Recurrent Neural Network and Auto-Regressive Recurrent Neural Network for trend prediction of COVID-19 in India

Samya BOUHADDOUR[1*], *Chaimae* SAADI[2], *Ibrahim* BOUABDALLAOUI[3], *Fatima* GUEROUATE[4] and *Mohammed* SBIHI[5]

Laboratory of Analysis Systems, Processing Information and Industrial Management, EST Sale, Mohammed V University in Rabat, Morocco

Abstract. On 31st December 2019 in Wuhan China, the first case of Covid-19 was reported in Wuhan, Hubei province in China. Soon world health organization has declared contagious coronavirus disease (COVID-19) as a global pandemic in the month of March 2020. Since then, researchers have focused on using machine learning and deep learning techniques to predict future cases of Covid-19. Despite all the research we still face the problem of not having a good and accurate prediction, and this is due to the complex and non-linear data of Covid-19. In this study, we will implement RNN and Auto Regressive RNN. At first, we implement LSTM and GRU in an independent way, then we will implement deepAR with LSTM and GRU cells. For the evaluation of the obtained results, we will use the MAPE and RMSE metrics.

Abbreviations. ***ARIMA***, Autoregressive Integrated Moving Average; ***Bi-GRU***, Bidirectional Get Recurrent Unit; ***Bi-LSTM***, Bidirectional Long-Short Term Memory; ***Bi-Conv-LSTM***, Bidirectional Convolutional Long-Short Term Memory; ***CNN,*** Convolutional neural network ; ***Conv-LSTM***, Convolutional Long-Short Term Memory; ***Covid-19***, Coronavirus Disease 2019; ***DeepAR***, Deep Auto 'Regression; ***ED_LSTM***, Encoder Decoder-LSTM; ***ES***, Exponential Smoothing; ***EV***, Explained Variance; ***GRU***, Get Recurrent Unit; ***LR***, linear regression ; ***LSTM***, Long Short-Term Memory; ***MAE***, Mean Absolute Error; ***MAPE***, Mean Absolute Percentage Error; ***MSLE***, Mean Squared Log Error; ***NB***, Naive Bayes; ***RMSE***, Root Mean Square Error; ***RMSLE***, Root Mean Squared Log Error; ***RNN***, recurrent neural network ;***R2_score,*** R-Squared (Coefficient of Determination) Regression Score; ***SARIMAX,*** Seasonal Autoregressive Integrated Moving Average ; ***SVM***, Support Vector Machine; ***SVR***,Support Vector Regression; ***WHO***, World Health Organization;

1 Introduction

Coronavirus (COVID-19) is a respiratory disease of severe acute respiratory syndrome. The first identified was in December 2019 when a group of patients presented a new form of viral pneumonia in Wuhan, China. On March 11, 2020, the WHO, declared the new coronavirus (2019-nCoV) outbreak as a global pandemic. Since then, several preventive measures such as containment, rapid testing, wearing masks, self-quarantine, social distancing are applied by countries to stop the spread of COVID-19 pandemic. The most affected countries by COVID-19 are: USA, India, Brazil,France,Germany,UK,Russia,S.Korea,Italy,Turkey,Spain,Vietnam[1]. The first case of Covid-19 in India was confirmed on 30 January 2020 in Kerala. A few months later, the number of confirmed cases increased daily. 97847 was the maximum number of confirmed cases in 2020, specifically in September. From 5 April, the number of cases began to reach 400,000 cases[2].

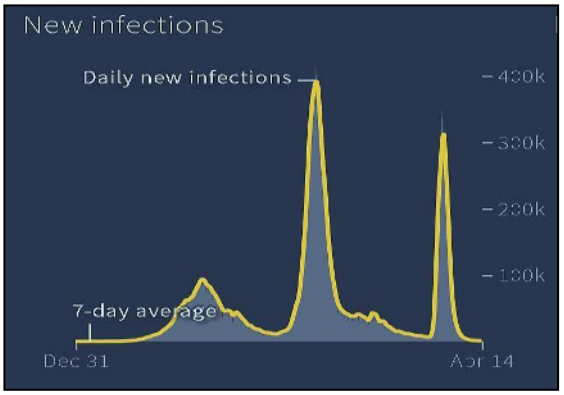

Fig. 1. Daily reported trends in India [3]

During this pandemic, machine learning or deep learning techniques have gained immense interest from researchers[4]. Most of the researches have been focused on the prediction and forecasting the future Covid-19 cases on short-term by implementing the machine learning and deep learning techniques. These researches and studies are aimed to take the proper

* Corresponding author: samyabouhaddour@research.emi.ac.ma

decisions in the future, whether it is for travel restrictions or confinement or any other kind of decision. In parallel, these researches have studied the impact of this pandemic on different sectors such as health, tourism....

Shahid et al. [5] have used a COVID-19 dataset and has been modelled it using various regressors including ARIMA, LSTM, GRU, and Bi-LSTM for future predictions on confirmed cases, deaths, and recovered cases for ten countries around the world (Brazil, China, Germany, India, Israel, Italy, Russia, Spain, UK, USA). The metrics used to assess performance are: MAE, RMSE and r2_score. The results showed that the ARIMA and SVR models are unable to track the trend of the features with higher prediction error and negative r2_score values. LSTM, GRU and Bi-LSTM proved to be robust with higher accuracy rate.

Verma et al. [6] have led to a comparative study of the RNN /CNN recurrent and convolutional recurrent neural network models: vanilla LSTM, stacked LSTM, ED_LSTM, Bi-LSTM, CNN and a hybrid CNN+LSTM model to capture the complex trend of the COVID-19 epidemic. This study was done to predict future cases of covid-19 in India and the United States. In order to evaluate these models, they used the mean square error RMSE and the mean absolute percentage error MAPE as metrics. The results showed the robustness of the LSTM model and the hybrid model CNN+LSTM+ compared to the other models.

Bhangu et al. [6] have focused on the monthly analysis of time series of confirmed, cured and deceased COVID-19 cases, which allows to identify the trend and seasonality of the data. They used the ARIMA (auto-regressive integrated moving average) and SARIMAX (seasonal auto-regressive integrated moving averages with exogenous regressor) models which was optimized to have good results.

Tiwari et al. [7] have implemented machine learning techniques, namely Naïve-Bayes, SVM and linear regression to predict the future growth and effects of the epidemic. Their demonstration showed that Naïve Bayes gives better results and better predicts confirmed Covid-19 cases with minimum MAE and MSE value. The results predicted by Naïve Bayes are almost similar to the actual confirmed Coronavirus cases.

Rustam et al. [8] have used four standard prediction models, such as linear regression (LR), least absolute shrinkage and selection operator (LASSO), support vector machine (SVM), and exponential smoothing (ES) to predict COVID-19 threat factors such as the number of new infections, number of deaths, and number of recoveries in the next 10 days. The results proved that ES performed the best of all the models used followed by LR and LASSO, while SVM performed poorly.

Ayoubi et al. [10] have used deep learning methods: LSTM and Gru and their bidirectional extensions Bi-LSTM, Bi-Conv-LSTM, Bi-Conv-LSTM and Bi-GRU to make a prediction on future Covid-19 cases. The results showed that the bidirectional models have a lower error rate with the following metrics: EV, MAPE, MSLE, RMSLE.

Table 1. Models reported in Literature for forecasting COVID-19 pandemic

	Models and their parameters cited in Literature										
Authors	**ARIMA**			**SARIMA**			**GRU**	**LSTM**	**BI-LSTM**	**Other models**	**Remarks**
	p	**d**	**q**	**(p,P)**	**(d,D)**	**(q,Q)**	**(NL) Number of Layers**	**(Nl) Number of Layers**	**(NL) Number of Layers**		
Shahid et al. [5]	**1**	**1**	**1**	***NA***	***NA***	***NA***	**3**	**3**	**3**	**SVR**	They concluded that Bi-LSTM as the best model for forecasting with enhanced accuracy.
Verma et al. [9]	***NA***	***NA***	***NA***	***NA***	***NA***	***NA***	***NA***	***NA***	***250***	**vanilla LSTM, stacked LSTM, ED-LSTM, CNN+LSTM model hybrid**	This study reported that de hybrid model as the best model for forecasting the next 21 days cases in India.
Bhangu et al. [6]	***2***	***1***	***0***	***(1,2)***	***(1,1)***	***(2,2)***	***NA***	***NA***	***NA***	***NA***	The chosen model is able to capture linear features but is not able to capture the complex and non-linear data of Covid 19

Tiwari et al. [7]	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	**Naïve Bayes, SVM, Linear Regression**	Study did not include the deep learning models.
Rustam et al. [8]	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	**LR, LASSO,SVM , ES**	Study did not include the deep learning models.
Ayoubi et al.[10]	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*3*	*3*	*3*	**Bi-GRU,Conv-LSTM, Bi-Conv-LSTM**	The results show that the bidirectional models have lower errors than other models.

The structure of this paper is: Section two describes the deep learning models that we used and the evaluation metrics. In the third section, we present the Covid-19 dataset, the experimental results, and the discussion. Finally, conclusions are in the fourth section.

2 Methodology and data

2.1 Experimental setup

The trend of COVID-19 epidemic is very dynamic and complex to be captured. To capture this complex trend, we perform the following steps during training, testing and forecasting.

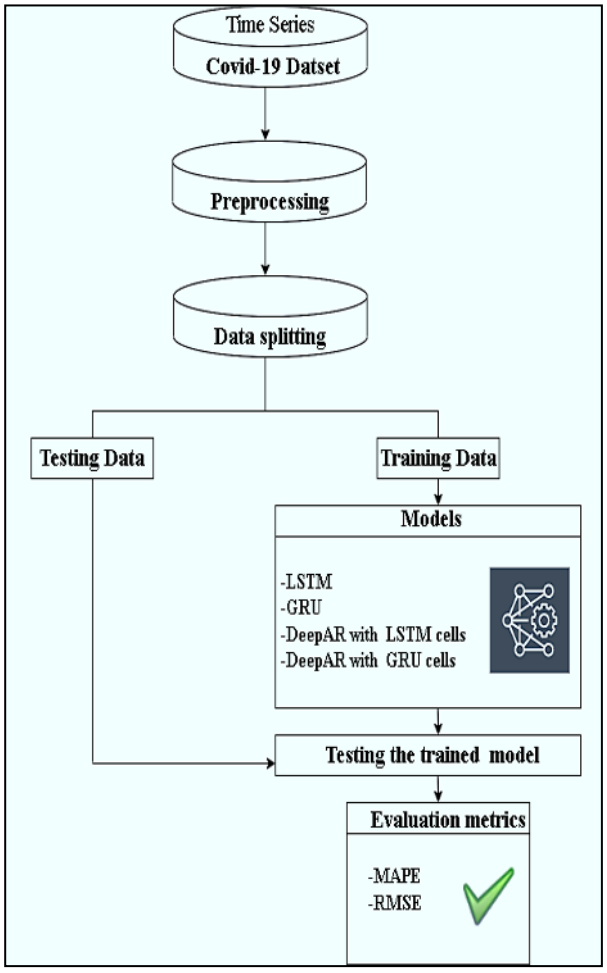

Fig. 2. Suggested Workflow

Table 2. Selection of data for experimentation.

Country	Confirmed cases
India	08 May 2020 - 07 March 2021

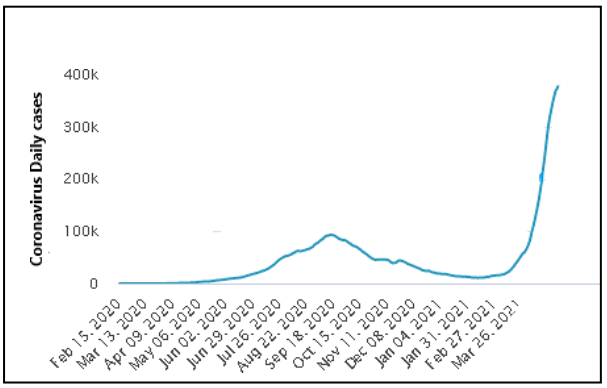

Fig. 3. Confirmed cases in India

The general steps of this study are as follows:
-We used India's Covid-19 dataset of confirmed cases from May 08, 2020 to March 07, 2021.Covid-19 time series data is accessed from https://data.covid19india.org/. The graphical representation of confirmed cases shows an increasing trend as shown in Figures 3.

-The dataset is a time series that we normalized in the interval [0,1].
-For the training phase we used 80% of the dataset and the rest for testing.

-The models have been implemented using Python. In order to evaluate these models, we used the following metrics: RMSE and MAPE.

2.2 Mathematical modelling

Deep learning methods are used to make predictions on data that can be linear or non-linear or both. When we want to make predictions with time series, we often use recurrent neural networks RNN which are known for their robustness in predictions. What makes RNNs robust is the way the information flows between the cells. The input from the current time step and the output from the previous time step will be introduced into the RNN cells so that the current state of the model is impacted by its previous states. RNN models cannot remember past information that is faraway in time.

LSTM & GRU

GRU is a simplified version of the LSTM cell, it requires the less training time with improved network performance. In terms of operations, LSTM has two different states transmitted between the cells: the cell state and the hidden state which carry the long-term and short-term memory respectively. On the contrary, GRU have only one hidden state transferred between time steps. LSTM has a cell state that stores and converts the input cell memory to the output cell state. The architecture of LSTM mainly consists of an input gate, an output gate, a forget gate and an update gate. The forget gate determines what to forget from the received information, the input gate determines what information to accept into the neuron, the output gate generates the new long-term memory and the update gate updates the cell. For GRU we have 2 gates: the update gate and the reset gate, these two gates are trained to filter out any irrelevant information. reset gate, these two gates are trained to filter out any irrelevant information.

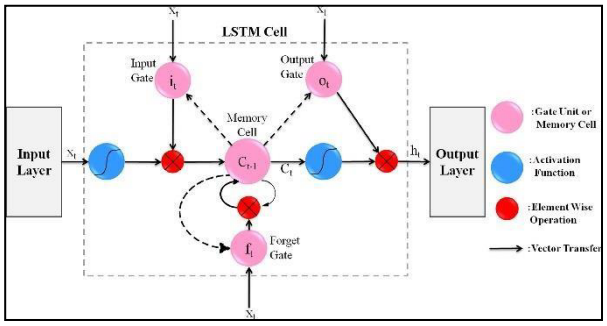

Fig.5. Internal architecture of LSTM cell[11]

-In the first stage, we tested the deep learning models LSTM, GRU, DeepAR with LSTM cells and DeepAR with GRU cells. The experimental work is shown in Fig.2.

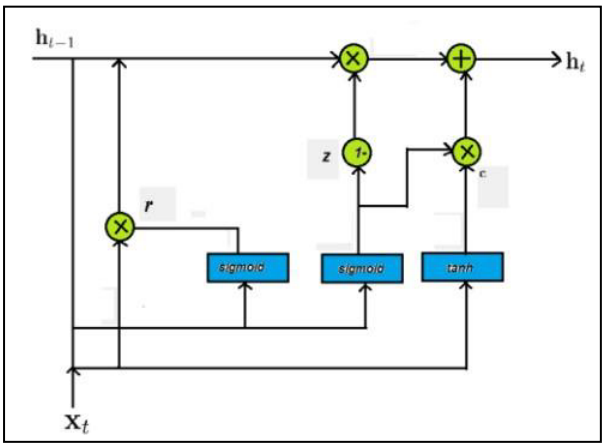

Fig.6. Internal architecture of GRU cell[12]

DeepAR is a forecasting method based on autoregressive RNN, which learns a global model from the historical data of all series in the data set. DeepAR, a methodology for producing accurate probabilistic forecasts, based on training a recurrent autoregressive neural network model on a large number of related time series. DeepAR, which uses a real value as a past value. DeepAR uses a recurrent neural network (RNN) as a basic component and accepts past sequences and its covariates as an input [13][14][15].
DeepAR use Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) cells that takes the previous time points and covariates as input[16].

The evaluation metrics used to assess the performance of the proposed models were RMSE and MAPE which are mathematically represented by the following equations: Eq. (1) and Eq. (2).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y - y)^2} \quad (1)$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - y}{y}\right| * 100 \quad (2)$$

Where y is model predicted value, y_i is actual value.

3 Results

We first calculated the descriptive statistics of our data concerning the confirmed cases, these statistics are represented in the figures 7.

Mean	249k	
Std. Deviation	228k	
Quantiles	23.1k	Min
	59.6k	25%
	128k	50%
	472k	75%
	667k	Max

Fig.7. Descriptive statistics of the confirmed cases

The implementation of the models was done in Kaggle using python 3.0.
We then consider individual models such as LSTM and GRU. The structure of LSTM is as follows: LSTM layer - Exclusion layer - Dense layer. The structure of GRU is as follows: GRU layer and Dense layer. GRU and LSTM can have the same structures and parameters.
We divided the input data into 80% for training and 20% for testing and normalized them using MinMaxScaler, its mathematical representation is presented in (3).

$$Z_n = \frac{Z - Z_{min}}{Z_{max} - Z_{min}} \quad (3)$$

Where Z is original time-series data, $Z_n$is normalized time-series data, $Z_{min}$is minimum value in the timeseries, and Z_{max} is the maximum value of the time series.

At first, we implemented the models individually. For GRU and LSTM had respectively 2 layers. We add a dense layer to the model to connect each neuron to the next neuron. To overcome the explosive gradient or evanescent gradient problem, we used the ReLu activation function. In DeepAR we kept the same parameters that we selected with the individual models. This means that DeepAR with LSTM cell and DeepAR with GRU cell were implemented with two layers for each cell type. As an optimizer, we used Adam and RMSE to evaluate the models.

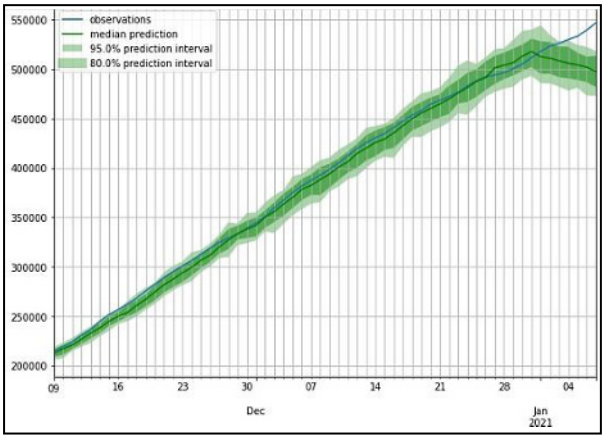

Fig.8. Fig. 8. DeepAR with LSTM cells "trainingdata"

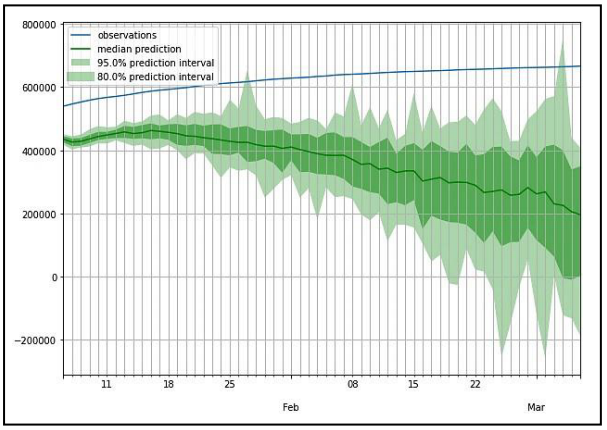

Fig.9. DeepAR with LSTM cells "testing data"

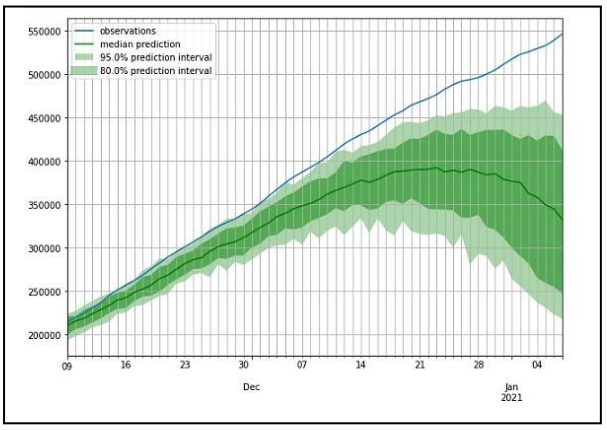

Fig.10. DeepAR with GRU cells "training data"

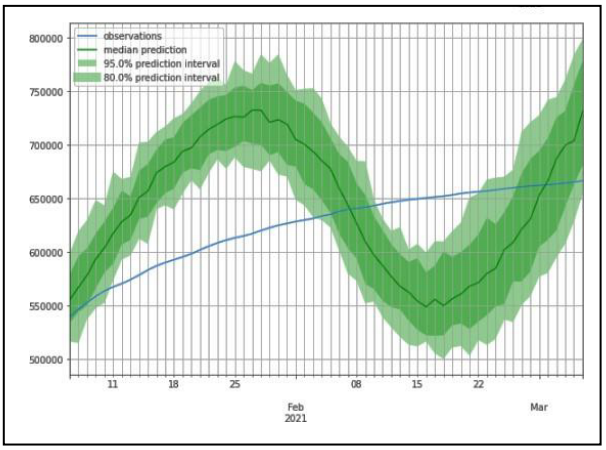

Fig.11. DeepAR with GRU cells "testing data"

Table 3. MAPE AND RMSE comparison of all models.

Models	MAPE	RMSE
LSTM	0.2583	26231
GRU	0.3483	30745
DeepAR- LSTM cells	0.009218	10801.75
DeepAR- GRU cells	0.01112	20745.75

From the figures we can see that the DeepAR model reacts well to the training phase, the data of this phase are linear. On contrary, in the test phase the model does not give very good results as in the training phase. This is due to the complexity of the data as we have linear and non-linear data. At the beginning the flow was increasing weakly but from December 2021 the flow is increasing rapidly. The problem we are confronted with when working with the covid data is to find a model that can detect these sudden and unexpected changes, and this is where the data starts to be complex and the model does not perform well.
By analyzing the results, we can see that the DeepAR-LSTM cells model has given good results, but the GRU, either in individual mode or by introducing it as a cell for the DeepAR, remains less efficient.

4 Conclusion

The COVID-19 data of some countries are not linear which is the case for India. From the results obtained we can see that the Auto-Regressive Recurrent Neural Network give good results compared to Recurrent Neural Network.
From our research and those cited in the literature, we can observe that we have not yet succeeded in having a model that detects the complex trend of Covid-19. This study has several limitations. Deep learning and machine learning models remain weak in detecting the complexity of this trend. What we see and propose as future work is to work with hybrid models like RNN with CNN to detect the complex trend of Covid flow.

References

1. "Worldometers.info," 2020, [Online]. Available: https://www.worldometers.info/.
2. "World Health Organization (WHO)," 2020. https://covid19.who.int/region/searo/country/in.
3. "REUTERS Covid-19 Tracker." https://graphics.reuters.com/world-coronavirus-tracker-and-maps/countries-and-territories/india/.
4. S. Hanumanthu, "Role of Intelligent Computing in COVID-19 Prognosis: A State-of-the-Art Review," *Chaos, Solitons FractalsInterdiscip. J. Nonlinear Sci. NonequilibriumComplex Phenom.*, p. 109947, 2020, doi: 10.1016/j.chaos.2020.109947.
5. F. Shahid, A. Zameer, and M. Muneeb, "Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM," *Chaos, Solitons Fractals Interdiscip. J. Nonlinear Sci. Nonequilibrium Complex Phenom.*, vol. 140, p. 110212, 2020, doi: 10.1016/j.chaos.2020.110212.
6. K. S. Bhangu, J. K. Sandhu, and L. Sapra, "Time series analysis of COVID-19 cases," vol. 1, no. November 2020, pp. 40–48, 2022, doi: 10.1108/WJE-09-2020-0431.
7. D. Tiwari, B. S. Bhati, and B. Nagpal, "Pandemic coronavirus disease (Covid-19): World effects analysis and prediction using machine-learning techniques," vol. 2, no. April, pp. 1–20, 2021, doi: 10.1111/exsy.12714.
8. F. Rustam *et al.*, "COVID-19 Future Forecasting Using," vol. 4, pp. 1–12, 2020, doi: 10.1109/ACCESS.2020.2997311.
9. H. Verma, S. Mandal, and A. Gupta, "Temporal deep learning architecture for prediction of COVID-19 cases in India," no. January, 2020.
10. N. Ayoobi, D. Sharifrazi, R. Alizadehsani, and A. Shoeibi, "Results in Physics Time series forecasting of new cases and new deaths rate for COVID-19 using deep learning methods," vol. 27, no. March, 2021, doi: 10.1016/j.rinp.2021.104495.
11. N. Science *et al.*, "Time series forecasting of Covid-19 using deep learning models : India-USA comparative case study," *Chaos, Solitons Fractals Interdiscip. J. Nonlinear Sci. Nonequilibrium Complex Phenom.*, vol. 140, p. 110227, 2020, doi: 10.1016/j.chaos.2020.110227.
12. P. T. Yamak, "A Comparison between ARIMA , LSTM , and GRU for Time Series Forecasting," 2017.
13. Y. Jeon and S. Seong, "Robust recurrent network model for intermittent time-series forecasting," *Int. J. Forecast.*, no. xxxx, 2021, doi: 10.1016/j.ijforecast.2021.07.004.
14. D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR : Probabilistic forecasting with autoregressive recurrent networks," *Int. J. Forecast.*, no. xxxx, 2019, doi: 10.1016/j.ijforecast.2019.07.001.
15. V. Flunkert, D. Salinas, and J. Gasthaus, "DeepAR : Probabilistic Forecasting with Autoregressive Recurrent Networks," 2017.
16. A. Tadjer, A. Hong, and R. B. Bratvold, "Machine learning based decline curve analysis for short-term oil production forecast," 2021, doi: 10.1177/01445987211011784.