Chaos, Solitons and Fractals 139 (2020) 110017

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

Nonlinear Science, and Nonequilibrium and Complex Phenomena

journal homepage: www.elsevier.com/locate/chaos

Prediction and analysis of COVID-19 positive cases using deep learning models: A descriptive case study of India

Parul Arora[a,*], Himanshu Kumar[b], Bijaya Ketan Panigrahi[a]

[a] Department of Electrical Engineering, Indian Institute of Technology, Delhi, New Delhi, India
[b] Infosys, Pune, India

ARTICLE INFO

Article history:
Received 20 May 2020
Accepted 12 June 2020
Available online 17 June 2020

Keywords:
COVID-19
Prediction
Deep learning
RNN
LSTM

ABSTRACT

In this paper, Deep Learning-based models are used for predicting the number of novel coronavirus (COVID-19) positive reported cases for 32 states and union territories of India. Recurrent neural network (RNN) based long-short term memory (LSTM) variants such as Deep LSTM, Convolutional LSTM and Bi-directional LSTM are applied on Indian dataset to predict the number of positive cases. LSTM model with minimum error is chosen for predicting daily and weekly cases. It is observed that the proposed method yields high accuracy for short term prediction with error less than 3% for daily predictions and less than 8% for weekly predictions. Indian states are categorised into different zones based on the spread of positive cases and daily growth rate for easy identification of novel coronavirus hot-spots. Preventive measures to reduce the spread in respective zones are also suggested. A website is created where the state-wise predictions are updated using the proposed model for authorities,researchers and planners. This study can be applied by other countries for predicting COVID-19 cases at the state or national level.

1. Introduction

The world would remember the year 2020 as a catastrophic year for humanity on this planet earth. Pneumonia of unknown aetiology(novel coronavirus) identified in the city of Wuhan, China in December 2019 [1] with its first mortality reported on January 10, 2020, has become a pandemic [2] and quickly gulping the entire world under its net. It is named as COVID-19 (Coronavirus disease 2019) by the World Health Organisation(WHO) [3]. As per John Hopkins University, 4,563,458 confirmed cases of COVID-19 [4] reported as of 16-May-20. India contributes 1.9% with 86,508 cases and has fatality rate of 3.2% with 0.2 deaths per 100k population [5]. All countries are trying to save their people lives by implementing measures like travel restrictions, quarantines, event postponements and cancellations, social distancing, testing, hard and soft lockdowns [6]. More than the lives this virus has taken, the economic and social impact is far more disastrous and especially for developing and underdeveloped countries. It is terrifying to imagine the disaster this COVID-19 may cause in India where world's 18% of the population resides [7] with a population density of 32,303 people per square kilometre in cities like Mumbai [8]. So, this novel coronavirus may spread at a very high pace in India's large population. The Government of India is proposing multiple lockdowns to prevent the spread of this virus. Initially, in lockdown 1.0 (March 25, 2020, to April 14, 2020), the entire nation was under complete lockdown except for essential services and lockdown 2.0 (April 15, 2020, to May 3, 2020) was implemented with relaxation in areas where the virus was contained and lockdown 3.0 (May 4, 2020, to May 17, 2020) with more relaxations in areas where there were fewer number of coronavirus cases. Due to these lockdowns, there has been a decrease in the number of cases from 11.8% to 6.3% on a daily basis [9]. But the government cannot shut the entire nation forever as the economy may fall drastically. So, a practical solution could be to quarantine the very critical zones, so that the people affected by this virus shall remain in that zone only.

As per the World Health Organisation, no vaccine and anti-viral treatments are yet available for this virus [10], and medical organisations are trying hard to find out the vaccine for this novel coronavirus. However, even after fast-tracking the usual vaccine period of 5–10 years, the vaccine may take at least 18–24 months before it is available and may take further more time to produce it enough for the majority of the world [11]. Also, we don't know how long a vaccine would stay effective as the virus mutates. Every effort is made to slow down the spread of the coronavirus and prepare medical response systems to tackle the increase in patient loads and to protect the front line medical staff with adequate supplies of personal protective gears like personal protective equipment (PPE), masks and other essentials. So,if we know beforehand

* Corresponding author.
E-mail address: parularora@ee.iitd.ac.in (P. Arora).

https://doi.org/10.1016/j.chaos.2020.110017
0960-0779/

the number of novel coronavirus cases say for next few days, we can plan our inventory accordingly. There are less number of papers on the prediction of novel coronavirus cases in the literature, and few of them are reviewed below.

Wang et al. [12]have reported Patient Information Based Algorithm (PIBA) for estimating the number of deaths due to this COVID-19 in China. The overall death rate in Hubei and Wuhan was predicted 13% and between 0.75% to 3% in the rest of China. They also reported that the mortality rate would vary according to different climates and temperatures. In [13], a case was presented, which showed that there is a direct relationship between temperature and COVID-19 cases based on the United States spread analysis. It showed that there would be a drastic reduction in the number of cases in India in summer months which actually didn't happen. Ahmar and Val [14] have used ARIMA and Sutte ARIMA for short term forecasting of COVID-19 cases and Spanish stock market. They have reported their predictions with MAPE of 3.6% till April 16, 2020. Ceylan [15] have used ARIMA models for predicting the number of positive cases in Italy, Spain and France. He has reported MAPE in range of 4% to 6%. Fanelli and Piazza in [16], have done forecasting and analysis of COVID-19 in Italy, France and China. Based on their analysis, they have forecasted the number of ventilation units required in Italy. They have divided the population in susceptible, recovered, infected and dead, and based on that they have predicted the number of cases. Reddy and Zhang [17] have used deep learning model (LSTM) for predicting the end date of this epidemic in Canada. Their model accuracy is 93.4% for short-term whereas 92.67% for the long-term.

We are proposing a deep learning-based model for predicting the number of patients, who may get infected with COVID-19. We have predicted the number of novel coronavirus positive cases for one day ahead to one week ahead for various states and union territories of India. We have used recurrent neural networks, and long short-term memory(LSTM) based models for the prediction. Contrarily, we have tested multiple LSTM models on Indian dataset and have found that more deep LSTM models like stacked LSTM, convolutional LSTM and bi-directional LSTM give better accuracy than simple LSTM models. As per authors knowledge, no research paper on the prediction of COVID-19 cases of all Indian states have reported so far, and hopefully, our contribution to this area would be beneficial to the government, planners and researchers.

The organisation of this paper is as: Section 2 explains the background of long-short-term memory (LSTM) and its variants such as stacked /deep LSTM, convolutional LSTM and bi-directional LSTM. In Section 3, descriptive data analysis of state-wise Indian data is presented, and the division of states in various zones is discussed. Section 4 explains the model architecture, error calculations predictions using the proposed model and discussion on measures to control the spread in various zones. Section 5 concludes with a summary of the work performed.

2. Methods

2.1. Background:Recurrent neural networks(RNN)

Deep learning speculates that a deep sequential or hierarchical model is more efficient in classification or regression tasks than shallow models [18]. Recurrent neural networks contain hidden states distributed across time, and this allows them to store a lot of information about the past. They are most commonly used in forecasting applications due to their ability to process variable length sequential data [19]. Recurrent neural networks have major disadvantage that they cannot overcome vanishing gradient or exploding gradient problem and also they can store only short-term memory because they involve hidden layer activation functions of the previous time step only [20].

2.2. Long-short-term memory (LSTM) & its variants

For prediction tasks,LSTMs are considered to be among the most feasible solutions, and they anticipate the future forecasts dependent on various highlighted features present in the dataset. With LSTMs, the data moves through components known as cell states. LSTMs can accurately recollect or overlook things. Information gathered over progressive time frames are portrayed as time series data and to produce forecasts with these data values generally LSTMs are proposed to be a stable methodology. In this sort of design, the model passes the past shrouded state to the subsequent stage of the arrangement. Since RNNs can store only limited amount of information, for long term memory storage long short-term memory cells (LSTM) [20] are used along with RNNs. LSTMs overcome the issues of vanishing gradient and exploding gradient [21], which plagues RNN. LSTM cells are similar to RNN with hidden units replaceable with memory blocks.

Fig. 1 represents the LSTM memory block with input, forget and output gates. This structure prevents the memory cell to preserve information over multiple time steps [22]. The states of the gates can be represented mathematically as given by (1)-(5).

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right) \tag{1}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \tag{2}$$

$$c_t = i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) + f_t c_{t-1} \tag{3}$$

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right) \tag{4}$$

$$h_t = o_t tanh(c_t) \tag{5}$$

In (1)–(5) i,o,f,c represents the input gate, output gate, forget gate, cell and σ is the logistic sigmoid activation function which are of the same size as the hidden vector. W represents the weight matrices where W_{ci} represents the cell input gate matrix. Input gate determines how much information should be passed through based on its significance in the current time step, and it protects the cell from irrelevant inputs. Forget gate decides which information should be deleted that is not relevant from the previous timestamp. The output gate controls the flow of information in the rest of the network. LSTM can turn off a memory block if it is generating irrelevant outputs. In this paper,we have used different variants of LSTM like stacked LSTM,convolutional LSTM and bi-directional LSTM. In these, either structural changes in LSTM are performed or hidden layers are made more deep for improving the performance.

2.2.1. Deep LSTM/Stacked LSTM

Stacked LSTM [23], also known as Deep LSTM is the extension of standard LSTM which we have described above. In stacked LSTM, there are multiple hidden layers with multiple memory cells. Stacking multiple layers increases the depth of the neural networks where each layer possesses some information and passes it on to the next. Top LSTM layer provides sequence data to the preceding layer and so on. Stacked LSTM structure is shown in Fig. 2. For every time-step, it provides individual output rather than providing single output for all time-steps.

2.2.2. Convolutional LSTM (Conv-LSTM)

In convolutional LSTM [24], the input x vector,cell output vector y, hidden state vector h and the gates(i_t, f_t, o_t) are 3D tensors with the last two dimensions as spatial dimensions. The inputs in Conv-LSTM determines the future state of any cell in the grid and also

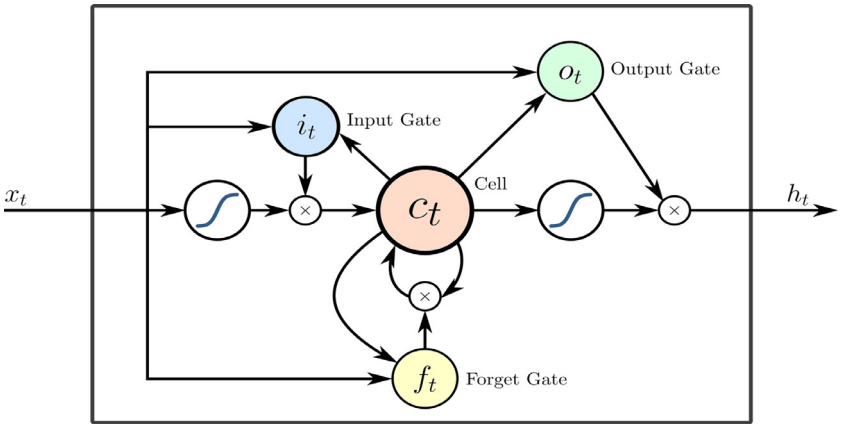

Fig. 1. LSTM Cell.

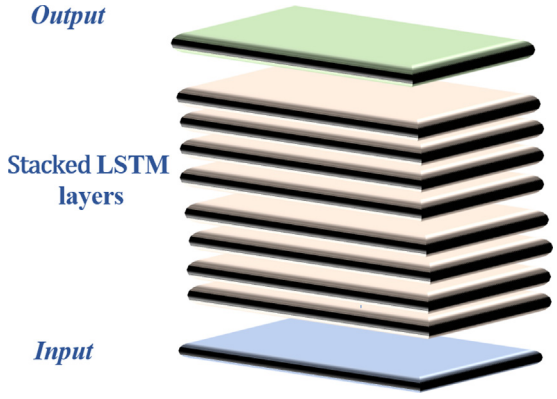

Fig. 2. Stacked LSTM/Deep LSTM

the past states of connecting cells. This is achieved by using convolution operation instead of multiplication function in state transition Eqs. (1)–(5) of LSTM. The key equations of Conv-LSTM are as mentioned in (6)–(10) where $'*'$ represents the convolution operator and $'o'$ represents Hadamard product:

$$i_t = \sigma\left(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} \circ c_{t-1} + b_i\right) \tag{6}$$

$$f_t = \sigma\left(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} \circ c_{t-1} + b_f\right) \tag{7}$$

$$c_t = i_t \circ tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) + f_t \circ c_{t-1} \tag{8}$$

$$o_t = \sigma\left(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} \circ c_t + b_o\right) \tag{9}$$

$$h_t = o_t \circ tanh(c_t) \tag{10}$$

The convolutional network used for predicting COVID-19 cases is shown in Fig. 3. The details of encoding and forecasting network can be found in [24].

2.2.3. Bidirectional LSTM(Bi-LSTM)

Standard RNNs process the inputs in only one direction and ignores the information possessed in future. This issue is overcome by following the bidirectional topology of LSTM. Bidirectional LSTM (Bi-LSTM) [25] extracts the complete temporal information of time t by considering both past and future information. In this, hidden neurons of standard RNN are split into forward and backward states in which neurons of forward states are not connected to the backward states and vice-versa. The unfolded 3 time-step basic structure of Bi-LSTM is shown in Fig. 4. Without the backward states, this structure is similar to the standard unidirectional RNN. With this structure, there is no need to include additional time delays as taken in standard RNN. The training process of Bi-LSTM over time can be summarised as below:

1. ***Forward Pass***
   - Run input data for time $1 \leq t \leq T$ through Bi-LSTM and evaluate the predicted outputs as calculated in standard RNN
   - Run forward pass for forward states from $t = 1$ to $t = T$ and backward states from $t = T$ to $t = 1$
   - Run the forward pass for output neurons
2. ***Backward Pass***
   - Evaluate the objective function derivative for time $1 \leq t \leq T$ calculated in forward pass
   - Run backward pass for output neurons
   - Run backward pass for forward states from $t = T$ to $t = 1$ and backward states from $t = 1$ to $t = T$
3. ***Update Weights***

3. Data analysis

3.1. Data set description

The data-set used in this paper is taken from the Ministry of Health and Family Welfare (Government of India) [9]. This data is highly stochastic in nature as increase/decrease in number of cases depend on other environmental/physical variables. It consists of 32 individual time-series data of confirmed COVID-19 cases in each of the states (28) and union territories (4) since March 14, 2020. The missing values in each of the individual series are imputed with the missing data statistics technique known as linear weighted moving average such that the model maintains the sequential learning ability and is feasible to produce accurate future predictions. We have taken data for study purposes (from March 14, 2020 to May 14, 2020). We have splitted data into training data from March 14,2020 to May 8,2020 and testing data from May 9,2020 to May 14,2020. Fig. 5 shows the 11 states of India in which the number of positive COVID-19 cases reaches above 1000 in 60 days. This graph shows that there are few states like Maharashtra (14.3%), Tamil Nadu (18.3%),Gujarat (19.3%) and Delhi (13.7%) in which the number of cases are increasing at very high rate and states like Rajasthan , Madhya Pradesh in which cases are increasing linearly and in states like Telangana, West Bengal and Punjab,

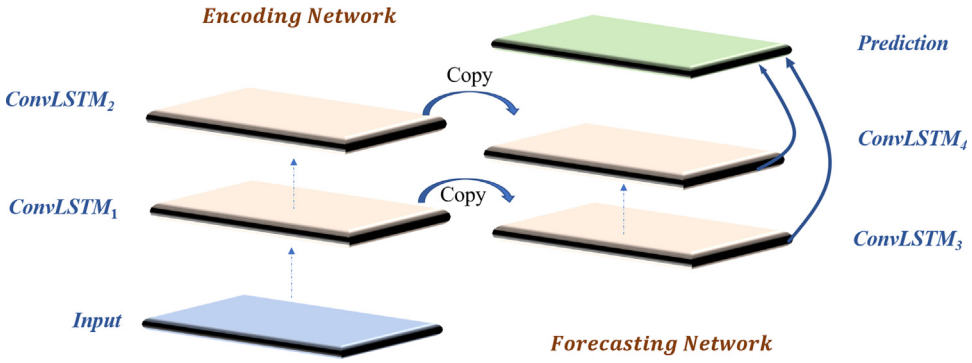

Fig. 3. Convolutional LSTM network for forecasting.

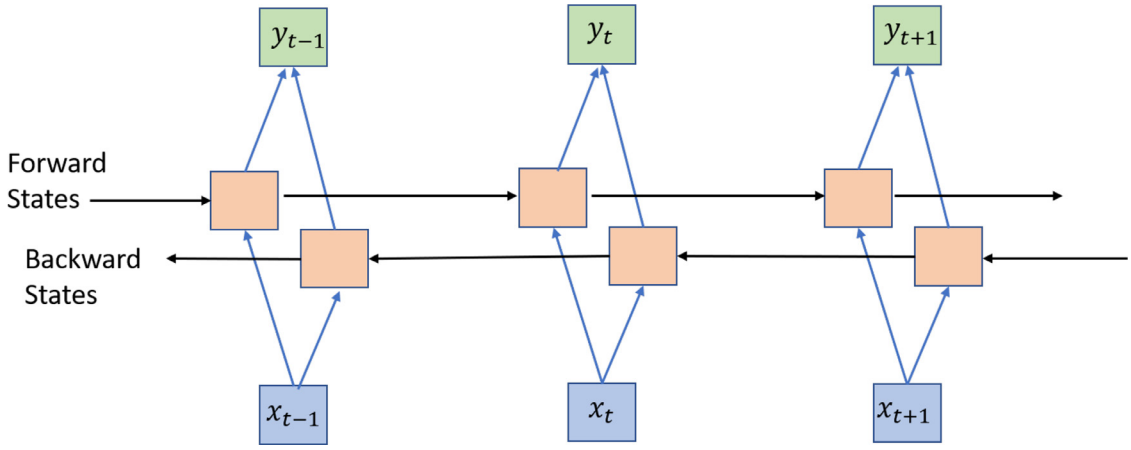

Fig. 4. Bidirectional LSTM.

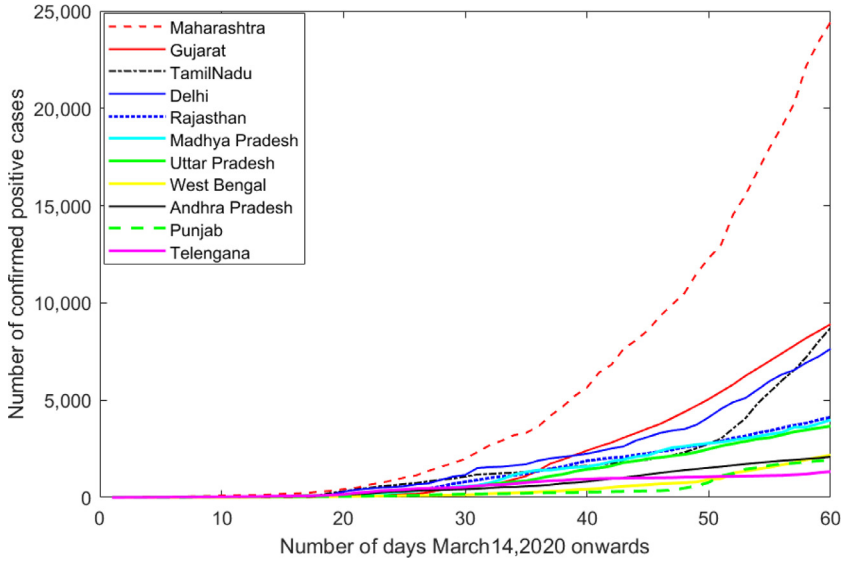

Fig. 5. Indian states with number of COVID-19 positive cases above 1000 from March 14,2020 to May 14,2020.

linear curve with less growth rate is obtained. In other states and union territories (except these 11 states),growth rate is very less, that's why they are not shown in this graph.

3.2. Spread analysis

We have divided India into three different categories based on the positive number of COVID-19 cases and daily rise as shown in India map in Fig. 6.

- Mild Zone- All states where total number of positive COVID-19 cases are below 200 and daily rise is below 2% are kept in the mild zone. It is indicated in green colour in Fig. 6,
- Moderate Zone-States with positive COVID-19 patients between 200 and 2000 and daily increment less than 5% are under moderate zone. It is indicated in yellow colour in Fig. 6.
- Severe Zone-States with positive COVID-19 patients above 2000 and and daily increment greater than 5% are under severe zone. It is indicated in red colour in Fig. 6.

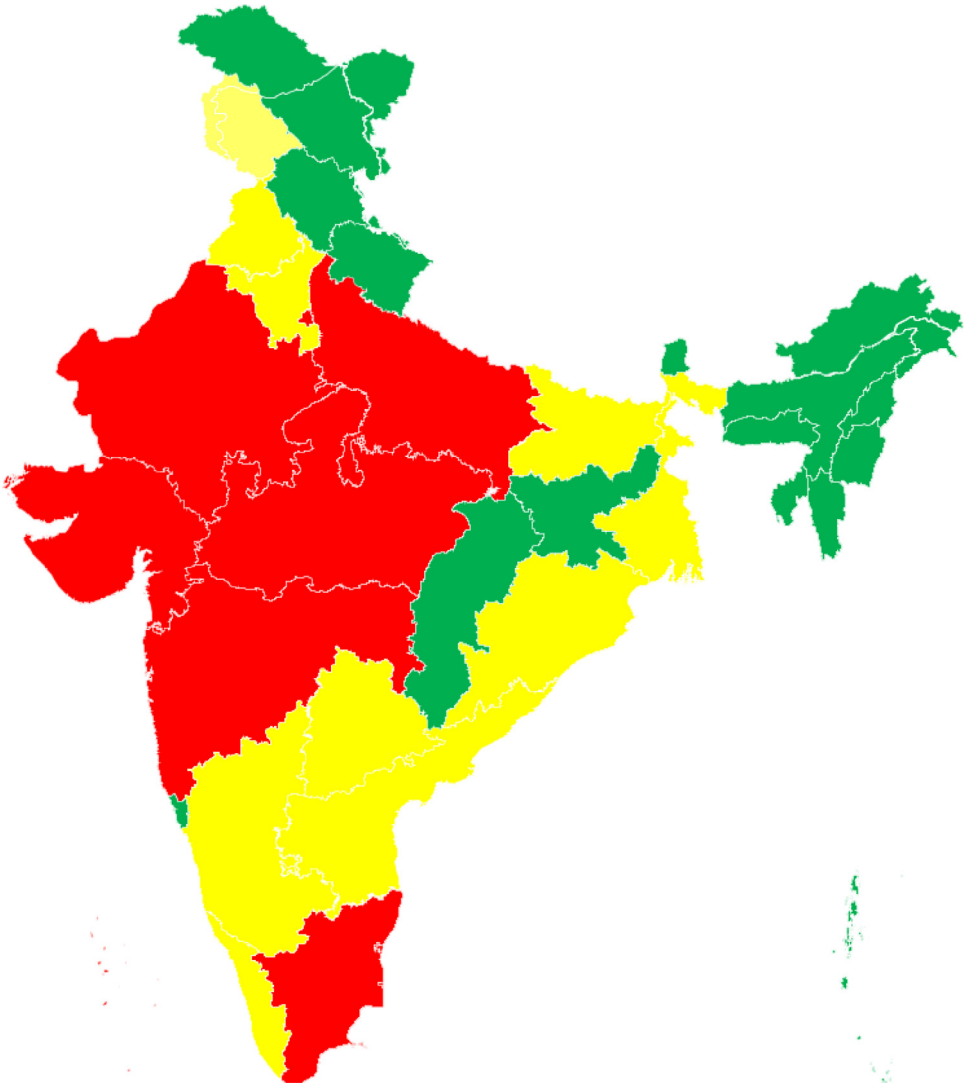

Fig. 6. Division of India in the severe (red), moderate (yellow) and mild (green) zones depending upon the number of confirmed COVID-19 positive cases and daily rise based on the data till May14,2020. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4. Results & discussion

4.1. Model design

The experiments are conducted on open source libraries such as Numpy [26], Pandas [27], Tensorflow(Google) [28]and Keras [29]. Python [30], as a high-level general-purpose programming language, is used to interact with deep learning libraries as application program interfaces(APIs). The obtained APIs is used to design the current model structure for various recurrent neural network variants as Deep LSTMs, Bi-LSTMs and Conv-LSTMs. These models are used to learn the dynamic dependent structure present in the data and also maps the learning sequence present, to produce future forecasts of the number of confirmed cases present in any particular region. These models are provided with region-based historical data of the number of cases appearing daily, and in consideration to the dynamically changing structure of the dataset, we used the historical data ranging from March14, 2020 to May 14,2020 for training and testing our prediction models. Hyper-parameter tuning of each of these models is done rigorously and selection procedure of these parameters is explained in Sections 4.1.1–4.1.3. In these models,adam optimiser is used for optimising the mean squared error loss. The error of these models is evaluated on testing data-set in Section 4.2 and based on the errors, the best model with maximum accuracy is selected as the prediction model for COVID-19 data. Output values are then rounded off to the nearest integer value as the number of confirmed cases cannot be presented in the decimal number system. The layout of our proposed methodology is shown in Fig. 7.

4.1.1. Stacked LSTM/Deep LSTM model design

Stacked LSTM can be characterised as LSTM model that involves multiple LSTM layers as explained in section 2.2.1. Selecting how deeper the model should be is another aspect of hyperparameter optimisation which can generally go from a single layer to three to four-layer deep model architecture where a three-layer architecture is used mostly in complex learning tasks. The model used in this experiment uses a two-layer deep LSTM setup with each of the layer having 100 hidden neurons units. The input shape in the model is found to be the lag structure with number of steps as 3 and number of features to be 1. Also, the model uses the ReLu activation function to overcome the most commonly existing problem in recurrent neural networks as vanishing gradient problem.

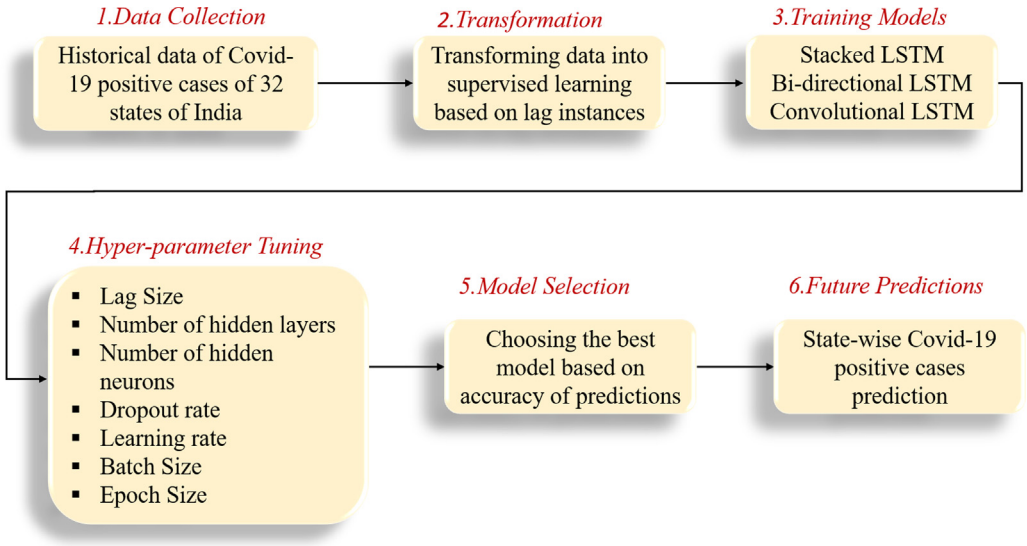

Fig. 7. Layout of proposed method.

4.1.2. Convolutional LSTM (Conv-LSTM) model design

Conv-LSTMs considered to be a LSTM variant which replaces the matrix multiplication with convolutional operation at each of the present cell gates inside the LSTM cell and it helps the model to map long term sequential dependencies present in the data as explained in section 2.2.2 Input and recurrent transformations inside each cell are convolutional. The Conv-LSTM model used in this experiment used a single convLSTM2D layer stacked along with a densely connected output layer. Input shape fed into the model is the lag structure along with the number of subsequences and input number of features are found to be 4,2 and 1, respectively. The data input is restructured to the above input shape along with the number of filters(64) and kernel size. The convolution window (1,2) is an integer value that is used in generating the feature map. The trained model is then used to generate day ahead forecasts for each of the states/UTs using the test data with three time lags.

4.1.3. Bi-directional(Bi-LSTM) LSTM model design

Bi-LSTM is different from unidirectional in the sense that the LSTM that runs in reverse protects the data from the future and utilising the two concealed states jointly, they are capable at any point to safeguard data from both past and future. Bi-LSTMs comprehend the settings better as they look into both past and future directions as discussed in section 2.2.3 The Bi-LSTM model used in this experiment uses a single hidden layer with 100 hidden neuron units wrapped inside a bidirectional wrapper class structure. This includes copying the principal intermittent layer in the system so that there are presently two layers, one next to the other. The selected mode of merging both the layers is selected to be 'concat' out of multiple options in hand as 'sum' or 'mul' value. The input shape fed into the model is selected to be the number of steps also called lag structure behind in space. The bidirectional wrapper classes are finally passed through a dense layer structure to produce prediction values. The ReLu function helped to eliminate gradient issues, and the model is trained on the mean squared error as a loss parameter and trained through a fixed number of epochs.

4.2. Error comparison

The performance of our proposed prediction methods is compared in terms of certain performance measure indices like mean absolute percentage error (MAPE) and is described as follows

$$MAPE = \left(\frac{1}{k} \sum_{x=1}^{k} \left| \frac{A_x - F_x}{A_x} \right| \right) * 100 \tag{11}$$

MAPE quantifies exactness as a rate, and can be determined as total absolute percent error for each time frame rate as actual values minus predicted values divided over actual values. In Table 1,we have evaluated the MAPE of the 32 Indian states by convolutional LSTM,stacked LSTM and bi-directional LSTM models. Stacked LSTM has average MAPE of 4.81%, bi-directional LSTM has 3.22% and conv-LSTM has 5.05%. MAPE of 0% (ideal) in few states indicates that our model exactly predicted the actual number of cases. MAPE ranges upto 30.67% in stacked LSTM, upto 21.6% in convolutional LSTM and upto 15.35% in Bi-directional LSTM. These errors are calculated on 15 days testing dataset,for selecting the best model. The model with high range in errors will fluctuate more with small change and prone to high deviations in predicted values. Moreover,Bi-LSTM has limited error range with minimum average error among all models and is more suitable for prediction purposes rather than convolutional and stacked LSTM. Therefore,we have used Bi-LSTM model for evaluating predictions of COVID-19 positive cases.

4.3. Prediction

The number of COVID-19 positive cases predicted by Bi-LSTM follows the actual number of cases closely as tested on the dataset of India for 15 days (April 30, 2020, to May 14, 2020). In Fig. 8, error percentage is highlighted in red, for example for May 14,2020 this model has error of 0.279%. This model is also tested state-wise for daily and weekly predictions for May 9, to May 15, data. Using Bi-LSTM model, i.e. using data till May 8, 2020,we have calculated 1-week prediction from May 9,2020 to May 15,2020 and next day prediction is evaluated considering last day actual values. In Table 2,daily and weekly prediction errors are calculated for four states of India (Maharashtra, Tamil Nadu, Delhi and Rajasthan). The predictions of other states are also available at our website [31]developed for novel coronavirus predictions. From this table, it is evident that daily MAPE is within 3% and weekly prediction MAPE is in range of 4% to 8%. For weekly predictions,we can see that the error increases

Table 1
Mean Absolute Percentage Error (MAPE) of states and union territories(UTs) of India by convolutional, stacked and bi-directional LSTM models.

S.No.	States/UTs	Convolutional LSTM	Stacked LSTM	Bi-directionalLSTM
1	Andaman and Nicobar	0	0.2	0
2	Andhra Pradesh	3.2	1.6	1.24
3	Arunachal Pradesh	0	0	0
4	Assam	7.28	6.3	5.49
5	Bihar	7.03	4.95	5.3
6	Chandigarh	8.76	8.3	6.64
7	Chhattisgarh	12.94	11.05	10.9
8	Delhi	2.86	3.4	2.13
9	Goa	0	0	0
10	Gujarat	2.78	2.02	0.99
11	Haryana	5.94	5.23	4.35
12	Himachal Pradesh	5.57	3.81	2.68
13	Jammu and Kashmir	2.36	1.82	1.53
14	Jharkhand	5.46	3.53	2.95
15	Karnataka	3.06	2.31	1.71
16	Kerala	2.04	0.74	0.63
17	Ladakh	12.23	11.19	7.63
18	Madhya Pradesh	4.38	4.44	1.9
19	Maharashtra	2.43	2.23	1.29
20	Manipur	0	0	0
21	Meghalaya	1.1	0.55	0.55
22	Mizoram	0	0	0
23	Odisha	7.79	6.4	5.88
24	Puducherry	3.13	12.65	3.13
25	Punjab	18.02	12.07	7.95
26	Rajasthan	1.3	2.35	1.35
27	Tamil Nadu	7.17	5.33	3.53
28	Telangana	1.83	1.39	0.97
29	Tripura	21.16	30.67	15.35
30	Uttar Pradesh	3.37	2.32	1.11
31	Uttarakhand	2.03	2.26	1.8
32	West Bengal	6.25	4.95	4.16
Average		**5.05**	**4.81**	**3.22**

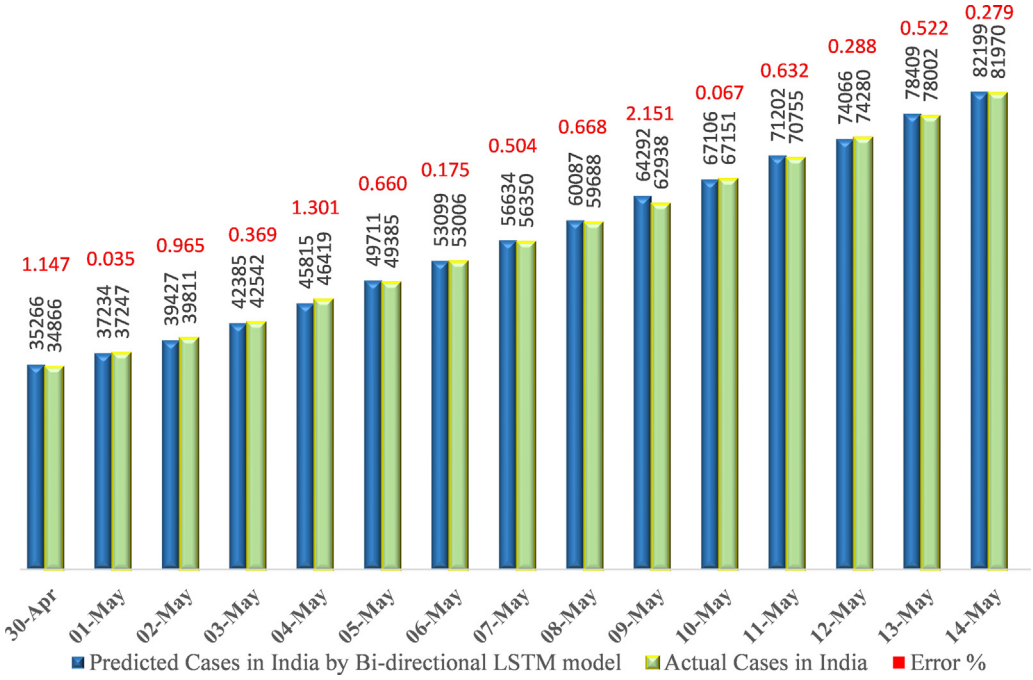

Fig. 8. 15 days comparison of predicted and actual Covid-19 positive cases by bi-directional LSTM model for India for the year 2020.

high more from fourth day onwards. Thus, we can say that this model is highly accurate for short-term predictions (1–3 days) ahead.

The highly accurate short-term, state-wise predictions will help the state-authorities to balance the load which medical infrastructure can take. Depending upon predictions other several decisions can be taken like imposition or removal of lockdowns and this would also make sure the economic activities can resume which otherwise may create livelihood challenge for millions of people.

Table 2
Daily and weekly error percentages for one-week testing data using Bi-directional LSTM model.

	Maharastra		Tamil Nadu		Delhi		Rajasthan	
	Daily Error %	**Weekly Error%**	**Daily Error %**	**Weekly Error %**	**Daily Error %**	**Weekly Error %**	**Daily Error %**	**Weekly Error %**
09-May	2.70	2.20	5.31	5.00	2.74	1.57	0.65	0.30
10-May	1.35	0.41	1.01	7.61	0.58	3.18	6.82	4.30
11-May	0.96	2.17	0.30	12.30	1.11	5.43	0.15	3.64
12-May	0.87	6.45	0.78	12.50	0.14	6.28	0.73	5.28
13-May	2.63	8.96	6.55	9.80	3.63	7.58	0.85	6.56
14-May	1.18	10.51	1.20	5.93	0.61	8.58	2.14	5.82
15-May	0.00	12.66	1.88	0.89	0.15	10.55	3.12	6.00
MAPE	**1.39**	**6.20**	**2.43**	**7.72**	**1.28**	**6.17**	**2.06**	**4.56**

4.4. Discussion

The output of our proposed model can help planners and authorities to decide on lockdown measures. The state-wise predictions will help the state-authorities to balance the load which medical infrastructure can take, and this would also make sure that the economic activities can resume which otherwise may create livelihood challenge for millions of people . We have divided the zones in mild, severe and moderate category and have explained the division process in Section 3.2. For each zone different preventive measures should be taken either to prevent or to contain the increase in the number of novel coronavirus patients. Zone wise preventive measures are listed below:

- Mild Zone
  - As all of these states are under quarantine zones, but still, they can start economic activities. However, a close watch is required to make sure that the situation does not go out of control, and COVID-19 tests should be mandatory on all the incoming travellers from other states.
- Moderate Zone
  - This is a zone, which can go in towards mild or severe zone in coming days depending upon the preventive measures it will take. Identification, along with sealing of containment zones, is the key to control virus spread. Soft lockdown in non-contained zones with partial economic activity can sustain in these areas. Extensive testing is required to check unknown and asymptomatic cases to prevent further spread. All efforts should be on moving towards the mild zone.
- Severe Zone
  - This is a zone where things have gone out of control and already put down enormous pressure on healthcare workers and facilities. Hard lockdown with no economic activity and identification along with sealing of containment zones is the key to control virus spread. Multiple rounds of testing are required before declaring any zone virus free. All efforts should be on moving towards the moderate zone and then towards the mild zone.

5. Conclusion

In this paper, we have proposed deep learning models for predicting the number of COVID-19 positive cases in Indian states. An exploratory data analysis of the increase in the number of positive cases in India has been done. Based on the number of cases and daily growth rate, states are categorised into mild, moderate and severe zones for realistic lockdown measures state-wise in comparison to locking down the whole nation, which may cause socio-economic issues. Recurrent neural network (RNN) based long short term memory (LSTM) cells are used as prediction models. LSTM variants such as deep LSTM, convolutional LSTM and bi-directional LSTM models are tested on 32 states/union territories and based on absolute error, the model with maximum accuracy is chosen. Based on prediction errors, bi-directional LSTM gives the best results, and convolutional LSTM gives the worst. Daily and weekly predictions are calculated for all states, and it is found that bi-LSTM gives very accurate results(error less than 3%) for short-term prediction (1–3 days). Predictions are publicly available at a website developed for the general public. These predictions will be helpful for state and national government authorities, researchers and planners for managing services and arranging medical infrastructure accordingly. The proposed model and preventive strategy can be followed by other nations as well.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Huang C, Wang Y, Li X, Ren L, Zhao J, Hu Y, et al. Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. Lancet 2020;395(10223):497–506.
[2] Sohrabi C, Alsafi Z, OfiNeill N, Khan M, Kerwan A, Al-Jabir A, et al. World health organization declares global emergency: a review of the 2019 novel coronavirus (COVID-19). Int J Surg 2020.
[3] Organization W.H., et al. Naming the coronavirus disease (COVID-19) and the virus that causes it. 2020a.
[4] Johns hopkins coronavirus resource center. 2020. https://coronavirus.jhu.edu/ (Accessed on 05/16/2020).
[5] Mortality analyses - johns hopkins coronavirus resource center. 2020. https://coronavirus.jhu.edu/data/mortality (Accessed on 05/16/2020).
[6] Acter T, Uddin N, Das J, Akhter A, Choudhury TR, Kim S. Evolution of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) as coronavirus disease 2019 (COVID-19) pandemic: aglobal health emergency. Sci Total Environ 2020:138996.
[7] India population (2020) - worldometer. 2020. https://www.worldometers.info/world-population (Accessed on 05/16/2020).
[8] List of cities proper by population density - wikipedia. 2020. https://en.wikipedia.org/wiki/List of cities proper by population density (Accessed on 05/17/2020).
[9] Mohfw | home. 2020. https://www.mohfw.gov.in/ (Accessed on 05/16/2020).
[10] Organization W.H., et al. Q&a on coronaviruses. 2020b.
[11] Grenfell R., Drew T.. Here's why it's taking so long to develop a vaccine for the new coronavirus. Science Alert Archived from the original on 28 2020.
[12] Wang L, Li J, Guo S, Xie N, Yao L, Cao Y, et al. Real-time estimation and prediction of mortality caused by COVID-19 with patient information based algorithm. Sci Total Environ 2020:138394.
[13] Gupta S, Raghuwanshi GS, Chanda A. Effect of weather on COVID-19 spread in the us: aprediction model for India in 2020. Sci Total Environ 2020:138860.
[14] Ahmar AS, del Val EB. SutteARIMA: short-term forecasting method, a case: COVID-19 and stock market in spain. Sci Total Environ 2020:138883.
[15] Ceylan Z. Estimation of COVID-19 prevalence in Italy, Spain, and France. Sci Total Environ 2020:138817.
[16] Fanelli D, Piazza F. Analysis and forecast of COVID-19 spreading in China, Italy and France. Chaos Solitons Fractals 2020;134:109761.
[17] Chimmula VKR, Zhang L. Time series forecasting of COVID-19 transmission in canada using LSTM networks. Chaos Solitons Fractals 2020:109864.
[18] Bengio Y. Learning deep architectures for ai. Found Trends Mach Learn 2009;2(1):1–127.
[19] Graves A.. Generating sequences with recurrent neural networks. arXiv preprintarXiv:130808502013;.

[20] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.
[21] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 1994;5(2):157–66.
[22] Abbasimehr H, Shabani M, Yousefi M. An optimized model using LSTM network for demand forecasting. Comput Ind Eng 2020:106435.
[23] Graves A, Mohamed A-r, Hinton G. Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing. IEEE; 2013. p. 6645–9.
[24] Xingjian S, Chen Z, Wang H, Yeung D-Y, Wong W-K, Woo W-c. Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems; 2015. p. 802–10.
[25] Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Trans Signal Process 1997;45(11):2673–81.
[26] Oliphant TE. A guide to NumPy, vol. 1. Trelgol Publishing USA; 2006.
[27] McKinney W, et al. Data structures for statistical computing in python. In: Proceedings of the 9th Python in science conference, vol. 445. Austin, TX; 2010. p. 51–6.
[28] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. TensorFlow: a system for large-scale machine learning. In: 12th {USENIX} Symposium on operating systems design and implementation ({OSDI} 16); 2016. p. 265–83.
[29] Chollet F., et al. Keras. https://github.com/fchollet/keras; 2015.
[30] Van Rossum G, Drake FL. Python 3 Reference Manual. Scotts Valley, CA: CreateSpace; 2009. ISBN 1441412697
[31] Arora P, Kumar H, Panigrahi B.K, Gupta P, Gupta P. Corona predictor. 2020. https://covid-19predictor.herokuapp.com/ (Accessed on 05/17/2020).