

Research Article

Forecasting COVID-19 Pandemic Using Prophet, ARIMA, and Hybrid Stacked LSTM-GRU Models in India

Sweeti Sah,¹ B. Surendiran ,¹ R. Dhanalakshmi,² Sachi Nandan Mohanty,³ Fayadh Alenezi ,⁴ and Kemal Polat ⁵

¹Department of Computer Science and Engineering, National Institute of Technology Puducherry, Karaikal, India

²Department of Computer Science and Engineering, Indian Institute of Information Technology Tiruchirappalli, Trichy, India

³Department of Computer Science & Engineering, Vardhaman College Engineering (Autonomous), Hyderabad, Telangana, India

⁴Department of Electrical Engineering, Jouf University, Sakaka 72388, Saudi Arabia

⁵Department of Electrical and Electronics Engineering, Bolu Abant Izzet Baysal University, Bolu, Turkey

Correspondence should be addressed to Kemal Polat; kpolat@ibu.edu.tr

Received 4 November 2021; Accepted 8 April 2022; Published 5 May 2022

Academic Editor: Andrei Korobeinikov

Copyright © 2022 Sweeti Sah et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the proliferation of COVID-19, the world is in a terrible condition and human life is at risk. The SARS-CoV-2 virus had a significant impact on public health, social issues, and financial issues. Thousands of individuals are infected on a regular basis in India, which is one of the populations most seriously impacted by the pandemic. Despite modern medical and technical technology, predicting the spread of the virus has been extremely difficult. Predictive models have been used by health systems such as hospitals, to get insight into the influence of COVID-19 on outbreaks and possible resources, by minimizing the dangers of transmission. As a result, the main focus of this research is on building a COVID-19 predictive analytic technique. In the Indian dataset, Prophet, ARIMA, and stacked LSTM-GRU models were employed to forecast the number of confirmed and active cases. State-of-the-art models such as the recurrent neural network (RNN), gated recurrent unit (GRU), long short-term memory (LSTM), linear regression, polynomial regression, autoregressive integrated moving average (ARIMA), and Prophet were used to compare the outcomes of the prediction. After predictive research, the stacked LSTM-GRU model forecast was found to be more consistent than existing models, with better prediction results. Although the stacked model necessitates a large dataset for training, it aids in creating a higher level of abstraction in the final results and the maximization of the model's memory size. The GRU, on the other hand, assists in vanishing gradient resolution. The study findings reveal that the proposed stacked LSTM and GRU model outperforms all other models in terms of R square and RMSE and that the coupled stacked LSTM and GRU model outperforms all other models in terms of R square and RMSE. This forecasting aids in determining the future transmission paths of the virus.

1. Introduction

Coronavirus is a combination of RNA viruses that usually trigger diseases in animals and birds, but in humans, coronavirus may lead to infections in the respiratory tract. Coronavirus is a Latin word “corona” meaning “crown.” The name coronavirus was given by June Almeida and David Tyrrell in 1930 when they initially noticed and made a study on coronaviruses that infect humans. In 1960, human coronaviruses

were discovered [1]. COVID-19 first originated from the region of China, Wuhan, and then transmitted throughout the world. The name COVID-19 was given by WHO (World Health Organization) [2] and was named later SARS-CoV-2 by the International Committee on Taxonomy of Viruses [1]. So far, no particular treatment has been discovered. COVID-19 is a pandemic disease and is generated by the SARS-CoV-2 virus. In 80% of individuals, the pandemic illness is mild, 13% severe, and 6% serious. Fever, fatigue, and

a dry cough are the most common symptoms, while others may have body pain, a runny nose, a sore throat, shortness of breath, and diarrhoea [3].

In India, COVID-19 was first detected in Kerala on 20 January 2020 [4]. To raise awareness of the pandemic condition, we proposed a method of forecasting cases using several approaches such as recurrent neural network (RNN), long short-term memory (LSTM), gated recurrent unit (GRU), linear regression, polynomial regression, Prophet, autoregressive integrated moving average (ARIMA), and amalgamated stacked LSTM-GRU. As a result, individuals worldwide may take measures before the number of cases reaches a peak. Awareness can be raised by foreseeing proven cases in advance and holding up a mirror to the real world of a terrible scenario. This awareness can protect the life of a considerable number of people globally if precautions are taken beforehand.

The planned work of the paper is as follows. Analyses of the past study on the COVID-19 dataset using various techniques are given in Section 2. Section 3 presents the various datasets available on the internet for COVID-19. Section 4 discusses the methodology. Section 5 shows the COVID-19 experimental analysis of cases, showing the prediction for the upcoming 20 days using Prophet, ARIMA, and stacked LSTM-GRU. Section 6 illustrates the challenges faced due to COVID-19 in India overall. Finally, results, conclusion, and future scope are described further in Sections 7 and 8.

2. Related Works

The ARIMA model [5] is presented for forecasting confirmed cases. To develop the ARIMA model, a total of 41 days were gathered. In nations such as Italy, China, South Africa, Iran, and Thailand, they utilized the ARIMA model to produce two types of graphs: autocorrelation function graph (ACF) and partial autocorrelation graph (PACF). Finally, a predictive analysis of confirmed instances was displayed. During and after the lockdown period, the corona effect was studied, focusing on positive examples, and the Prophet model, a time series analytic model with superior performance in real-time data, was employed. They discovered that positive instances increased in a regulated manner during the lockdown period compared to the relaxation period, and they argued that the lockdown was justified with strict rules that may prevent the spread of positive cases in India [6]. COVID-19 new and total deaths were analyzed and projected. Time series prediction models such as ARIMA, SARIMA, and Prophet models were used to forecast. The best-fit models were built using the minimum values of Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Akaike Information Criterion (AIC), Mean Square Error (MSE), and Mean Absolute Percentage Error (MAPE). The model's fit was evaluated, and projections for the following 15 to 20 days were created for future usage from these time series models [7–9]. An autoregressive integrated moving average (ARIMA) was used to predict COVID-19 cases based on Johns Hopkins epidemiology data. Between January 20, 2020, and February 10, 2020, the

COVID-19 dataset is available. An autoregression (A.R.), a moving average (M.A.), and a seasonal ARIMA model make up the ARIMA model (SARIMA). Differences were favoured to stabilise the time series log transformation. The autocorrelation function (ACF) and partial autocorrelation function (PACF) correlograms were used to test the ARIMA model. ARIMA (1,0,4) outperformed ARIMA (1,0,3) in forecasting COVID-19 incidence, according to the findings.

The incubation duration, count of fundamental regenerations, and a reasonable number of days to cure were all parameters included in the COVID-19 study, which was based on existing Hubei epidemic data. They have also predicted progress in nations like South Korea, Italy, and Iran by presenting existing epidemics and controlling time by tracing transmission dates [10]. Identification of main applications of artificial intelligence for COVID-19 was used to explain COVID-19 analysis. They demonstrate the importance of technology in detecting many instances and forecasting the virus' harmfulness in the next few days by combining and analyzing all primary data. This crucial application was created to keep track of data on recovered, confirmed, or dead patients. The primary role in health monitoring was by A.I. to record the catastrophe of COVID-19 at various scales like epidemiology, medical, and molecular application [11].

COVID-19 has been investigated in nine countries, including India. In COVID-19, they utilized the ARIMA model to forecast the trend. Various ARIMA models were created using various ARIMA parameters, such as ARIMA (3,1,1) models, and the best models were chosen based on the lowest RMSE and MAE values [12]. On the dataset obtained from John Hopkins University repositories, the study of COVID-19 is described using prediction based on two models: SEIR and regression model, with a period of 30th January to 30th March. SEIR and regression models were used to compute and get the real work. Between the regression model and SEIP, the error rate was determined to be 2.01. The illness spread was calculated to be 2.02, which will help the authorities and physicians adjust policy over the next two weeks. These models can tune for predicting long-term intervals using these short-term interval predictions. By retrieving the dataset in a daily layoff, an algorithm can be created in the future to forecast the confirmed case count in a weekly and biweekly period [13].

The topic of the coronavirus' future circumstances was explored. They utilized stacked LSTM, convolutional LSTM, bidirectional LSTM, and month-ahead forecasting. Convolution LSTM outperforms the other two models in terms of forecast accuracy, and the forecast was made with less error [14]. The interplay between various prevalent temperatures in confirmed, suspected, and deceased victims is used to explain COVID-19 analysis. *K*-means clustering was utilized as the machine learning technique, and it was used in the WHO dataset from various parts of China. They discovered that not just the temperature parameter was involved in the circulation of the COVID-19 epidemic after analyzing the information. Future work includes analysis of a whole world dataset of COVID-19 and considering more attributes [15].

Analysis of COVID-19 was explained using an enhanced adaptive neuro-fuzzy inference system (ANFIS) using the Salp Swarm Algorithm (SSA) and Enhanced Flower Pollination Algorithm (FPA). SSA is used in order to avoid the drawback of FPA. The primary purpose of FPASSA-ANFIS was to enhance the performance and is being evaluated using data of WHO (World Health Organization) for future prediction, and also, the model FPASSA-ANFIS was correlated with various other models in order to show good performance as compared to error metrics and computing power [16] and applied six statistical and deep learning algorithms driven by time series techniques to investigate the influence of corona. As a result, ARIMA and TBAT outperformed ARIMA and TBAT in terms of prediction accuracy [17].

COVID-19 analysis is described by predicting released cases, confirmed cases, and death cases on the data using an LSTM-GRU-based RNN model, which produces good results and can undoubtedly recheck the existence of COVID-19 instances. The goal of the recurrent neural network-based model is to provide an automated tool for insisting, assuming the current pandemic scenario, and aiding the government, determining the harshness and health workers to perform for a better decision-making law [18]. The study of COVID-19 was described by utilizing multiple-step prediction to forecast the aggregate confirmed case curves of coronavirus cases across China from 20 January 2020 to 20 April 2020. Combining and subgrouping cities projected the period points of the predicted transmit dynamic curves from January to April 2020 [19].

Shastri et al. [14] evolved an ARIMA model for each nation with a lower RMSE to forecast the evolution of the 145 countries divided into six regions. The ARIMA model allows for the prediction of a virus' behavior. With an RMSE mean value of 144.81, the author postulated a relationship between countries with similar geographic areas and expected COVID-19 cases. As a consequence, the model discovered a correlation between the expected error and the population of each million people. In order to identify the model's optimal parameter, the model was evaluated with 10% of the actual data.

Pandey et al. [13] work is aimed at using LSTM machine learning and GRU in Python to foresee COVID-19 data in Indonesia. Two datasets were utilized as a comparison from other nations with strong associations. The dataset can be found on the ourworldindata.org website. The LSTM model with epoch 15 and an RMSE of 68,417 takes less time to process and is more accurate than the GRU model with an RMSE of 90,173. Indonesian COVID-19 data correlates well with Azerbaijan, Bangladesh, Bhutan, Cape Verde, Curacao, Slovenia, South Africa, and Thailand. The epoch features of LSTM and GRU present a challenge because the amount of COVID-19 data is relatively small.

Bandyopadhyay and Dutta [18] study provided an overview of the major forecasting methods for predicting new COVID-19 cases. In this context, researchers looked at the pandemic's dynamic evolution through time using univariate time series models. Furthermore, incorporating multivariate time series forecasting with weather and daily test data was also used to study the impact of external factors

on COVID-19 development. Finally, using MAE, RMSE, and MAPE to analyze the outcomes presented an ensemble learning model based on LSTM and GRU. The results show that the ensemble technique performed well in comparison to other models.

In contrast to these existing approaches, we have proposed a novel hybrid model, which is a stacked LSTM-GRU optimized model. The major goal of the hybrid model is to forecast the COVID-19 cases with improved performance giving better efficiency. This hybrid model is distinct from the existing models in terms of performance and minimum loss while predicting the COVID-19 cases. The model is optimized using Adam optimizer, and the activation function used is ReLu, and the learning rate used is 0.01. Our hybrid model performs better than the Prophet and ARIMA model with minimum loss. Not much work has been done in applying hybrid models to predict COVID-19 cases. It is well known that hybrid models generally perform well over standalone models. So, to enhance the accuracy of the COVID-19 prediction, hybrid models need to be explored.

3. Various Dataset Sources of COVID-19

Data is always essential in investigating, studying, and responding to public health emergencies, especially in the situation of a worldwide disaster. Access to datasets and tools to evaluate that data at a large scale is becoming increasingly important in the research process. It is essential in the worldwide response to the new coronavirus. These datasets eliminate obstacles and enable rapid and easy access to vital information, removing the need to search for enormous data files on board. The datasets are available to researchers. These datasets' contents are exclusively made available to the public for educational and research reasons. Table 1 shows the various other data sources for COVID-19. There are many datasets of COVID-19 available on the internet through various sources as listed below which contain COVID-19 count of cases, text data, radiology pictures, Twitter data, and biological sequences [20].

4. Methodology

Kaggle is a great website for learning about machine learning and practicing with R and Python. The best element of Kaggle is the kernel, which assists newbies by bringing everything together in one spot, such as an editor and a dataset with high processing power. Kaggle is the best site for finding datasets on any topic, and it has a variety of COVID-19 datasets. In our proposed work, the dataset is referred from Kaggle (up to 12th Dec 2020), the world's largest dataset repository. This dataset is in .csv format. The dataset has many attributes related to COVID-19 such as date, time, state/union territory, confirmed foreign national, confirmed Indian national, cured, deaths, and confirmed cases. These attributes best describe the COVID-19 cases in India.

Figure 1 represents the per day statistics of cured cases, death cases, and confirmed and new cases in India. The graph is presented up to 12th Dec 2020. The yellow color presents confirmed cases, orange represents death cases,

TABLE 1: Various other data sources for COVID-19.

Data source	Type of data	Link
WHO	This dataset is provided by WHO (World Health Organization), providing up-to-date data across the world. This is the official website to get the COVID-19 case live updates and get information about outbreaks	https://systems.jhu.edu/research/public-health/ncov/
Kaggle	Kaggle is the largest data science community to provide an open-source dataset globally. Every day, the count of new cases increases around the world. This collection contains data from India's states and also union territories on a daily basis	https://www.kaggle.com/sudalairajkumar/covid19-in-India
Johns Hopkins University	Johns Hopkins University's Center for Systems Science and Engineering team maintains the dashboard and dataset (CSSE). Since January 22, 2020, it has been posting data on confirmed cases and deaths for all nations. Every day, JHU updates its data many times. This information comes from governments and national and subnational agencies all across the world; a complete list of data sources for each country may be seen on the Johns Hopkins GitHub page. It also makes its data available to the general public there [21]. This provides public access for testing data, resources, and analysis. Web-based outline for cases globally	https://systems.jhu.edu/research/public-health/ncov/
C. R. Well's GitHub	Open-source, writable linked data repository, developed by Thomson Reuters and Refinitiv and is used as a central knowledge graph database [22]	https://github.com/WellsRC/Coronavirus-2019
China CDC (CCDC)	This dataset shows the daily count of cases particularly in China consisting of confirmed, asymptomatic, and recoveries	http://weekly.chinacdc.cn/news/Trackingtheepidemic.htm
DataHub	DataHub provides time series data on COVID-19 cases. The data is in .csv format and is updated on a daily basis. It comes from this upstream repository, which is managed by the fantastic team at Johns Hopkins University's Center for Systems Science and Engineering (CSSE), who have been collecting data from all around the world since the beginning	https://datahub.io/core/covid-19
Conference of State Bank Supervisors	This provides a country-level map of COVID cases in the U.S., which is revised hourly	https://www.csbs.org/information-covid-19-coronavirus
U.S. CDC	This provides the cases in U.S. Consumers, financial institutions, and fellow regulators will get timely information on actions taken to support communities during the COVID-19 epidemic from state regulators. This website will be updated with all public CSBS updates related to COVID-19	https://www.csbs.org/information-covid-19-coronavirus
Italy Ministry of Health	This provides the COVID-19 cases in Italy	https://www.salute.gov.it/portale/nuovocoronavirus/homeNuovoCoronavirus.jsp
U.S. National Institutes of Health	This provides overall COVID-19 cases in the U.S.	https://covid19.nih.gov/
W. Zeng's website	This website provides the global COVID-19 cases country-wise	http://open-source-covid-19.welleizeng.com/
COVID-19 Radiography Database	This provides COVID-19, 45k scholarly articles, and its family. This publicly available dataset is made available to the world's researchers to use recent developments in natural language processing (NLP) and other A.I. approaches to bring about fresh intuition in support of the ongoing fight against this disease	https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge

TABLE 1: Continued.

Data source	Type of data	Link
A. G. Chung's GitHub—Actualmed Initiative	It includes an X-ray image dataset of the chest initiative of COVID-19	https://github.com/agchung/Figure1-COVID-chestxray-dataset/tree/master/images
Georgia State University's Panacea Lab	This chatter in Twitter provides the different linguistic datasets. The data obtained from the stream includes all languages, but English, Spanish, and French have the highest incidence	http://www.panacealab.org/covid19/
NCBI GeneBank	This provides the sequence of SARS-CoV-2	https://www.ncbi.nlm.nih.gov/sars-cov-2/
The GISAID Initiative	This is the global proposal revealing all types of cold data	https://www.gisaid.org/
China National GeneBank	This provides the sequence database of COVID-19	https://db.cngb.org/datamart/disease/DATAdis19/
EMBL-EBI	This provides the sequence of COVID outbreak isolates and record	https://www.covid19dataportal.org/

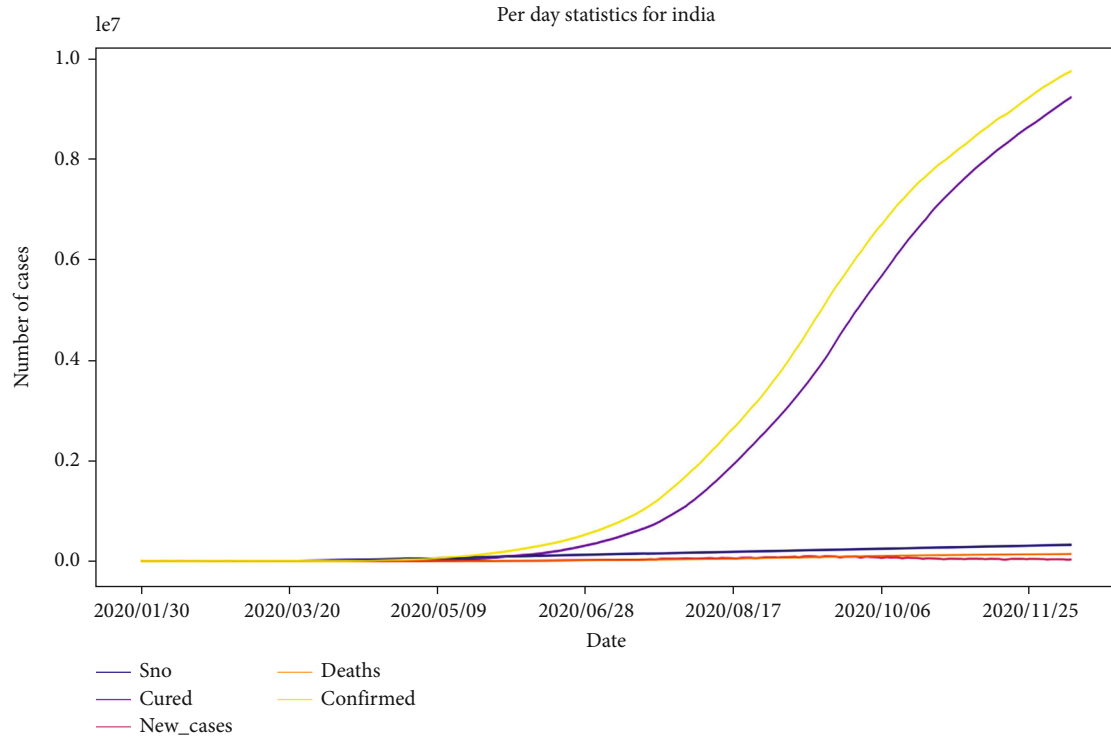


FIGURE 1: Per day statistics of India (considering all cured cases, death cases, confirmed cases, and new cases), where the x -axis is “date” and y -axis represents “number of cases.”

pink represents new cases, and purple represents cured cases. The x -axis is labeled as dates, and the y -axis is labeled as the number of cases. This graph is represented without the marker value.

Figure 2 mainly focuses on death, cured, and confirmed cases. This graph represents the per day statistics of cases in India with the marker value. The yellow color shows death cases, the green color shows cured patients, and the purple color shows the confirmed cases. The x -axis is labeled as to date, and the y -axis is labeled as the number of cases.

Figure 3 shows the per day statistics for new cases in India, where the x -axis shows the date up to 12th Dec 2020, and the y -axis shows the number of new cases. This graph shows the visual view of how daily the new cases are increasing.

4.1. Data Analysis Model. The model of data analysis phase includes reading the .csv file, fetching the number of attributes, computing the shape of the dataset, and visualizing the information of the dataset, then grouping the number of confirmed cases, deaths, and cured according to their date and summing them all. Next was retrieving the data of all states and finally plotting the graph for per day statistics in India showing all confirmed, death, and cured cases with marker and without marker values followed by the diagram showing the count of new cases as shown in Figure 4. The visualization and modeling of COVID-19 are crucial and can be extremely useful in the current world condition. Therefore, we used multiple graphs to display the data spe-

cifically for India in this paper to assess the trend of COVID-19 growth.

The following are the steps for data analysis:

Step 1: the data analysis phase includes first importing necessary libraries for analyzing the dataset like NumPy, Pandas, matplotlib, and DateTime.

Step 2: these libraries help the programmer to analyze any data. As the dataset is in .csv format, we import the dataset.

Step 3: we first read the dataset and then store it in pandas data frame.

Step 4: now, the per day statistics graph is plotted to visualize the data. The graph which is used for analysis includes a pie chart and a bar graph.

The infectious disease spreads across communities in predictable ways, which are governed by the pathogen’s transmission mechanisms and the relations that the pathogenic agent can employ to propagate around a society. Thus, even small and temporary risks to minute molecules in the air, in confined locations where sufferers breathe, can be acceptable to perpetuate infectious diseases where transmission is forthright from person to person and airborne [23–42].

In Figure 5, the blue color shows confirmed cases, the green color shows the migrated/cured/discharged cases, and the red indicates death cases. Finally, Figure 6 describes the top five states of India that are majorly affected by COVID-19. These bar graphs show the total confirmed cases, cured/discharged/migrated, and death cases.

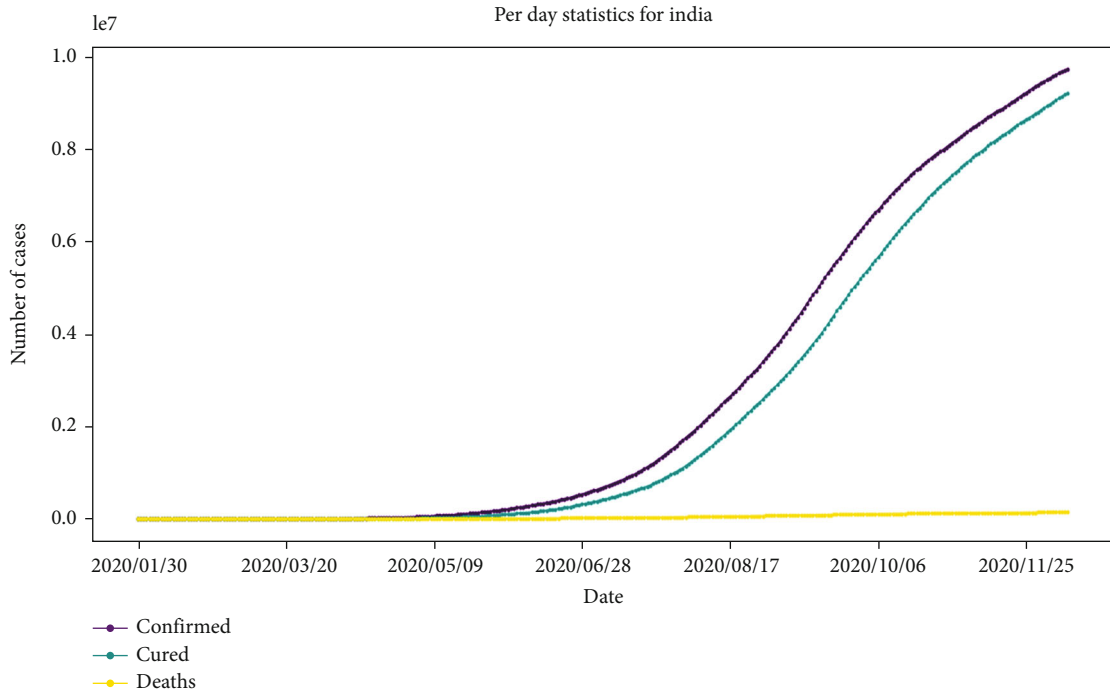


FIGURE 2: Per day statistics of India with marker values showing no. of confirmed, cured, and death cases.

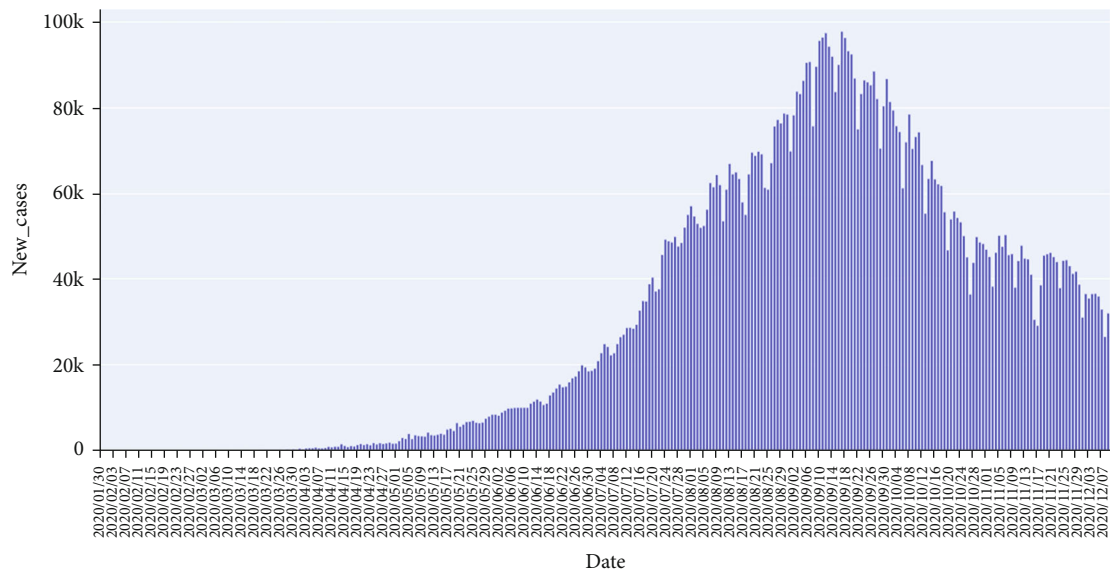


FIGURE 3: Per day statistics for new cases in India.

5. Experiment Results and Analysis

Our predictive proposed models are implemented with the required materials as described in Table 2. Table 2 gives the implementation detail of models used.

These models target dividing the population into multiple health states to understand the dynamics of a disease propagation process. The models used are Prophet, ARIMA, and hybrid stacked LSTM-GRU optimized models. The

Prophet framework features its data frame, making it simple to work with time series and seasonality data. Two fundamental columns are required in the data frame. “ds” is one of these columns, and it records the date-time series. The relevant values in the time series data frame are stored in the “y” column [43].

ARIMA models are statistical models that can be used to examine and forecast time series data. It gives a simple yet powerful way of creating skilled time series forecasts by

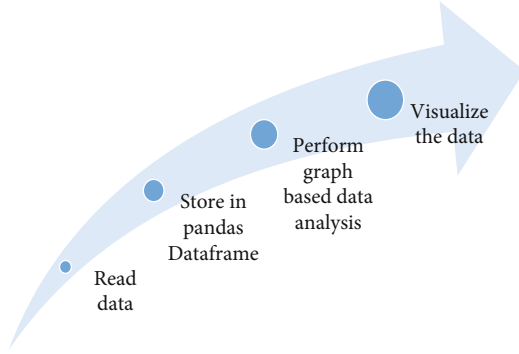


FIGURE 4: Model of the data analysis phase.

explicitly catering to a set of common structures in time series data. It is a more complex version of the basic autoregressive moving average that incorporates the concept of integration [44].

The stacked LSTM-GRU optimized hybrid model differs from the existing models in terms of performance and minimum loss while predicting the COVID-19 cases. The model is optimized using Adam optimizer, and the activation function used is ReLu, and the learning rate used is 0.01. Our hybrid model performs better than Prophet and ARIMA models with minimum loss. Not much work has been done in applying hybrid models to predict COVID-19 cases. It is common knowledge that hybrid models outperform standalone models. As a result, hybrid models must be investigated to enhance the performance of the COVID-19 forecast.

5.1. Prediction through Prophet. A team on Facebook has released Prophet. This system is used for predicting a series of times. This majorly works well with time series, and it is tough to lose data values and transfer in the trend and outlier [45]. Prophet includes data analysis flow as discussed in Figure 7.

Prophet is an additive regression model. The components of the Prophet model consist of a logistic gain curve trend, and it finds variations in trends by choosing the data's increasing points. A yearly seasonal component can be simulated with the help of the Fourier series. A seasonal weekly element can be represented using dummy variables, and the user can be given a list of holidays [46]. It works well with time series with a lot of seasonal variation and historical data from different seasons. Prophet is forgiving of missing data and trend shifts, and it handles outliers well most of the time [47].

It is given by an equation (Ankit Choudhary et al., 2018):

$$\mathbf{y}(\mathbf{t}) = \mathbf{g}(\mathbf{t}) + \mathbf{s}(\mathbf{t}) + \mathbf{h}(\mathbf{t}) + \epsilon_t, \quad (1)$$

where $\mathbf{g}(\mathbf{t})$ is for demonstrating nonperiodic alterations in period series; the piecewise linear or logistic curve is used. $\mathbf{s}(\mathbf{t})$ are the changes that are done periodically like weekly, yearly, and seasonally. $\mathbf{h}(\mathbf{t})$ is the effect of holidays that the user gives with irregular schedules. ϵ_t is any unusual

changes; this is the error term accounts and not accommodated by the model.

Prophet requires the variable names y (target) and ds (Date/Time) in the time series. Thus, the Prophet model performs best with data from numerous seasons and time series with significant seasonal influences. Furthermore, in comparison to traditional exponential smoothing methods, the Prophet model can extract trends and periodic signals across a wider range of time scales and has no limits on measurement spacing regularity. Thus, the Prophet model simplifies various time series analyses [48].

Some Prophet trend parameters are as follows [49]:

- (1) *Growth.* Growth is either linear or logistic to show the trend. A piecewise linear curve is fitted over the direction or nonperiodic section of the series to model the trend. The linear fitting procedure shows that spikes and missing data are minimized.
- (2) *Changepoints.* It includes the list of dates. These changepoints are chosen at random. However, if necessary, a user can manually feed the changepoints. The fit becomes more adjustable as the count of changepoints allowed increases. When working with the trend component, an analyst may encounter one of two issues: overfitting.
- (3) *Changepoint Prior Scale.* This is the factor for transforming the elasticity of automatic selection on changepoint. A parameter known as the changepoint prior scale could alter the trend flexibility and solve the above two concerns. A higher number will give the time series a more flexible curve.
- (4) *Daily, Weekly, and Yearly Seasonality.* This contains seasonal patterns like hourly-wise, daily-wise, or yearly-wise data. Prophet uses the Fourier series to fit and forecast the impacts of seasonality and give a flexible model. The Fourier order N is an important parameter to choose here, which determines whether more frequency variations can be represented. If the user considers more frequency, components in a time series are just noise.

Figures 8 and 9 show the prediction graph for the number of confirmed and active cases in India from 11th Dec '20 to 31st Dec '20. The black dots represent the original data values. The blue color line shows the predicted values. The light blue color shows the upper and lower limits for the prediction curve, where ds represent the date and y represents the target value of the number of cases in India.

5.2. Prediction through ARIMA. This is referred to as the "autoregressive integrated moving average." It is a model of linear regression. This is also useful for forecasting time sequences. It employs statistical models to forecast future values based on historical data. The notation for this model is as follows: (p, d, q) . The autoregression portion of the parameter p indicates that it aids in the incorporation of prior ideals into the model means to incorporate the amount

of difference in time series, and lastly, q is the average running part. This helps in setting the error of the model. This is given by [50, 51]

$$\text{ARIMA}(\mathbf{p}, \mathbf{d}, \mathbf{q}) (\mathbf{P}, \mathbf{D}, \mathbf{Q}) \mathbf{m}, \quad (2)$$

where \mathbf{p} is the order of A.R. term, no. of lags used as a predictor; \mathbf{d} is the no. of differencing required to make the time series data stationary, if $d = 0$, then stationary; \mathbf{q} is the order of M.A. term, no. of lagged predicted errors which go into the ARIMA model; \mathbf{m} is each season of the no. of periods; and $(\mathbf{P}, \mathbf{D}, \mathbf{Q})$ signifies the periodic section of the time series that is (p, d, q) .

ARIMA comprises a data analysis flow as shown in Figure 10, training the data and fitting the ARIMA model by installing the packages. This helps in predicting the next N days. Hence, the predicted data can be visualized in a better way.

5.2.1. AR (Autoregression). The AR, MA, ARMA, and ARIMA models are used to predict the observation at $(t + 1)$ using past data from earlier time spots. However, it is vital to ensure that the time series remains stationary over the observation period's historical data. Autoregression (A.R.) is a type of model that calculates the regression of past time series and estimates the current or future values in the series. The moving average (M.A.) model calculates the residuals or errors of past time series and then determines the current or future values in the series. The AR and M.A. models are merged in the ARMA model. For forecasting future data points of the time series, this model considers the impact of previous lags as well as residuals [52].

The mathematical formula for the ARIMA model is given below by [50]

$$y_t = \alpha + \beta_1 y_t + \beta_2 y_{t-1} + \dots + \beta_p y_{t-p} + \epsilon_1, \quad (3)$$

where y_t are functions of the lags of y_t , α is the intercept term, β_1 is the coefficient of lag1, and y_{t-1} is lag1 of the time series.

5.2.2. M.A. (Moving Average). The mathematical formula is given below by [50]

$$y_t = \alpha + \epsilon_1 + \varphi_1 \epsilon_{t-1} + \varphi_2 \epsilon_{t-2} + \dots + \varphi_q \epsilon_{t-q}, \quad (4)$$

where ϵ_1 are errors.

In general, the ARIMA model formula can be expressed as [50]

$$\begin{aligned} \text{Predicted value } y_t = & \text{constant value} + \text{linear consolidation lags of } Y \\ & + \text{linear consolidation of lagged forecast errors.} \end{aligned} \quad (5)$$

The parameter of the ARIMA model was approximated using the ACF (autocorrelation function) graph and PACF (partial autocorrelation function) [8, 9].

ARIMA shows the prediction of future values in a linear fashion also called the Box-Jenkins model and is given by the equation [53]

$$y_t = \theta_0 + \varphi_1 y_{t-1} + \dots + \varphi_p y_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \dots - \theta_q \epsilon_{t-q}, \quad (6)$$

where y_t is the actual data point at particular " t " time, θ and φ are the coefficient model, \mathbf{p} and \mathbf{q} are form of the autoregressive and moving average, and integer value ϵ_t is an error.

Various steps involved in the ARIMA model are described below [54]:

- (1) *Checking Stationary.* If the time series has a seasonal component or trend, then before using ARIMA to forecast, it must be made stationary.
- (2) *Difference.* Through differencing, the time series needs to be stationarized in case of nonstationary. Take as many differences and ensure to check seasonal differences as well.
- (3) *Filter Out a Validation Pattern.* This is to prove how precise the model is with the use of train test validation.
- (4) *Select the A.R. and M.A. Terms.* In order to include A.R. term(s), M.A. term(s), or both, use ACF and PACF.
- (5) *Build the Model.* This includes building the model and setting the duration count to predict N , where N 's value depends on our needs.
- (6) *Validate the Model.* This is comparing the forecasted value to the actual value in the validation sample.

Figures 11 and 12 show the prediction graph for the number of confirmed and active cases in India from 11th Dec '20 to 31st Dec '20. The orange color represents the original data values. The blue color line shows the predicted values, where the x -axis represents the date and y -axis represents the number of cases in India.

5.3. Prediction through Stacked LSTM-GRU Neural Network. LSTM is an exclusive type of RNN introduced by Schmidhuber and Hochreiter [55]. It stands for long short-term memory and is a repetitive network for forecasting the value by learning long observations [56]. LSTM solves the RNN's long-term dependence problem. The input gate, the middle that is the forgotten gate, and the output gate are the three gates of the LSTM. These gates control only the information flow between the cells and the neural structure [57]. The cell state, employed as a carrier of information, is a critical notion in LSTM. Forget gates determine which information should be preserved and which should be discarded. The input gate is used to determine which fresh input, data, and cell state to store. The output gate determines which conditions of the cell become outputs [55].

The stacked model includes data loading, splitting the data into train/test, reshaping data for LSTM, creating the

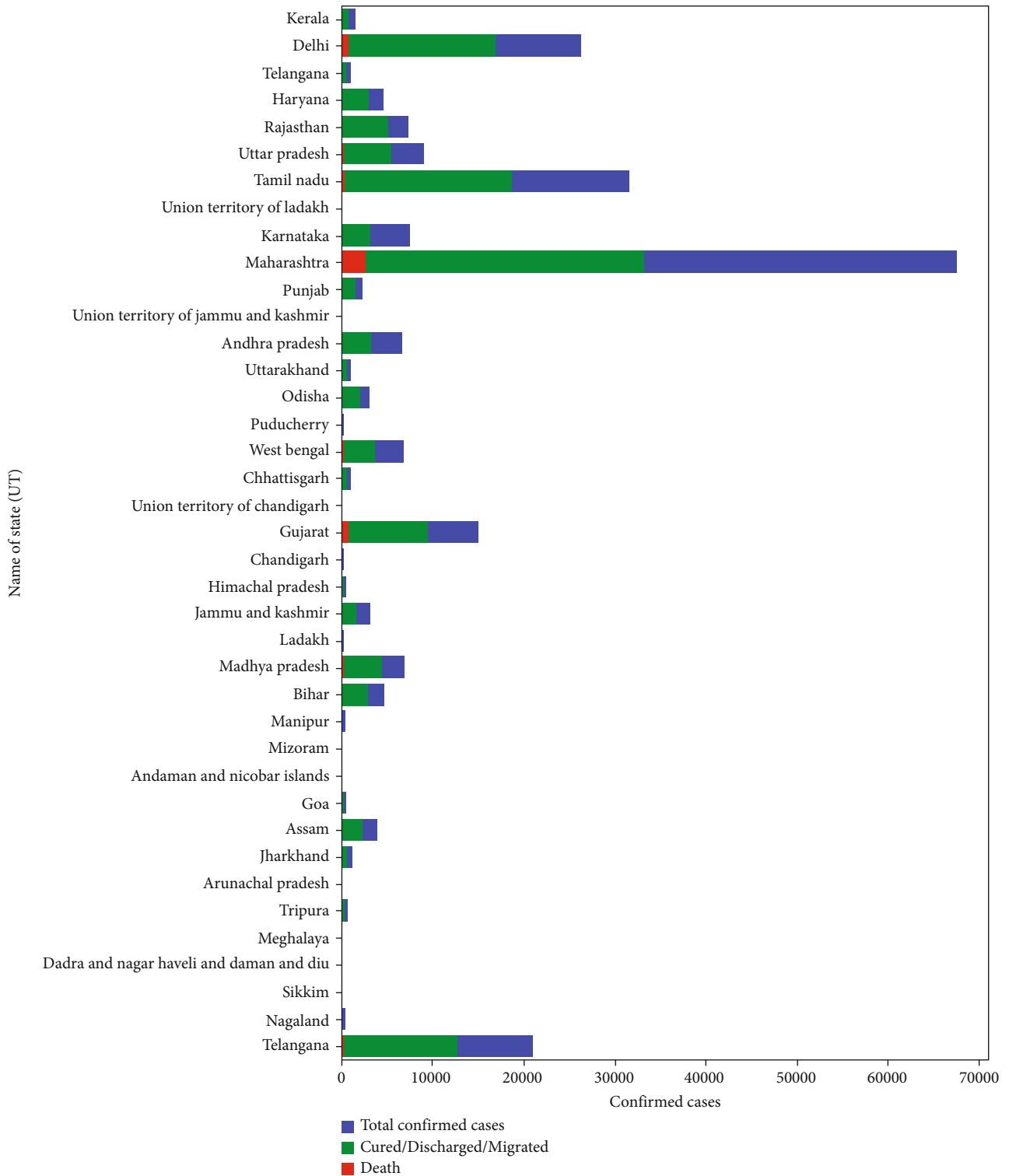


FIGURE 5: Bar graph showing the percentage of confirmed/cured/death cases per state (India).

stacked LSTM-GRU model, filling the data into a model, and finally making a prediction, as shown in Figure 13.

The first step is to comprehend the differences between a neural network (N.N.) and a recurrent neural network (RNN), with RNN referring to multilayer neuron networks. Although neural networks are commonly used for

classification, grouping, and regression issues, they have one important drawback: they can only use the current input and cannot incorporate the model's prior output. If the input is sequencing input, then we have to consider the previous state also. N.N. do not perform well with sequence input.

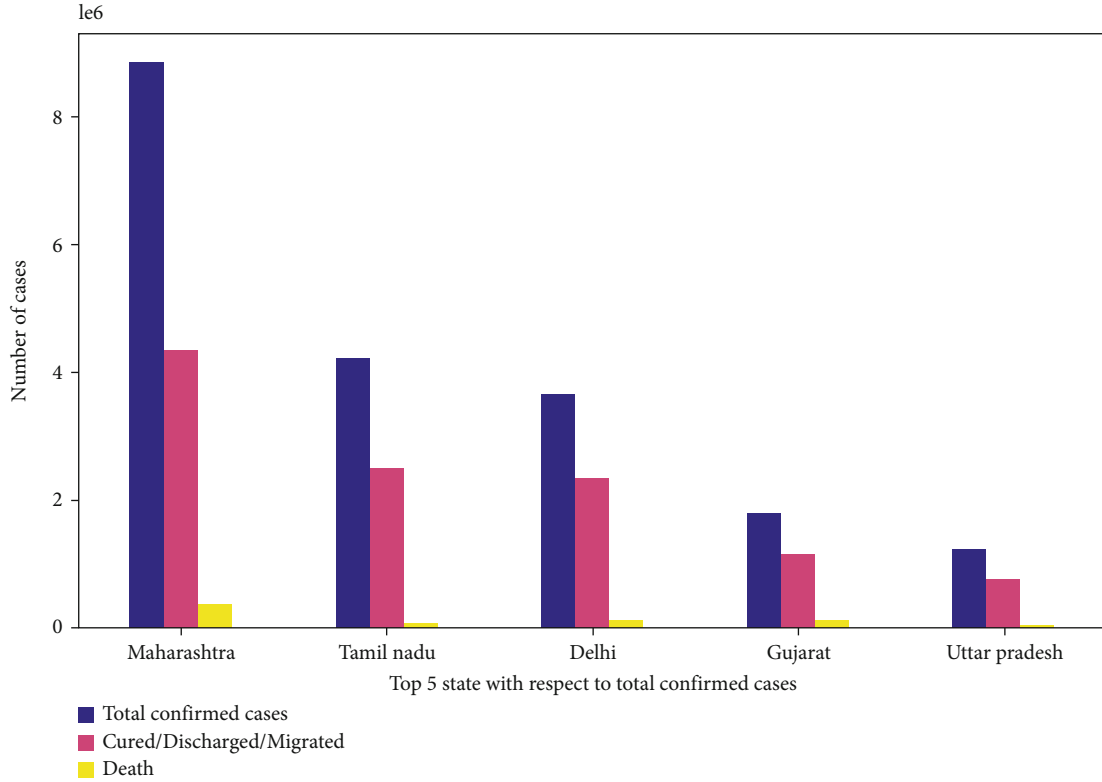


FIGURE 6: Top five states of India (cases of COVID-19).

TABLE 2: Implementation detail of models used.

Dataset used	COVID-19 Indian dataset from Kaggle (up to 12 th Dec 2020)
Models used	Prophet, ARIMA, stacked LSTM-GRU
Language	Python
System software	Co-Lab
Libraries used	Pandas, NumPy, Sklearn, Matplotlib, FbProphet, ARIMA, LSTM, Sequential, Dense, Math, DateTime

Recurrent network handles this problem. It also takes into account the model’s current input and prior output. For prediction problems, machine translation, speech recognition, and text summarization, recurrent networks are utilized.

It was initially proposed for language models in 1997. On the output layer of our dataset, a linear regression layer is used (LSTM cell). It is given by [49]

$$h_1 = H(W_{hx} + W_t + W_{hx}h_{t-1} + b_h), \quad (7)$$

$$P_t = W_{hy}W_{hx}y_{t-1} + b_y, \quad (8)$$

where $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ denotes the input time series, $\mathbf{h} = (\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n)$ denotes the hidden state of memory cells, $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n)$ denotes the output time series, \mathbf{W} denotes weighted matrices, and \mathbf{B} denotes the bias vectors.

GRU is an LSTM (long short-term memory) version with only one updated gate consisting of input, forget, and reset gates. Because there is no extra memory cell to store data, it can only control information already within the unit. GRU is a new creation of RNN that is comparable to LSTM in that the update gate is identical to the forget gate and the input gate of LSTM [58]. In this approach, the input and output gates are combined to form a single gate called the update gate. The update gate decides how the unit changes its content by interpolating between prior and candidate activation. Memory modules, rather than hidden units, are used in GRU, ensuring that the gradient does not evaporate or explode after many repeats. The suggested model captures the learning elements of both the LSTM and GRU models due to its ability to capture the distinct benefits of both models in a single framework. The GRU model, which processes the time series data, uses the stacked LSTM output as an input. The GRU model’s generated output represents the output of the hybrid stacked LSTM and GRU model. Using stacked LSTM in a neural network to improve prediction accuracy while learning a large number of temporal features is a frequent practice. They also help to generate a better level of representation of the incoming sequence data over time. Although the stacked model requires a large dataset for training, it aids in increasing the model’s memory size [59].

Now, the curve in Figure 14 shows the train prediction and test prediction (8:2) with the original curve.

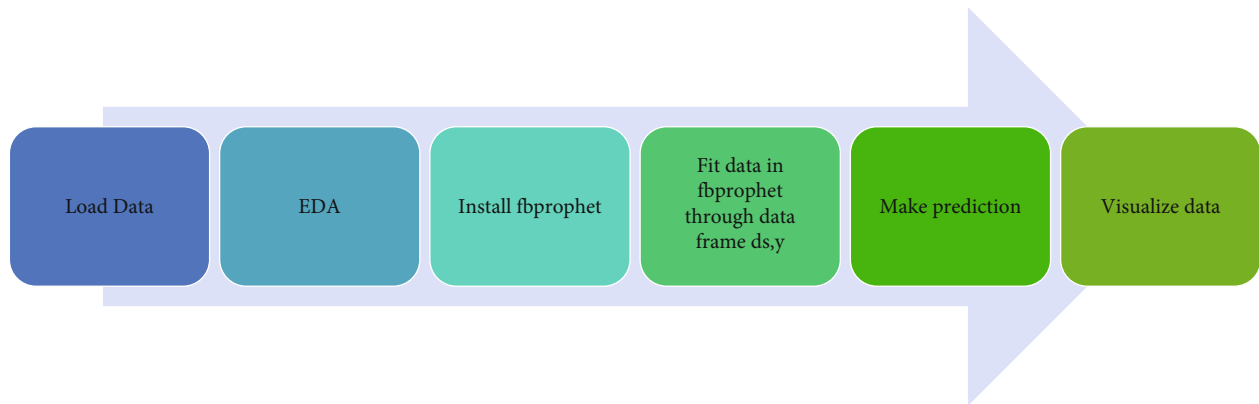
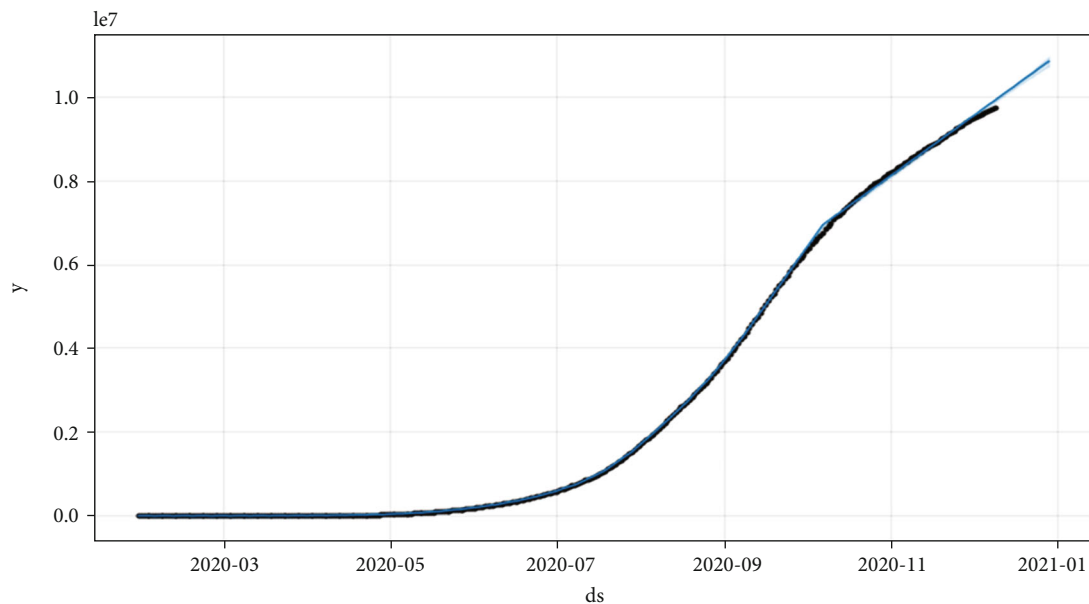


FIGURE 7: Working model of Prophet.

FIGURE 8: Prediction through Prophet for a count of confirmed cases (for the next 20 days), where the x -axis is “ds” and y -axis is “ y ” which is the target.

Figures 14 and 15 show the prediction graph for the number of confirmed and active cases in India from 11th Dec ‘20 to 31st Dec ‘20. The blue color represents the original data values. The orange color line shows the trained values. The green color shows the prediction curve, where the x -axis represents the date and y represents the number of cases in India. The dataset was divided into train and train ratio that is 8:2. Further, the hybrid model is optimized using Adam optimizer for more accurate results.

5.4. Adam Optimizer and Activation Function. With the inclusion of momentum, Adam is a cross between RMSprop and stochastic gradient descent. It uses squared gradients to scale the learning rate, similar to RMSprop. However, it uses a moving average of the gradient rather than the gradient itself to take advantage of momentum,

identical to SGD with momentum. Adam is a learning rate algorithm that calculates individual learning rates based on a variety of variables. Adam uses adaptive moment estimation to modify the learning rate for each weight of the neural network by estimating the first and second moments of the gradient [60]. When constructing your neural network, the learning rate may be the most crucial hyperparameter. As a result, it is critical to understand how to examine the effects of the learning rate on model performance and to develop an intuition for the learning rate’s dynamics on model behavior [8, 61]. Because the derivative of the ReLu function is 1 for positive input, it can help deep neural networks train faster than typical activation functions. As a result, deep neural networks do not require additional time during the training phase to compute error terms due to a constant [62].

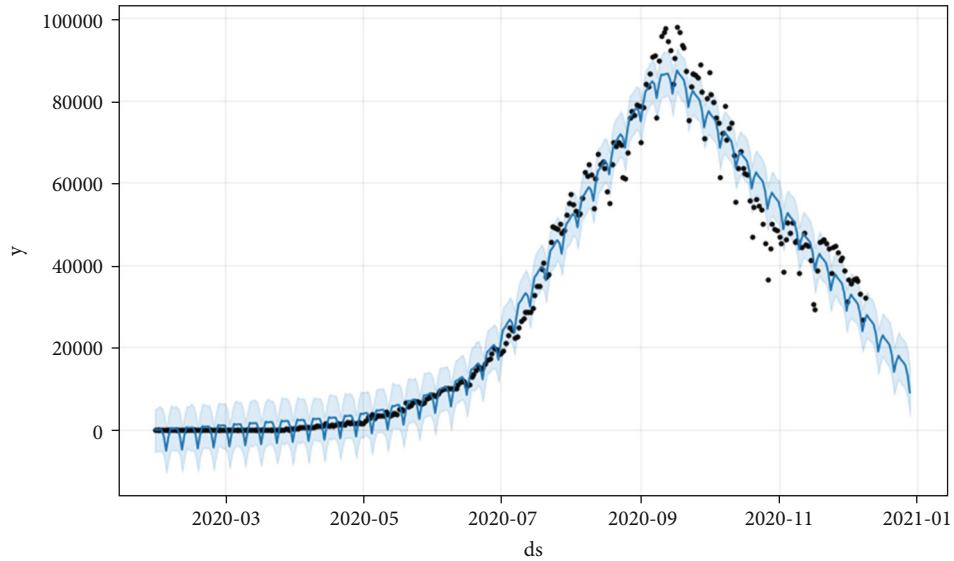


FIGURE 9: Prediction through Prophet for the number of new cases (for the next 20 days), where x -axis is “ds” and y -axis is “y” which is target.

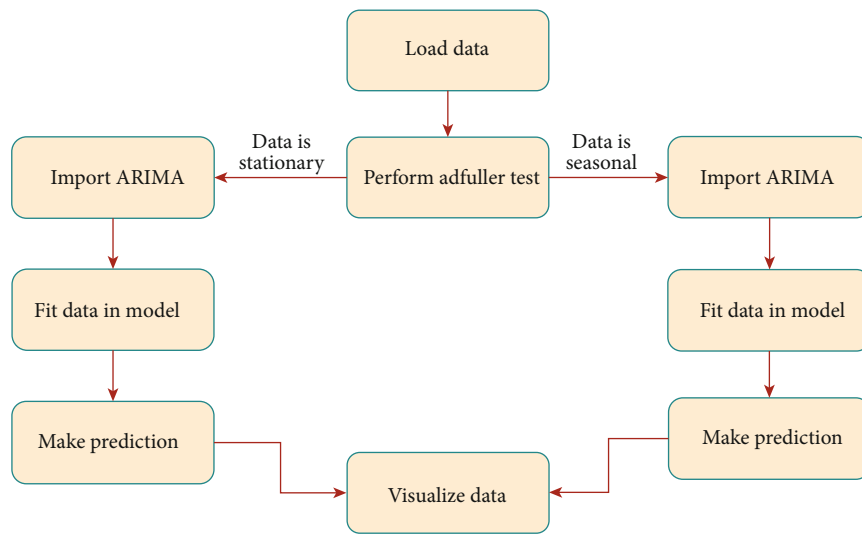


FIGURE 10: Working model of ARIMA.

5.5. Comparison between Prophet, ARIMA, and LSTM-GRU. The comparison between models are specified below [43, 47, 48, 50, 51, 59, 63]:

- (1) ARIMA is a type of autoregressive forecasting in which the lag values and error terms are used to fit a linear regression line. Facebook’s FbProphet is known for being simple to use but challenging to master, especially if you want to learn more about what is going on with the model. The Fourier transform is at the heart of it
- (2) Before utilizing ARIMA, you must have a basic understanding of statistics. You can achieve a rea-

sonable forecast on messy data using FbProphet with no manual effort

- (3) ARIMA is susceptible to white noise and nonstationary signals, whereas FbProphet is resistant to outliers, missing data, and significant changes in your time series
- (4) The autoregressive component p , the integrated component d , and the moving average component q can all be experimented with in ARIMA. Linear or logistic growth, holidays, seasonalities, and changepoints are all parameters that can be experimented with in FbProphet

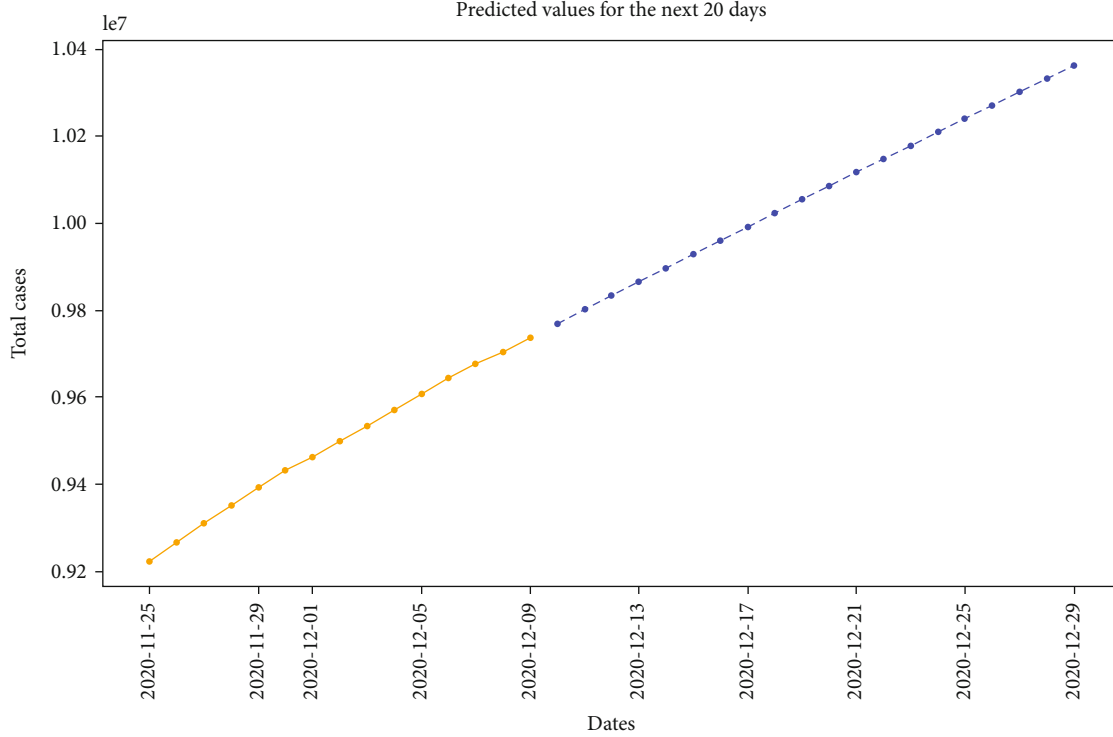


FIGURE 11: Prediction through ARIMA for the confirmed cases (for the next 20 days), where the x -axis is “dates” and the y -axis is “total cases.”

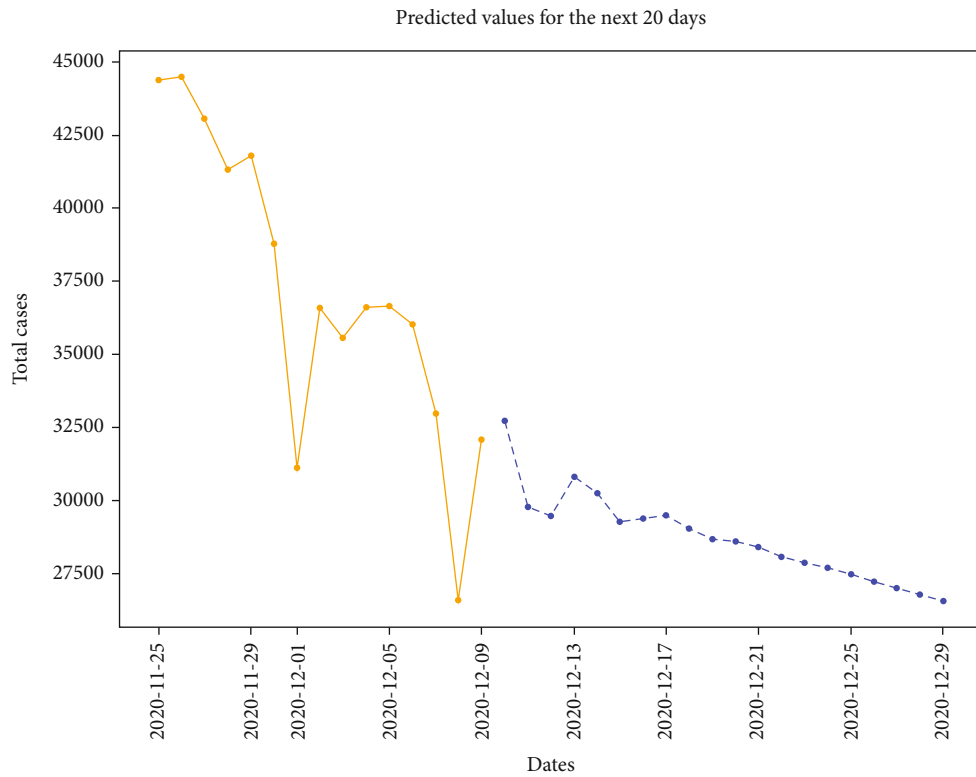


FIGURE 12: Prediction through ARIMA for the new cases (for the next 20 days), where the x -axis is “dates” and the y -axis is “total cases.”

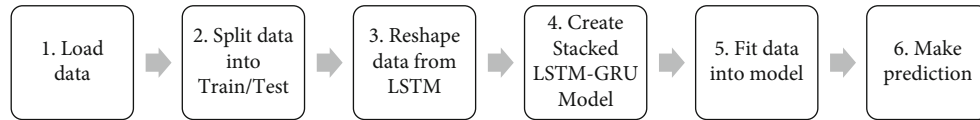
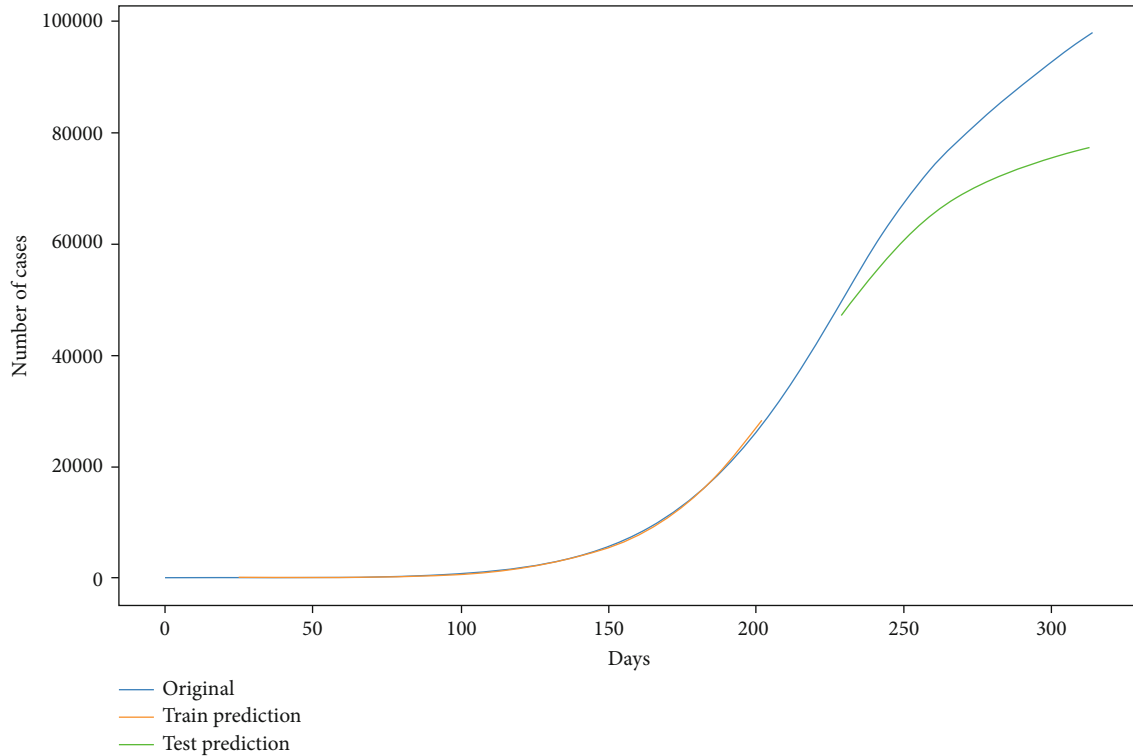


FIGURE 13: Model of stacked LSTM-GRU.

FIGURE 14: Prediction through stacked LSTM-GRU neural network for new cases (for the next 20 days), where the x -axis is “days” and the y -axis is “number of cases.”

- (5) The autocorrelation function (ACF) and partial autocorrelation function (PACF) graphs can be used to find correct values for p and q while tuning an ARIMA model. Tuning the “ n changepoints” parameter in FbProphet helps the model catch the trend in the training data. The large number of “ n changepoints” may result in model overfitting, lowering the model’s prediction accuracy on new data
- (6) The GRU model, which processes the time series data, uses the stacked LSTM output as an input. The GRU model’s generated output shows the output of the hybrid stacked LSTM and GRU model. Using stacked LSTM in a neural network to improve prediction accuracy while learning a large number of temporal features is a frequent practice. They also help to generate a better level of representation of the incoming sequence data over time
- (1) The COVID-19 pandemic has wreaked havoc on the construction industry, which is particularly vulnerable to economic cycles
- (2) The COVID-19 pandemic has raised attention to the already overburdened and understaffed home and institutional-based care systems in various nations. This short highlights the issues of recruiting, deploying, retaining, and protecting a sufficient number of well-trained and motivated care workers
- (3) The COVID-19 epidemic is wreaking havoc on public health and wreaking havoc on the economy and labor market, particularly for forest workers and enterprises. It has exacerbated existing issues, and as a result, many businesses and workers have suffered
- (4) As individuals can freely move across jurisdictions and countries, the road transportation business is vital to social and economic development. In an attempt to curb the spread of COVID-19, numerous nations worldwide have implemented domestic transit restrictions and/or closed border crossings for road freight transportation services. However, if the crisis is to be adequately handled, governments,

6. Challenges Faced due to COVID-19 in India

There are several challenges faced due to the impact of COVID-19 in overall India. Few are given below [64]:

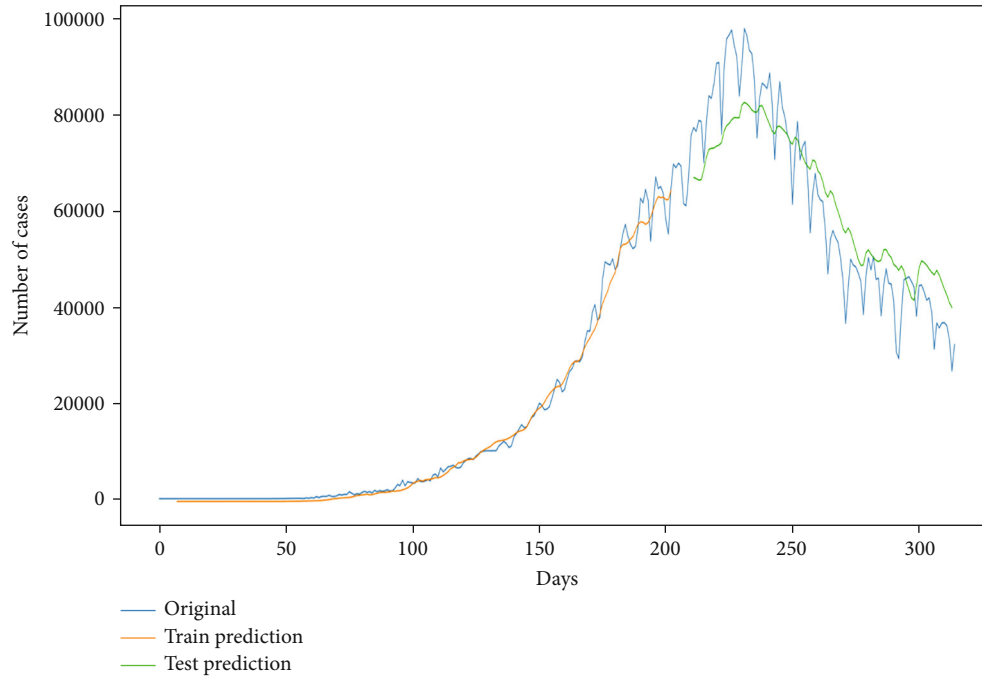


FIGURE 15: Prediction curve for confirmed cases, where the x -axis shows “days” and y -axis shows the “number of cases.”

TABLE 3: Statistical values for forecasted COVID-19 new confirmed cases in India.

Date	Forecasted COVID-19 new confirmed cases
11/12/2020	29,340
12/12/2020	26,819
13/12/2020	31,100
14/12/2020	27,546
15/12/2020	22,051
16/12/2020	26,111
17/12/2020	24,000
18/12/2020	22,700
19/12/2020	21,354
20/12/2020	29,456
21/12/2020	34,567
22/12/2020	23,965
23/12/2020	24,700
24/12/2020	23,765
25/12/2020	24,567
26/12/2020	27,912
27/12/2020	18,722
28/12/2020	16,400
29/12/2020	19,500
30/12/2020	21,800
31/12/2020	21,121

social partners, and parties involved in the road transport supply chain—such as shippers, receivers, transport purchasers, and middlemen—must act

rapidly to solve the issue of decent work for these critical individuals

- (5) Due to a combination of aircraft cancellations and restrictions aimed at preventing the spread of COVID-19, international travel has practically ground to a halt
- (6) Tourism is a significant source of job creation and economic development. COVID-19, on the other hand, has made a considerable difference. The impact on tourism businesses and workers, the majority of whom are young women, is unprecedented
- (7) As the disease develops, food delivery networks must remain operational to avoid a food crisis and reduce the worldwide economic impact
- (8) Even though many teachers in developing countries lack the essential skills and equipment to conduct successful remote education, most instructors and their organizations have accepted this challenge

7. Results and Conclusion

As a result, we investigated various forecasting models critical to illness prediction and demonstrated the expansion of COVID-19 cases in India. The prediction values from the hybrid model are more fit than the state-of-art models, as evidenced by the above data and line plots. These forecasted numbers are for the next few days. When the predictive values rise, the number of new cases will increase as well. Humans would have to battle the coronavirus in the future and obey the Government of India’s guidelines. Our models are capable of forecasting instances for the following 20 days

TABLE 4: Performance metrics for various models showing comparison with the state-of-the-art models.

Performance metrics	RNN	GRU	LSTM	Linear regression	Polynomial regression	ARIMA	Prophet	LSTM-GRU
R square	0.30	0.74	0.05	0.01	0.31	0.56	0.46	0.74
RMSE	120.35	94.558	134.505	284809.4	149117.8	1260	568.58	69.92

TABLE 5: Evaluation metrics.

R^2	$R^2 = 1 - \frac{\sum(Y_i - X_i)^2}{\sum(Y_i - Z_i)^2}$	Where Y_i is the actual value at certain i^{th} assumption, X_i is the forecasted value at i^{th} assumption, Z_i shows the average of the overall observations, and n is the total count of observations
RMSE	$RMSE = \sqrt{\frac{\sum(Y_i - X_i)^2}{n}}$	Where Y_i shows the forecasted value, X_i shows the true value, and n is the count of observations [65]

and even farther ahead, as shown in Table 3. Table 3 shows the statistical values for forecasted COVID-19 new confirmed cases in India.

Finally, the graphs of these models were compared, and the findings revealed that the stacked LSTM-GRU prediction was superior to the state-of-the-art models, as shown in Table 4. Hence, the characteristics of good predicting models should fit the past data value and be simple but effective. The state-of-the-art model taken is linear regression, polynomial regression, LSTM, GRU, RNN, ARIMA, and Prophet. Table 4 shows the performance metrics for various models showing comparison with the state-of-the-art models. Table 5 indicates the evaluation metrics.

As a result, as shown in Table 4, we investigated several models such as the recurrent neural network (RNN), gated recurrent unit (GRU), long short-term memory (LSTM), linear regression, polynomial regression, autoregressive integrated moving average (ARIMA), prophet, and hybrid stacking LSTM-GRU. The Root Mean Square Error (RMSE) is a quadratic scoring technique for calculating the mean magnitude of an error. To put it another way, the difference between the predicted and actual values is squared before being averaged over the sample value. Finally, the square root value of the average is determined. A lower RMSE value indicates that the model performs better [66]. It indicates how effectively the regression line predicts real-world values by the degree to which a data point fits the linear regression. You may calculate the amount of variance presented by the model's self-standing variables using the coefficient of determination. It determines the model's goodness-of-fit and assigns a score between 0 and 1 to it [67].

8. Future Works

The majority of current instances in India are asymptomatic, indicating that the person is infected but does not show any symptoms of coronavirus infection. The disease's name is COVID-19, and it is a terrible one. The virus must be well understood by the public. In proportion to our population, India's new occurrence figures are not as alarming at the time. This is only possible through a series of lockdowns. The information gathered in this study was compared to

confirmed daily COVID-19 cases in India. According to the data, the daily number of cases in India is likely to approach 1,500,000 instances per day in the next four weeks. Based on the dataset, this prediction was created in light of the current circumstances. While forecasting is not flawless, it can undoubtedly be used as a preventative measure. Future work can be improved by combining new components and algorithms with the hybrid model to generate more accurate findings.

Data Availability

We can send the datasets at the request of the authors.

Conflicts of 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] T. P. Velavan and C. G. Meyer, "The COVID-19 epidemic," *Tropical Medicine & International Health*, vol. 25, no. 3, pp. 278–280, 2020.
- [2] M. Barstugan, U. Ozkaya, and S. Ozturk, "Coronavirus (COVID-19) classification using C.T. images by machine learning methods," 2020, <http://arxiv.org/abs/2003.09424>.
- [3] "US Centers for Disease Control and Prevention," May 2020, <https://www.cdc.gov/coronavirus/2019-ncov/downloads/COVID19-symptoms-11x17-en.pdf>.
- [4] P. S. G. Unnithan, "Kerala reports first confirmed coronavirus case in India," January 2020, <https://www.indiatoday.in/india/story/kerala-reports-first-confirmed-novel-coronavirus-case-in-india-1641593-2020-01-30>.
- [5] T. Dehesh, H. A. Mardani-Fard, and P. Dehesh, "Forecasting of COVID-19 confirmed cases in different countries with ARIMA models," *Medrxiv*, vol. 3, no. 13, article 20035345, 2020.
- [6] V. Tulshyan, D. Sharma, and M. Mittal, "An eye on the future of COVID-19: prediction of likely positive cases and fatality in India over a 30-day horizon using the Prophet model," *Disaster Medicine and Public Health Preparedness*, pp. 1–7, 2020.

- [7] P. Mishra, A. Mohammad, G. Al Khatib et al., "Modelling and forecasting of COVID-19 in India," *Journal of Infectious Diseases and Epidemiology*, vol. 6, no. 5, 2020.
- [8] D. Benvenuto, M. Giovanetti, L. Vassallo, S. Angeletti, and M. Ciccozzi, "Application of the ARIMA model on the COVID-2019 epidemic dataset," *Data in Brief*, vol. 29, p. 105340, 2020.
- [9] D. Benvenuto, M. Giovanetti, L. Vassallo, S. Angeletti, and M. Ciccozzi, "Application of the ARIMA model on the COVID-2019 epidemic dataset," *Data in Brief*, vol. 29, article 105340, 2020.
- [10] L. Li, Z. Yang, Z. Dang et al., "Propagation analysis and prediction of the COVID-19," *Infectious Disease Modelling*, vol. 5, pp. 282–292, 2020.
- [11] R. Vaishya, M. Javaid, I. H. Khan, and A. Haleem, "Artificial intelligence (AI) applications for COVID-19 pandemic," *Diabetes and Metabolic Syndrome: Clinical Research & Reviews*, vol. 14, no. 4, pp. 337–339, 2020.
- [12] O.-D. Ilie, R.-O. Cojocariu, A. Ciobica, S.-I. Timofte, I. Mavroudis, and B. Doroftei, "Forecasting the spreading of COVID-19 across nine countries from Europe, Asia, and the American continents using the ARIMA models," *In Microorganisms*, vol. 8, no. 8, p. 1158, 2020.
- [13] G. Pandey, P. Chaudhary, R. Gupta, and S. Pal, "SEIR and regression model based COVID-19 outbreak predictions in India," <http://arxiv.org/abs/2004.00958>.
- [14] S. Shastri, K. Singh, S. Kumar, P. Kour, and V. Mansotra, "Time series forecasting of Covid-19 using deep learning models: India-USA comparative case study," *Chaos, Solitons and Fractals*, vol. 140, article 110227, 2020.
- [15] M. K. Siddiqui, R. Morales-Menendez, P. K. Gupta et al., "Correlation between temperature and COVID-19 (suspected, confirmed and death) cases based on machine learning analysis," *Journal of Pure and Applied Microbiology*, vol. 14, Supplement 1, pp. 1017–1024, 2020.
- [16] M. A. A. Al-Qaness, A. A. Ewees, H. Fan, and M. A. El Aziz, "Optimization method for forecasting confirmed cases of COVID-19 in China," *Journal of Clinical Medicine*, vol. 9, no. 3, 2020.
- [17] M. A. Shaik and D. Verma, "Deep learning time series to forecast COVID-19 active cases in India: a comparative study," *IOP Conference Series: Materials Science and Engineering*, vol. 981, 2020.
- [18] S. K. Bandyopadhyay and S. Dutta, "Machine learning approach for confirmation of COVID-19 cases: positive, negative death and release," *Medrxiv*, vol. 3, no. 25, 2020.
- [19] Z. Hu, Q. Ge, S. Li, L. Jin, and M. Xiong, "Artificial intelligence forecasting of Covid-19 in China," <http://arxiv.org/abs/2002.07112>.
- [20] T. T. Nguyen, "Artificial Intelligence in the battle against coronavirus (COVID-19): a survey and future research directions," <https://arxiv.org/abs/2008.07343>.
- [21] E. Dong, H. Du, and L. Gardner, "An interactive web-based dashboard to track COVID-19 in real time," *The Lancet Infectious Diseases*, vol. 20, no. 5, pp. 533–534, 2020.
- [22] C. R. Wells, P. Sah, S. M. Moghadas et al., "Impact of international travel and border control measures on the global spread of the novel 2019 coronavirus outbreak," *Proceedings of the National Academy of Sciences*, vol. 117, no. 13, pp. 7504–7509, 2020.
- [23] F. Villanustre, A. Chala, R. Dev, X. Lili, B. Furht, and T. Khoshgoftaar, "Modeling and tracking Covid-19 cases using big data analytics on HPC system platform," *Journal of Big Data*, vol. 8, no. 1, pp. 1–24, 2021.
- [24] A. Taneja, S. Rani, A. Alhudhaif, D. Koundal, and E. S. Gündüz, "An optimized scheme for energy efficient wireless communication via intelligent reflecting surfaces," *Expert Systems with Applications*, vol. 190, article 116106, 2022.
- [25] S. Bhat and D. Koundal, "Multi-focus image fusion using neurosophic based wavelet transform," *Applied Soft Computing*, vol. 106, article 107307, 2021.
- [26] R. Mehta, K. Aggarwal, D. Koundal, A. Alhudhaif, and K. Polat, "Markov features based DTCWS algorithm for online image forgery detection using ensemble classifier in the pandemic," *Expert Systems with Applications*, vol. 185, article 115630, 2021.
- [27] D. Koundal, B. Sharma, and Y. Guo, "Intuitionistic based segmentation of thyroid nodules in ultrasound images," *Computers in Biology and Medicine*, vol. 121, article 103776, 2020.
- [28] D. Koundal, S. Gupta, and S. Singh, "Computer aided thyroid nodule detection system using medical ultrasound images," *Biomedical Signal Processing and Control*, vol. 40, pp. 117–130, 2018.
- [29] M. S. Bali, K. Gupta, D. Koundal, A. Zaguia, S. Mahajan, and A. K. Pandit, "Smart architectural framework for symmetrical data offloading in IoT," *Symmetry*, vol. 2021, p. 13, 2021.
- [30] R. Singla, N. Kaur, D. Koundal, S. A. Lashari, S. Bhatia, and M. K. Imam Rahmani, "Optimized energy efficient secure routing protocol for wireless body area network," *Access*, vol. 9, pp. 116745–116759, 2021.
- [31] F. Alenezi, "Image dehazing based on pixel guided CNN with PAM via graph cut," *CMC-Computers, Materials & Continua*, vol. 71, no. 2, pp. 3425–3443, 2022.
- [32] F. Alenezi, A. Armghan, S. N. Mohanty, R. H. Jhaveri, and P. Tiwari, "Block-greedy and CNN based underwater image dehazing for novel depth estimation and optimal ambient light," *Water*, vol. 13, no. 23, p. 3470, 2021.
- [33] G. P. Joshi, F. Alenezi, G. Thirumoorthy, A. K. Dutta, and J. You, "Ensemble of deep learning-based multimodal remote sensing image classification model on unmanned aerial vehicle networks," *Mathematics*, vol. 9, no. 22, p. 2984, 2021.
- [34] F. Alenezi and K. C. Santosh, "Geometric regularized Hopfield neural network for medical image enhancement," *International Journal of Biomedical Imaging*, vol. 2021, Article ID 6664569, 12 pages, 2021.
- [35] F. Alenezi and E. Salari, "A fuzzy-based medical image fusion using a combination of maximum selection and Gabor filters," *International Journal of Engineering Science*, vol. 9, pp. 118–129, 2018.
- [36] F. S. Alenezi and S. Ganesan, "Geometric-pixel guided single-pass convolution neural network with graph cut for image dehazing," *IEEE Access*, vol. 9, pp. 29380–29391, 2021.
- [37] S. Majid, F. Alenezi, S. Masood, M. Ahmad, E. S. Gündüz, and K. Polat, "Attention based CNN model for fire detection and localization in real-world images," *Expert Systems with Applications*, vol. 189, article 116114, 2022.
- [38] N. Daldal, M. Nour, and K. Polat, "A novel demodulation structure for quadrature modulation signals using the segmentary neural network modelling," *Applied Acoustics*, vol. 164, p. 107251, 2020.
- [39] N. Daldal, A. Sengur, K. Polat, and Z. Cömert, "A novel demodulation system for base band digital modulation signals

- based on the deep long short-term memory model,” *Applied Acoustics*, vol. 166, article 107346, 2020.
- [40] N. Daldal, Z. Cömert, and K. Polat, “Automatic determination of digital modulation types with different noises using convolutional neural network based on time–frequency information,” *Applied Soft Computing*, vol. 86, article 105834, 2020.
- [41] M. Nour, N. Daldal, M. F. Kahraman, H. Sindi, A. Alhudhaif, and K. Polat, “A novel tilt and acceleration measurement system based on hall-effect sensors using neural networks,” *Mathematical Problems in Engineering*, vol. 2022, Article ID 7000486, 13 pages, 2022.
- [42] M. F. Kahraman and S. Öztürk, “Experimental study of newly structural design grinding wheel considering response surface optimization and Monte Carlo simulation,” *Measurement*, vol. 147, article 106825, 2019.
- [43] M. Indhuja and P. P. Sindhuja, “Prediction of covid-19 cases in India using prophet,” *International Journal of Statistics and Applied Mathematics*, vol. 5, no. 4, pp. 103–106, 2020.
- [44] V. Kulshreshtha and N. K. Garg, “Predicting the new cases of coronavirus [COVID-19] in India by using time series analysis as machine learning model in Python,” *Journal of The Institution of Engineers (India): Series B*, vol. 102, pp. 1–7, 2021.
- [45] J. Sean and B. Letham, “Prophet: forecasting at scale,” February 2017, <https://research.fb.com/blog/2017/02/prophet-forecasting-at-scale/>.
- [46] Facebook’s Core, “Prophet forecasting at scale, data science team,” July 2020, <https://facebook.github.io/prophet/>.
- [47] K. Hariharan, “Time series forecasting — ARIMA vs Prophet medium,” <https://medium.com/analytics-vidhya/time-series-forecasting-arima-vs-prophet-5015928e402a>.
- [48] D. Belikov, M. Arshinov, B. Belan et al., “Analysis of the diurnal, weekly, and seasonal cycles and annual trends in atmospheric CO₂ and CH₄ at tower network in Siberia from 2005 to 2016,” *Atmosphere*, vol. 10, no. 11, p. 689, 2019.
- [49] R. Fu, Z. Zhang, and L. Li, “Using LSTM and GRU neural network methods for traffic flow prediction,” in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 324–328, Wuhan, China, 2016.
- [50] S. Prabhakaran, “Arima model—complete guide to time series forecasting in python,” 2019, <https://machinelearningplus.com/>.
- [51] T. Vincent, “A guide to time series forecasting with ARIMA in Python 3,” <https://www.digitalocean.com/community/tutorials/a-guide-to-time-series-forecasting-with-arima-in-python-3>.
- [52] C. Shetty, “Time series models,” September 2020, <https://towardsdatascience.com/time-series-models-d9266f8ac7b0>.
- [53] S. Mehrmolaei and M. R. Keyvanpour, “Time series forecasting using improved ARIMA,” *2016 Artificial Intelligence and Robotics (IRANOPEN)*, vol. 2016, pp. 92–97, 2016.
- [54] Sangarshanan, “Time series forecasting-ARIMA models,” July 2020, <https://towardsdatascience.com/time-series-forecasting-arima-models-7f221e9eee06>.
- [55] Wikipedia, “Long short-term memory,” July 2020, https://en.wikipedia.org/wiki/Long_short-term_memory.
- [56] S. Li, “Time series analysis, visualization & forecasting with LSTM,” May 2019, <https://towardsdatascience.com/time-series-analysis-visualization-forecasting-with-lstm-77a905180eba>.
- [57] Y. Duan, Y. Lv, and F. Y. Wang, “Travel time prediction with LSTM neural network,” in *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*, pp. 1053–1058, Rio de Janeiro, Brazil, 2016.
- [58] F. Shahid, A. Zameer, and M. Muneeb, “Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM,” *Chaos, Solitons & Fractals*, vol. 140, p. 110212, 2020.
- [59] A. U. Muhammad, A. S. Yahaya, S. M. Kamal, J. M. Adam, W. I. Muhammad, and A. Elsafi, “A hybrid deep stacked LSTM and GRU for water price prediction,” in *2020 2nd International Conference on Computer and Information Sciences (ICCS)*, pp. 1–6, Sakaka, Saudi Arabia, 2020.
- [60] V. Bushaev, “Adam — latest trends in deep learning optimization. Medium,” October 2018, <https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c>.
- [61] J. Brownlee, *Understand the impact of learning rate on neural network performance*, Machine Learning Mastery, 2019.
- [62] “DeepAI. ReLu,” June 2020, <https://deeptai.org/machine-learning-glossary-and-terms/relu>.
- [63] A. Choudhary, “Generate quick and accurate time series forecasts using Facebook’s Prophet (with Python & R codes),” May 2018, <https://www.analyticsvidhya.com/blog/2018/05/generate-accurate-forecasts-facebook-prophet-python-r/>.
- [64] International Labour Organization, “ILO Monitor: COVID-19 and the World of Work,” in *Updated Estimates and Analysis. Int Labour Organ*, International Labour Organization, 2020.
- [65] J. Moody, “What does RMSE really mean,” September 2019, <https://towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e>.
- [66] “MAE and RMSE-which metrics is better?,” 2016, <https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d>.
- [67] O. Renaud and M.-P. Victoria-Feser, “A robust coefficient of determination for regression,” *Journal of Statistical Planning and Inference*, vol. 140, no. 7, pp. 1852–1862, 2010.
- [68] W. Hastomo, A. S. B. Karno, N. Kalbuana, and A. Meiriki, “Characteristic parameters of epoch deep learning to predict Covid-19 data in Indonesia,” *Journal of Physics: Conference Series*, vol. 1933, no. 1, article 012050, 2021.
- [69] A. Hernandez-Matamoros, H. Fujita, T. Hayashi, and H. Perez-Meana, “Forecasting of COVID19 per regions using ARIMA models and polynomial functions,” *Applied Soft Computing*, vol. 96, article 106610, 2020.
- [70] F. Khennou and M. A. Akhloufi, “Deep forecasting of COVID-19: Canadian case study,” in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 309–315, Springer, Cham, 2021.
- [71] S. Tuli, S. Tuli, R. Tuli, and S. S. Gill, “Predicting the growth and trend of COVID-19 pandemic using machine learning and cloud computing,” *Internet of Things*, vol. 11, article 100222, 2020.
- [72] “Univariate arima models,” July 2020, http://www.fhi.sk/files/katedry/kove/predmety/Prognosticke_modely/ARIMA.pdf.