. C.-S. Yu et al., A covid-19 pandemic artificial intelligence-based system with deep learning forecasting and automatic statistical data acquisition: development and implementation study. *J. Med. Internet Res.* **23**, e27806 (2021)
12. H. Khaloofi, J. Hussain, Z. Azhar, H.F. Ahmad, Performance evaluation of machine learning approaches for covid-19 forecasting by infectious disease modeling, in *2021 International Conference of Women in Data Science at Taif University (WiDSTaif)*, (2021), pp. 1–6. <https://doi.org/10.1109/WiDSTaif52235.2021.9430192>
13. J.S. Armstrong, *Long-Range Forecasting* (Wiley, New York, etc, 1985)
14. Y. Bengio, Y. LeCun, Scaling learning algorithms towards AI. *Largescale Kernel Mach.* **34**(5), 1–41 (2007)
15. I.H. Witten et al., *Data Mining: Practical Machine Learning Tools and Techniques*, 4th edn. (Morgan Kaufmann, Burlington, 2016) <https://www.amazon.com/exec/obidos/ASIN/0128042915/departmentofcompute>. Accessed on 30 Nov 2018
16. S. Hochreiter, J. Schmidhuber, Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997). <https://doi.org/10.1162/neco.1997.9.8.1735>
17. A. Sherstinsky, Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys D* (2020). <https://doi.org/10.1016/j.physd.2019.132306>
18. K. Singh, S. Shastri, A.S. Bhadwal, P. Kour, et al., Implementation of exponential smoothing for forecasting time series data. *Int. J. Sci. Res. Comput. Sci. Appl. Manage. Stud.* (2019) issn: 2319-1953
19. Z. Zhao, K. Nehil-Puleoa, Y. Zhao, How well can we forecast the COVID-19 pandemic with curve fitting and recurrent neural networks? medRxiv preprint 2020. <https://doi.org/10.1101/2020.05.14.20102541>



20. S. Shastri, A. Sharma, V. Mansotra, A model for forecasting tourists arrival in J & K. India. *Int. J. Comput. Appl.* **129**(15), 32–36 (2015) issn: 0975-8887
21. M. Fakhfakh, B. Bouaziz, F. Gargouri, L. Chaari, ProgNet: Covid-19 prognosis using recurrent and convolutional neural networks. medRxiv preprint 2020. <https://doi.org/10.1101/2020.05.06.20092874>
22. Y. Yu, S. Xi, C. Hu, J. Zhang, A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **31**, 1235–1270 (2019). [https://doi.org/10.1162/neco\\_a\\_01199](https://doi.org/10.1162/neco_a_01199)