https://doi.org/10.1007/s42979-022-01493-3

ORIGINAL RESEARCH

Machine Learning and Deep Learning Based Time Series Prediction and Forecasting of Ten Nations' COVID-19 Pandemic

Yogesh Kumar[1] · Apeksha Koul[2] · Sukhpreet Kaur[3] · Yu-Chen Hu[4]

Received: 29 July 2022 / Accepted: 3 November 2022 / Published online: 14 December 2022

Abstract

In the paper, the authors investigated and predicted the future environmental circumstances of a COVID-19 to minimize its effects using artificial intelligence techniques. The experimental investigation of COVID-19 instances has been performed in ten countries, including India, the United States, Russia, Argentina, Brazil, Colombia, Italy, Turkey, Germany, and France using machine learning, deep learning, and time series models. The confirmed, deceased, and recovered datasets from January 22, 2020, to May 29, 2021, of Novel COVID-19 cases were considered from the Kaggle COVID dataset repository. The country-wise Exploratory Data Analysis visually represents the active, recovered, closed, and death cases from March 2020 to May 2021. The data are pre-processed and scaled using a MinMax scaler to extract and normalize the features to obtain an accurate prediction rate. The proposed methodology employs Random Forest Regressor, Decision Tree Regressor, K Nearest Regressor, Lasso Regression, Linear Regression, Bayesian Regression, Theilsen Regression, Kernel Ridge Regressor, RANSAC Regressor, XG Boost, Elastic Net Regressor, Facebook Prophet Model, Holt Model, Stacked Long Short-Term Memory, and Stacked Gated Recurrent Units to predict active COVID-19 confirmed, death, and recovered cases. Out of different machine learning, deep learning, and time series models, Random Forest Regressor, Facebook Prophet, and Stacked LSTM outperformed to predict the best results for COVID-19 instances with the lowest root-mean-square and highest R^2 score values.

Keywords COVID-19 · Prediction · XG Boost · Facebook Prophet · Holt model · Stacked gated recurrent units · RANSAC regressor · Random forest regressor · Stacked long short-term memory

Introduction

Throughout history, the world has confronted several major pandemic and epidemic problems. The first recorded pandemic occurred in Athens during the Peloponnesian War in 430 BC, followed by the Antonine Plague in 165 A.D., in 250 A.D.—the Cyprian Plague, in 541 A.D.—the Justinian Plague, in the eleventh century—leprosy, in 1350—The Black Death, in 1492—The Columbian Exchange, in 1665—The Great Plague of London, in 1817—The First Cholera Pandemic, in 1855—The Third Plague Pandemic, in 1875—Fiji Measles Pandemic, in 1889—Russian Flu, in 1918—Spanish Flu, in 1957—Asian Flu, in 1981—HIV/AIDS, in 2003-SARS, and 2019—COVID-19 [1]. While still a public health concern, Coronavirus 19 (also known as COVID-19) is an infectious sickness that occurred by the severe acute respiratory syndrome coronavirus 2. The first recorded case of SARS (severe acute respiratory syndrome) was identified in December of 2019 in Wuhan, China. The disease has

✉ Yogesh Kumar
yogesh.arora10744@gmail.com;
yogesh.kumar@sot.pdpu.ac.in

Apeksha Koul
apekshakoulo9@gmail.com

Sukhpreet Kaur
er.sukhpreetkaur@gmail.com

Yu-Chen Hu
ychu@pu.edu.tw

[1] Department of Computer Science and Engineering, School of Technology, Pandit Deendayal Energy University, Gandhinagar, Gujarat, India

[2] Department of Computer Engineering, Punjabi University, Patiala, India

[3] Department of Computer Science and Engineering, Chandigarh Engineering College, Landran, Mohali, India

[4] Department of Computer Science and Information Management, Providence University, Taichung, Taiwan, ROC

since spread to many other nations and healthcare systems worldwide. At the same time, humans inhale contaminated air, including airborne droplets and particles that are smaller than 0.1 microns, and COVID-19 spreads [2].

Inhalation of these particles is more dangerous when people are closely packed together; nevertheless, they can be inhaled further apart, especially indoors. Infected fluids sprayed on the skin, in the eyes, nose, or mouth, or on surfaces contaminated with them may result in transmission. Someone can carry and spread the disease for up to 20 days even if they have no symptoms. During COVID-19, a first wave began in the spring, which receded significantly throughout the summer, and a second wave appeared in the fall of 2020. The initial wave of the epidemic devastated several nations, and many patients perished. The severity of this early phase was exacerbated by a lack of specialist equipment and a lack of understanding of the disease [4]. We all learned from our mistakes during the first wave of the pandemic, and as a result, our confidence in being able to handle the second wave much better was strong. Despite this, the second wave had considerably greater infection rates, more patients in ICUs, and, in certain countries, more fatalities [5].

Figure 1 depicts the death rates from March 6, 2020, to June 6, 2021, with Europe and the Americas having the most significant mortality rates compared to India and South and East Asia. Europe had 1,172,912 death cases, the Americas had 1,926,520, South and East Asia had 739,802 death cases, and India had 402,728 COVID death cases as of July 8, 2021. Europe accounted for 32% of all COVID fatality cases, followed by the Americas (55%), South and East Asia (15%), and India (11%) (approx). According to the survey, the top eight countries that have been severely affected by a novel coronavirus (in billion dollars) are the United States (3.39), India (3.09), Brazil (1.92), France (58.2), Russia (57.6), Turkey (54.9), the United Kingdom (51.9), Argentina (46.8), and Colombia (45.3) [6].

In the beginning, no curative medication or vaccine was available for COVID-19, but 18 months later, each of the vaccines was shown to be safe and effective in treating COVID-19 symptoms and lab-confirmed cases. Though vaccinations are pretty successful, SARS-CoV-2, the virus that causes COVID-19, will emerge even in this tiny number of individuals. Many different approaches for diagnosing the illness have been developed. RT-PCR, TMA, and RT-LAMP can be used to identify the virus's nucleic acid. However, there are some situations when RT-PCR may not be an option, such as when viral RNA must be analyzed in a hurry. According to the UNICEF and World Health Organization, around 342 million vaccinations have been supplied to medical facilities, resulting in the immunization of approximately 94 million people worldwide. China had the most excellent vaccination rate, with 22.3 million.

In this study, machine learning, time series, and deep learning-based models are developed to predict future COVID's active verified, mortality, and healed cases of random 5 days, using January 22, 2020, to May 29, 2021, verified, mortality, and recovered instances of the top ten

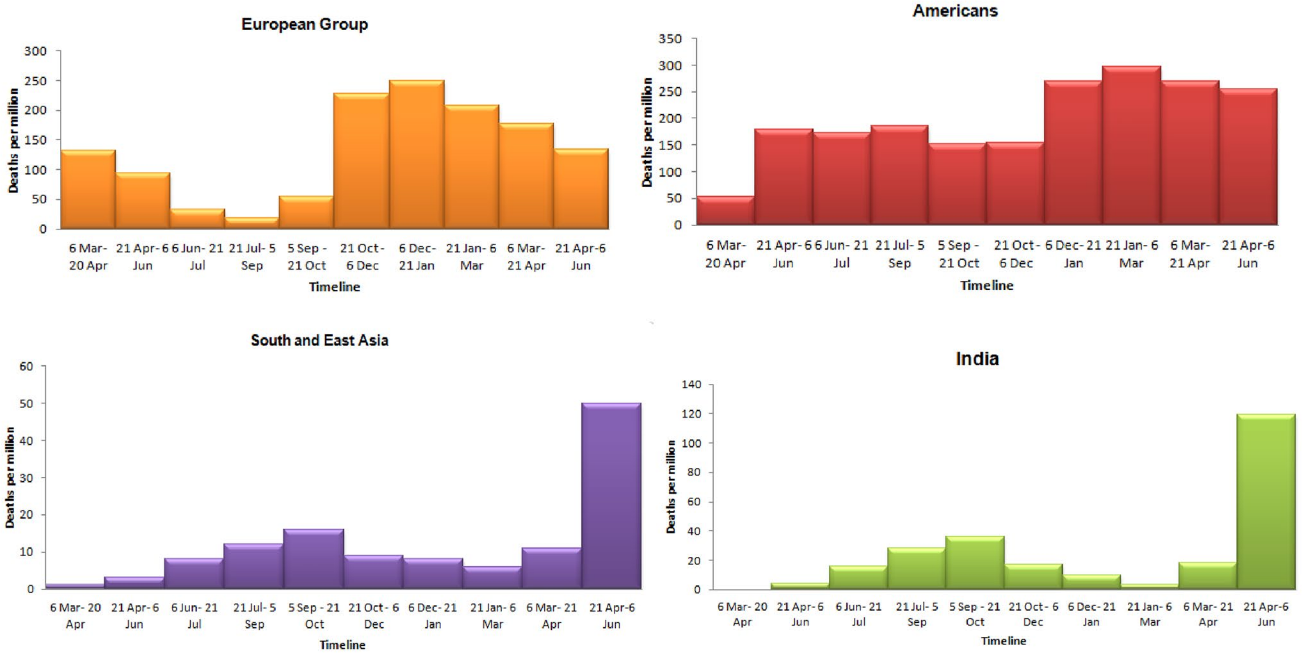


Fig. 1 Ravages of the pandemic

countries in the world, such as India, the United States, Russia, Argentina, Brazil, Colombia, Italy, Turkey, Germany, and France. We used random forest regressor, decision tree regressor, K-nearest neighbor regressor, kernel ridge regressor, X Boost, RANSAC regressor, linear regression, lasso regression, elastic net regressor, Bayesian regressor, and Theilsen regressor from machine learning algorithms; stacked LSTM and stacked GRU from deep learning models; and Facebook Prophet and Holt algorithms from time series model. To yet, in our investigation, we have been unable to locate any previous research case studies about the top 10 nations affected by the COVID-19 pandemic. Our participation in this research study would benefit all ten countries in rebuilding the plan and demography of COVID-19 preparation. The Root-Mean-Square Error (RMSE) and R^2 Score are the evaluative metrics used to assess these models. After this section, the rest of the paper is laid out as follows: The second section tells about related work. The section "Contribution Outline" presents the article's contribution in outline form. The section "Materials and Methods" focus on the subject matter and methods and are followed by a discussion of the outcomes. The section "Result Analysis" draws the conclusion and winds down the recommended research.

Related Work

Since 2020, researchers have made significant attempts to anticipate the onset of COVID illness in people or the end of the disease around the globe. Keeping this in mind, Shastri et al. [1] suggested a deep learning-based model, such as a recurrent neural network, to forecast the future circumstances of new coronaviruses by studying instances from India and the United States. Ten different nations with the most significant number of verified cases were investigated. It was shown that the predictive accuracy of a range of six separate time series modeling approaches for coronavirus epidemic detection varied by Papastefanopoulos et al. [2]. Using an LSTM model, Chimmula et al. [3] predicted the end of the COVID-19 pandemic and worldwide epidemics due to antiviral drugs and improved access to healthcare. Indicating the date of the pandemic's demise, the writers anticipate that it will be finished by June of 2020. Using a deep learning model, Togacar et al. [4] identified coronavirus in datasets containing instances of pneumonia, as well as standard X-ray imaging data. The COVID-19 disease can be diagnosed with 99.27% accuracy with the model that the authors used. COVID-19 drug and vaccine research achievements were evaluated using artificial intelligence techniques in a recent study by Arshadi et al. [5]. In addition, the scientists gave information about the compounds, peptides, and epitopes in the CoronaDB-AI library, which were discovered both in silico and in vitro.

Categorizing chest X-rays into two groups was proposed by the researchers led by Elaziz et al. [6]. The accuracy percentage for the first and second datasets was 96.09% and 98.09%, respectively. Alimadadi et al. [7] presented a deep learning algorithm based on AlphaFold to predict the structures related to COVID-19 illness. Alazab et al. [8] used real-world datasets to detect COVID-19 patients using artificial intelligence-based approaches on a deep convolution neural network. In Australia and Jordan, their methods obtained an accuracy of 94.80% and 88.43%, respectively. Alaska et al. [9] evaluated the efficacy of deep learning models in predicting COVID-19 illness using laboratory data from 600 patients and got 91.89% accuracy. Their approach was also utilized to help medical professionals validate test data and for clinical prediction research. The Johns Hopkins dashboard data, which were the primary source of the Punn et al.'s [10] research, were utilized with machine learning and deep learning models. The team's goal was to grasp the exponential growth of the COVID-19 and then make predictions about how widespread it may become across the country. Table 1 on the left shows the researchers who worked on the forecast and detection of COVID-19.

Contribution Outline

The overall goal of this research is to build models that can calculate two necessary evaluative measures: RMSE and R^2 Score for confirmed, death, and recovered cases from ten different nations to help future forecasts. The steps are as follows:

Step 1: Initially, data are pre-processed to capture characteristics utilizing various variables, such as active cases, recovered cases, and COVID-19 fatality cases.

Step 2: Exploratory Data Analysis of COVID-19's active cases, closed cases, confirmed cases, recovered cases, and death cases are calculated to summarize or interpret the information that is hidden in rows or columns, and scaling techniques such as Min–Max have been applied to normalize each feature that is obtained from these attributes.

Step 3: Later, utilizing confirmed cases, recovered cases, and death cases from 10 different nations, the gathered data were used to anticipate the future conditions of a new CoronaVirus. To get the findings, several machine learning models, time series models, and deep learning models were used, and they were assessed using parameters, such as root-mean-square error and R square.

Step 4: Finally, all of the results have been ranked to choose the technique with the lowest root-mean-square error and the highest R-squared score.

As depicted in Fig. 2, the proposed approach works by collecting and preparing a dataset from the Novel Corona Virus dataset.

Table 1 Analysis of the existing work

Author's name	Dataset	Technique	Results	Limitations
Wang et al. [11]	1065 CT pathogenic images	Transfer Learning Model, CNN, Graph-Net	Accuracy: 89.5% Specificity: 0.88 Sensitivity: 0.87	Factors such as low signal-to-noise ratio and complex data integration led to reducing the efficacy of deep learning models
Bandyopadhyay et al. [12]	Data collected from WHO (Jan. 16–20,2020)	Long Short-Term Memory, Gated Recurrent Unit	Accuracy: 87%	The model failed to represent the spatio temporal components of the LSTM network
Togacar et al. [4]	Data collected from Qatar University	Stacking Technique, Fuzzy Color, Deep Learning Model	Classification accuracy: 99.27%	Publications of COVID-19 images were limited. The system did not work with the low resolution and different size input images
Shastri et al. [1]	Dataset was sourced from the Ministry of Health and Family Welfare	Deep Neural Network, Long Short-Term Memory, Recurrent Neural Network, Polynomial Regression	Accuracy ConvLSTM: 98%	The comparative analysis had been performed only for two countries
Ghoshal et al. [13]	COVID-19 chest X-ray dataset	Bayesian Deep Learning	Accuracy: 80%	After reviewing the data, it was impossible to conclude anything regarding markers for imaging, discoveries concerning improved diagnosis and therapy for COVID-19
Punn et al. [10]	Data collected from Jan 22, 2020 to Apr 1 2020 at Johns Hopkins University	Support Vector Machine, Deep Neural Network, Long Short-Term Memory, Polynomial Regression	RMSE confirmed: 455.92 Death: 117.94 Recovered: 809.71	The study needed to work on more algorithms to enhance the RMSE score
Alakus et al. [9]	Samples collected from the Albert Einstein Israelite Hospital in Sao Paulo, Brazil	Artificial Neural Network, Convolution Neural Network, Long Short-Term Memory	Accuracy: 86.66% F1 Score: 91.89% Recall: 99.42% AUC: 62.50% Precision: 86.75%	The primary disadvantage of the study was the sheer amount of data. To increase the number of patients for whom the lab findings could not be assessed, the procedure was applied on 600 patients
Ismael et al. [14]	180 COVID-19 and 200 chest X-ray images	CNN model, SVM, ResNet50	Accuracy: 91.6%	The study needed to incorporate work on different imagistic patterns of COVID-19
Panwar et al. [15]	337 patient images from real-world data	Deep learning, nCOVnet	Accuracy: 97.62%	The system worked on a small dataset
Elaziz et al. [6]	Dataset collected from Joseph Paul Cohen and Paul Morrison Lan Dao	Manta Ray Foraging Optimization, Fractional Multichannel Exponent Moments	Accuracy: 96.09% Accuracy: 98.09%	The system dealt with resource limitations and high CPU time

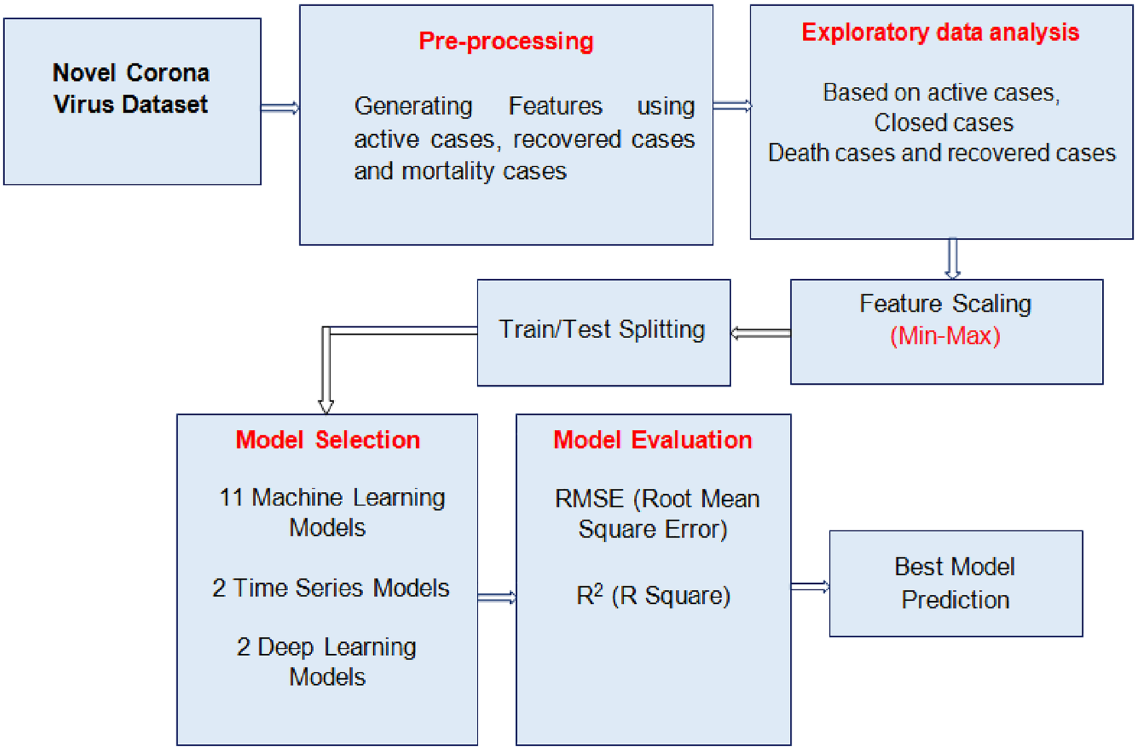

Novel Corona Virus Dataset
Pre-processing
Generating Features using active cases, recovered cases and mortality cases
Exploratory data analysis
Based on active cases, Closed cases Death cases and recovered cases
Feature Scaling (Min-Max)
Train/Test Splitting
Model Selection
11 Machine Learning Models
2 Time Series Models
2 Deep Learning Models
Model Evaluation
RMSE (Root Mean Square Error)
R2 (R Square)
Best Model Prediction

Fig. 2 Proposed system design for COVID-19 prediction

Materials and Methods

This section provides a general description of the dataset, along with libraries and methodologies imported during implementation.

Dataset

Coronavirus (2019-nCoV) is a virus (more specifically known as a coronavirus) discovered in Wuhan, China, and responsible for an influenza-like outbreak. One of China's earliest suspected sources of the COVID-19 epidemic was an extensive seafood and animal market, which indicated possible animal-to-human transmission.

However, an increasing number of cases are claimed to have occurred in the absence of contact with animal markets, suggesting that person-to-person transmission occurs. The CDC [16] is currently unaware of how fast or sustainably this virus spreads among humans. According to a report issued in Wuhan City, Hubei Province, China, on December 31, 2019, several instances of pneumonia have been discovered in the area. The virus has no similarity to any other virus currently known. This raised concerns, as we have no idea how a novel virus may affect humans. Everyday data on individuals with a disease can lead to intriguing results when released to the broader data science community [17].

This dataset is compiled daily to offer recent news on new coronavirus infections, fatalities, and recoveries for 2019. The data will be available from January 22, 2020 to May 29, 2021. This is a time series dataset with a total of 1248 time series datasets recorded for each day, while the count of time series datasets registered for each day indicates the cumulative total.

The dataset contains a serial number, the observation date in the format MM/DD/YYYY, the province or state of observation, the country or region of compliance, and the time in UTC at which the row is updated for the given province or country, the cumulative number of confirmed cases, the cumulative number of death cases, and the cumulative number of recovered patients from January 22, 2020 to May 29, 2021. The confirmed, dead, and recovered cases from ten different countries are included in Table 2.

Libraries

Several Python-based libraries, such as *Pandas*—a python-based software toolkit that contains data structures and strategies for working with numerical tables and *time series*—were imported during the prediction of COVID-19 confirmed, death, and recovered cases [18], and *Numpy*—a Python array manipulation library. It also contains functions for working with linear algebra, the Fourier transform, and matrices [19], among other things. *Matplotlib*—a

Table 2 Analysis of COVID-19 cases among the top ten countries

Countries	Confirmed cases	Death cases	Recovered cases
India	27,894,800	325,972	25,454,320
USA	33,251,939	594,306	–
Russia	4,995,613	118,781	46,16,422
Argentina	3,732,263	77,108	3,288,467
Brazil	16,471,600	461,057	14,496,224
Colombia	3,363,061	87,747	3,141,549
Italy	4,213,055	126,002	3,845,087
Turkey	33,251,939	47,271	5,094,279
Russia	4,995,613	118,781	4,616,422
Germany	3,684,672	88,413	3,479,700

cross-platform data visualization and graphical plotting program built-in Python for use with NumPy [19]; *Seaborn*—a python data visualization software based on Matplotlib. It uses a high-level interface to generate aesthetically beautiful and functional data visualizations [20], *Plotly* is a Python library that makes it easier to create professional-looking visualization by providing a flexible, open-source charting toolkit with over 40 chart types for a wide range of statistical, financial, geographic, scientific, and 3D use cases. [21], *Date–time* is a module that mixes date and time and characteristics like the year, month, day, hour, minute, second, microsecond, and info [22]. *Sklearn*—Scikit-learn is the most helpful Python machine learning package. The sklearn package contains several rapid machine learning and statistical modeling algorithms, including classification, regression, clustering, and dimension reduction [23]. Fbprophet utilizes time as a regressor and attempts to fit multiple linear/nonlinear time functions as components. FbProphet will provide the data using a linear model by default, but it may be modified to a nonlinear model (logistics growth) using its parameters [24]; *XGBoost* is an implementation of Gradient boosted Decision Trees (GDTs) designed for both high-performance and domination [25]; *Tensor Flow*—Tensor Flow is an open-source framework that processes datasets arranged as computational graph nodes. *Keras* is a Python-based open-source software framework that provides an artificial neural network interface. Keras is a user interface for the Tensor Flow library [26]. *StatsModel* is a Python package that includes classes and methods for estimating various statistical models, running statistical tests, and exploring statistical data [27]. *PmdarimaMath* is a statistical library created to cover a gap in Python's time series capabilities. CatBoost is an open-source package that offers a high-performance gradient boosting algorithm for decision trees. The search engine use is extensive. It is used in recommendation systems, personal assistants, self-driving vehicles, weather forecasting, and a wide variety of other applications [28].

Techniques

The pre-processing approach used to extract the characteristics is covered in the part of this work. This part also discusses the exploratory data analysis of the cleaned data, which is followed by the scaling approach. Following that, a section was presented in which many models from the COVID-19 testing dataset were described and shown.

Pre-processing

Data collected from the novel corona 19 dataset have been pre-processed using various mathematical formulas, such as active cases, percentage of recovery rate, percentage of mortality, and week of days to generate features. There is a significant likelihood that the number of active topics has increased, since some of the confirmed patients are now dead, and fewer new cases are being found. To calculate it, use Eq. (1). The recovery rate is the proportion of recovered instances, while the mortality rate is the percentage of death cases. Equations (2), (3) display the formulas. The last parameter, the week of days, is calculated by importing the library named WEEKOFYEAR [24]

$$\text{Active cases} = \text{Total number of confirmed cases} - (\text{total number of recovered cases} + \text{total number of death cases}, \tag{1}$$

$$\text{Recovery rate} = \frac{\text{Number of recovered cases}}{\text{number of confirmed cases}} \times 100, \tag{2}$$

$$\text{Mortality rate} = \frac{\text{Number of death cases}}{\text{Number of confirmed cases}} \times 100. \tag{3}$$

Exploratory Data Analysis

Exploratory Data Analysis is a vital process that entails performing preliminary analyses on data to uncover patterns, identify anomalies, test hypotheses, and verify assumptions using summary statistics and graphical representations. Some of the critical steps in exploratory data analysis are importing the data set in which we will get two data frames; one consisting of the data to be trained and the other for predicting the target value, identifying the number of features and columns, identifying the qualities or cues, identifying the data types of components, identifying the number of observations, checking if the dataset has empty cells or samples, identifying the number of empty cells by features or columns, and exploring categorical features [29].

This work employed an exploratory analysis of ten different countries after pre-processing to assess its features via statistical graphs. Figures shown below depicts the graphical analysis of active cases, death cases, closed points, and recovered cases that have been recorded from Jan 2020 to May 2021.

It was determined in Fig. 3 that 27,894,800 cases had been confirmed, 2,114,508 were still active, 325,972 had died, 25,780,292 had been closed, and 25,454,320 people had been recovered from Jan 2020 to 29 May 2021. Additionally, the numbers of confirmed cases, deaths, and recovered cases each day were, respectively, 57,397, 671, and 52,375.

According to Fig. 4, it has been discovered that US has 3,325,189,940 instances with high certainty, 3,266,576,333 cases with moderate certainty, 594,306 cases with low certainty, and 0 cases with a medium certainty which were seen from January 1st, 2020 to May 29th, 2021. Additionally, the daily average of confirmed cases was reported as 673,128, while the daily average of deaths was recorded as 12,030. Finally, the daily average of recovered cases was recorded as 0.

As demonstrated in Fig. 5, the numbers of confirmed, active, and death cases have been as follows: 49,956,313.0, 260,410.0, 118,781.0, 47,352,203.0, and 46,164,322.0 from January 1, 2020 to May 29, 2021. Finally, the total number of confirmed cases was 10,300. The number of death cases was 245, and the total number of recovered cases was 9518.

In Fig. 6, it was discovered that Argentina has reported 373,263.0 total cases, with 366,688.0 currently active cases, 77,108 currently known death cases, 336,575 previously known to be closed cases, and 328,467 previously known recovered cases from January 1st, 2020 to May 31st, 2021. In addition to this, there were around 8239.0 confirmed cases of the disease each day, approximately 170.0 deaths per day, and approximately 7259.0 recovered cases per day.

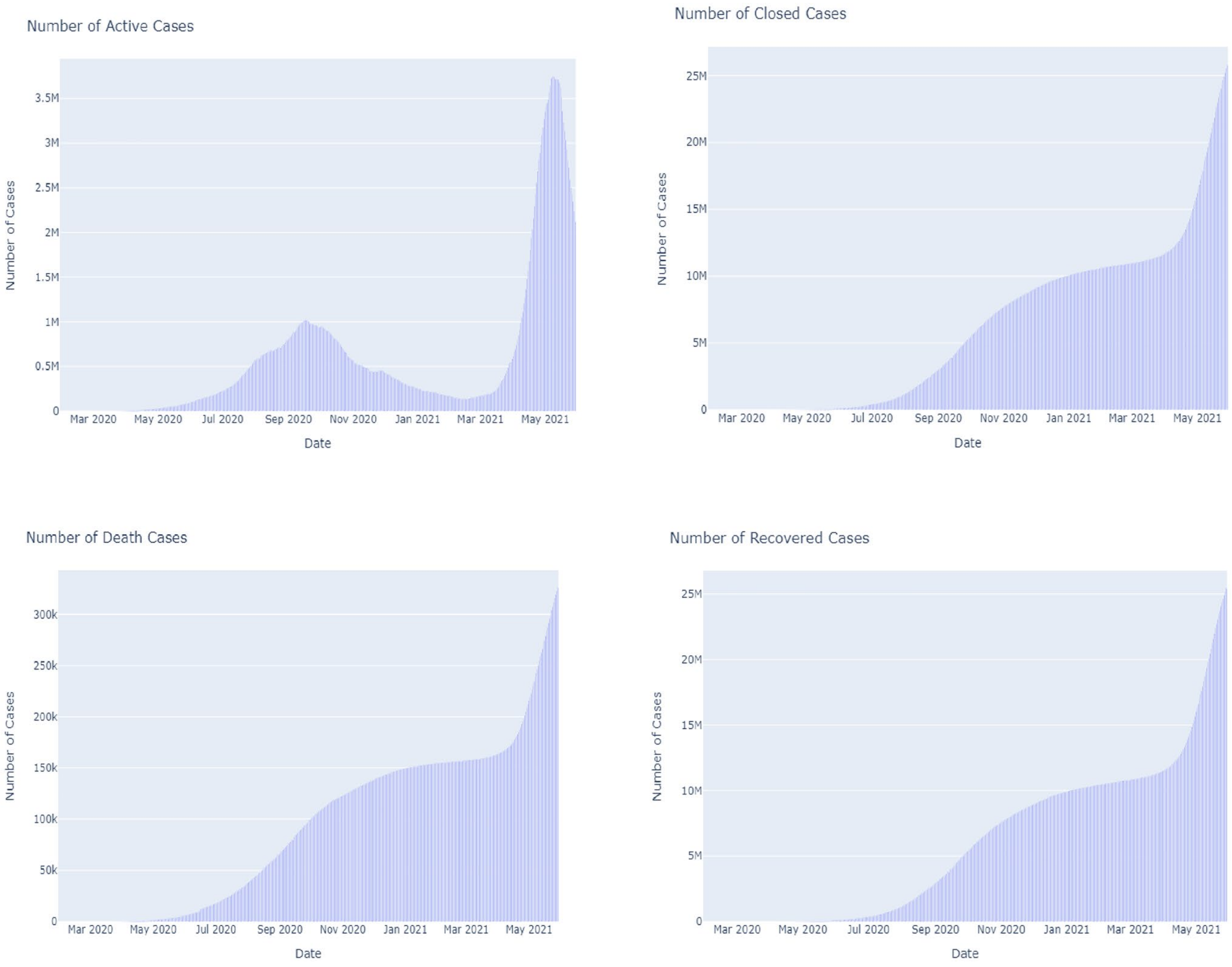


Fig. 3 India’s COVID-19 scenario

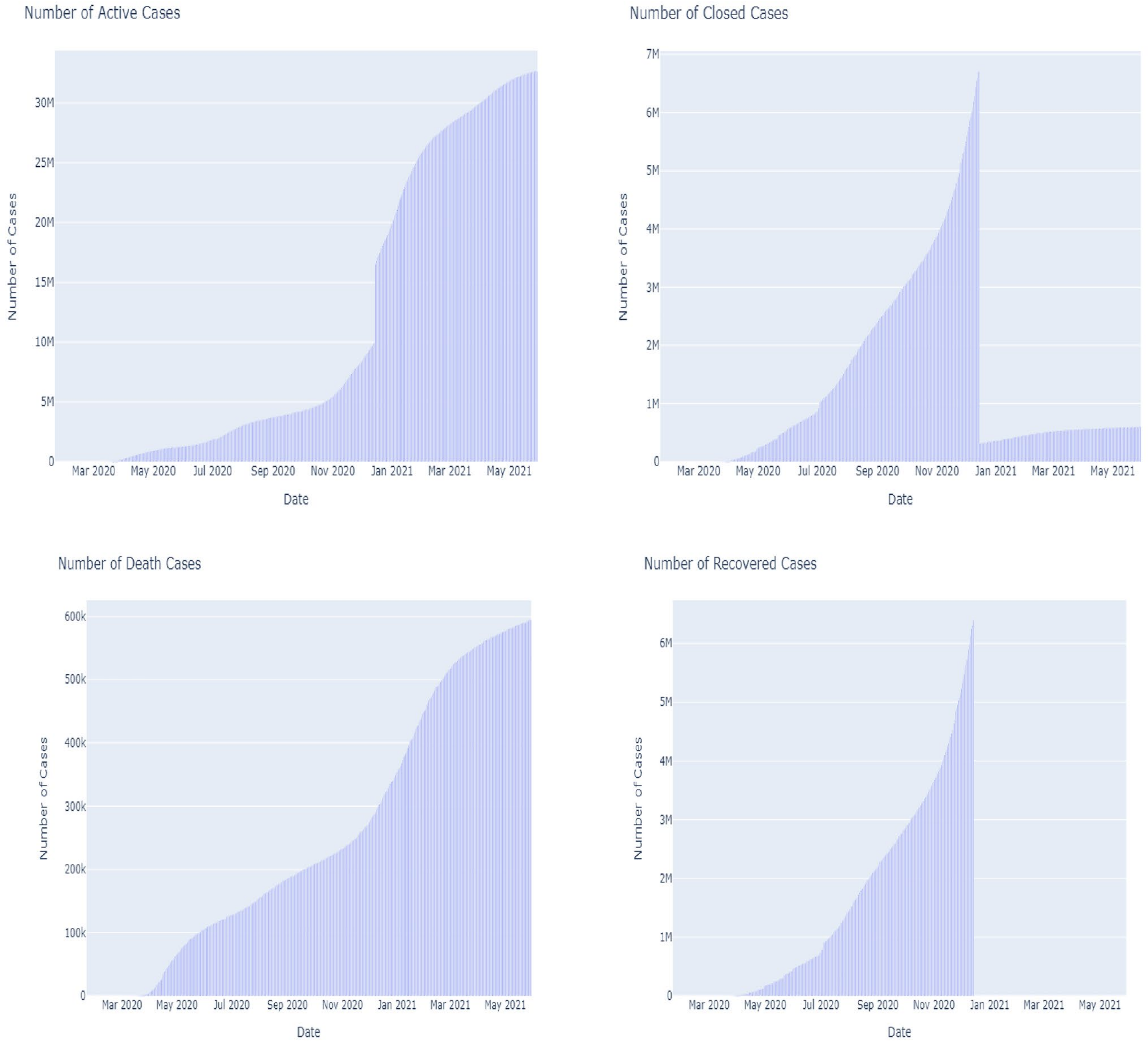


Fig. 4 US's COVID-19 scenario

According to Fig. 7, it was discovered that 16,471,600.0 cases had been confirmed, 133,765.0 were active, 87,747.0 had died, and 3,229,296.0 had been closed. In Brazil, from January 2020 to 29th May 2021, the total number of cases was 3,229,296.0 and 3,141,549.0 of those cases were recovered. In addition, the number of confirmed cases per day was found to be about 7,457.0, and the number of fatality cases per day was calculated to be around 195.0.

According to Fig. 8, it was discovered that Colombia has experienced 3,363,061.0 instances of confirmed disease, 133,765.0 cases of current cases, 87,747.0 cases of death, 3,229,296.0 cases of closed cases, and 3,141,549.0 cases of recovered disease during the first 5 months of 2020 and 2021. In addition, the number of confirmed cases per day was found to be about 7,457.0, and the number of fatality cases per day was calculated to be around 195.0.

The data as shown in Fig. 9 have been gathered by Italy's Department of Public Health which shows that there were 421,305,055.0 confirmed cases, 24,19,660 active cases, 12,6020 death cases, 39,710,890.0 closed cases, and 38,450,877.0 recovered cases from January 2020 to 29th May 2021. To this, we may add the approximate total number of cases each day: 8687.0, the approximate number of deaths each day: 260.0, and the approximate total number of cases each day: 7928.0.

In Fig. 10, it was discovered that from January 2020 to 29th May 2021, there were 523,596,780 confirmed cases, 944,281 active cases, 47,271 death cases, 514,155,50

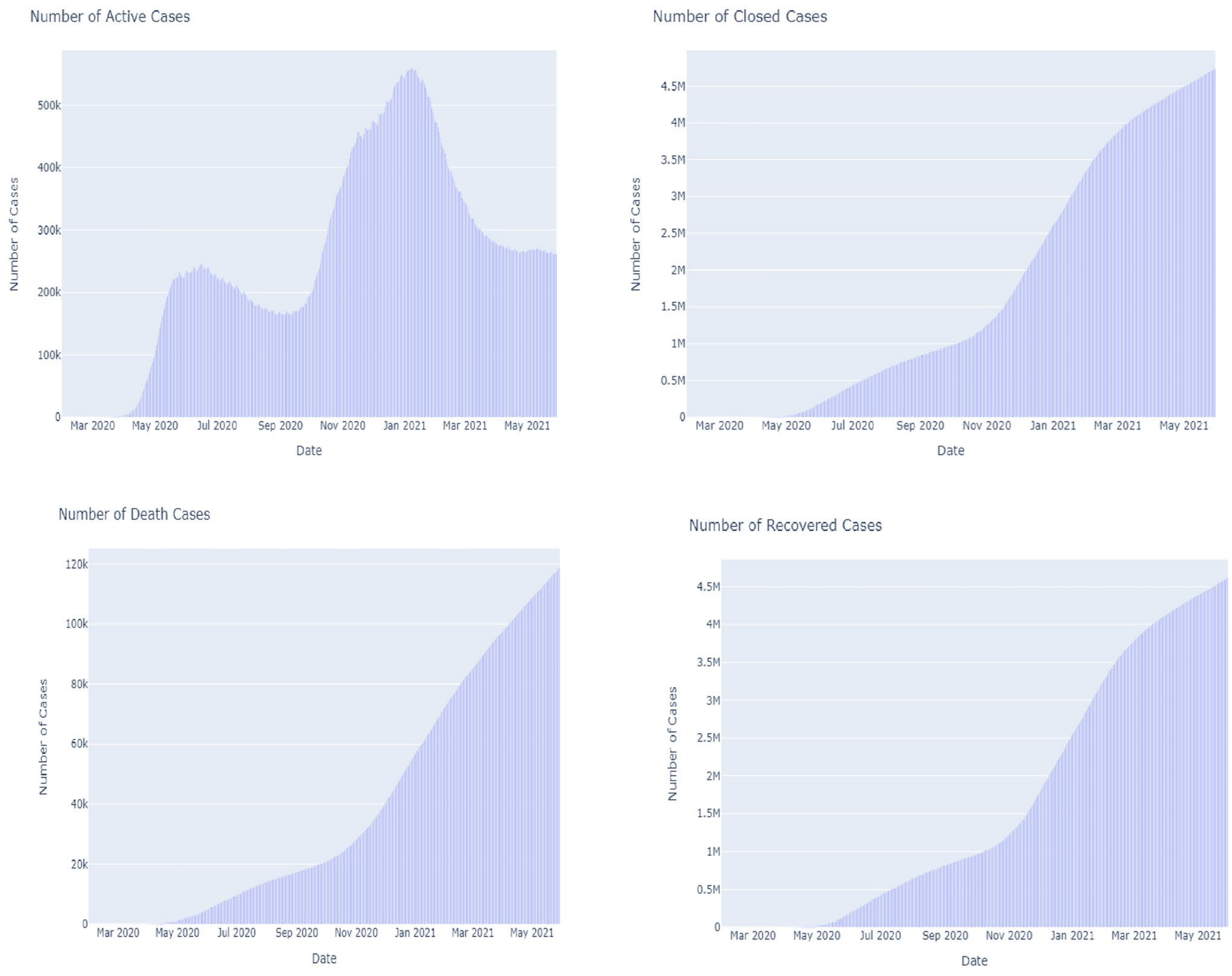


Fig. 5 Russia's COVID-19 scenario

recovered cases, and 50,942,79 newly discovered cases. In addition to this, on an average 11,766.0 confirmed cases were found every day, on an average 106.0 death cases were found every day, and on an average 11,448.0 cases were found every day.

Looking at Fig. 11, Germany had 368,674,702 cases throughout the time span from January 2020 to 29th May 2021, with 116,559 active cases, 884,130 death cases, and 35,684,113 open cases. The overall daily case counts were as follows: 7551.0 confirmed cases, 181.0 death cases, and 7131.0 recovered cases.

Figure 12 shows that in France, there were 57,198,777.0 confirmed cases, 52,191,481.0 active cases, 10,953,178.0 death cases, and 500,396.0 closed cases, with 39,087,780.0 recovered cases between January 2020 and May 2021. The results of this analysis also show that there were roughly 116,626 confirmed cases, nearly 223 deaths, and about 794 recovered cases each day.

Feature Scaling

Normalizing the range of independent variables or features of data using feature scaling is a feature scaling approach. Min–Max scaling technique has been used to perform normalization on the parts obtained during data pre-processing. The Min–Max Normalization or Min–Max Scaling technique creates a scale that goes from 0 to 1 or from 1 to − 1. Deciding on a range of data to aim for relies on the type of data you are working with. Min–Max for the range[0,1] can be computed using Eq. (4)

$$x^{'} = \frac{x - \min(x)}{\max(x) - \min(x)}. \tag{4}$$

Here, x is the original value and $x^{'}$ is the normalized value [30]. To rescale a range between any arbitrary set of values $[a, b]$, Eq. (4) becomes Eq. (5)

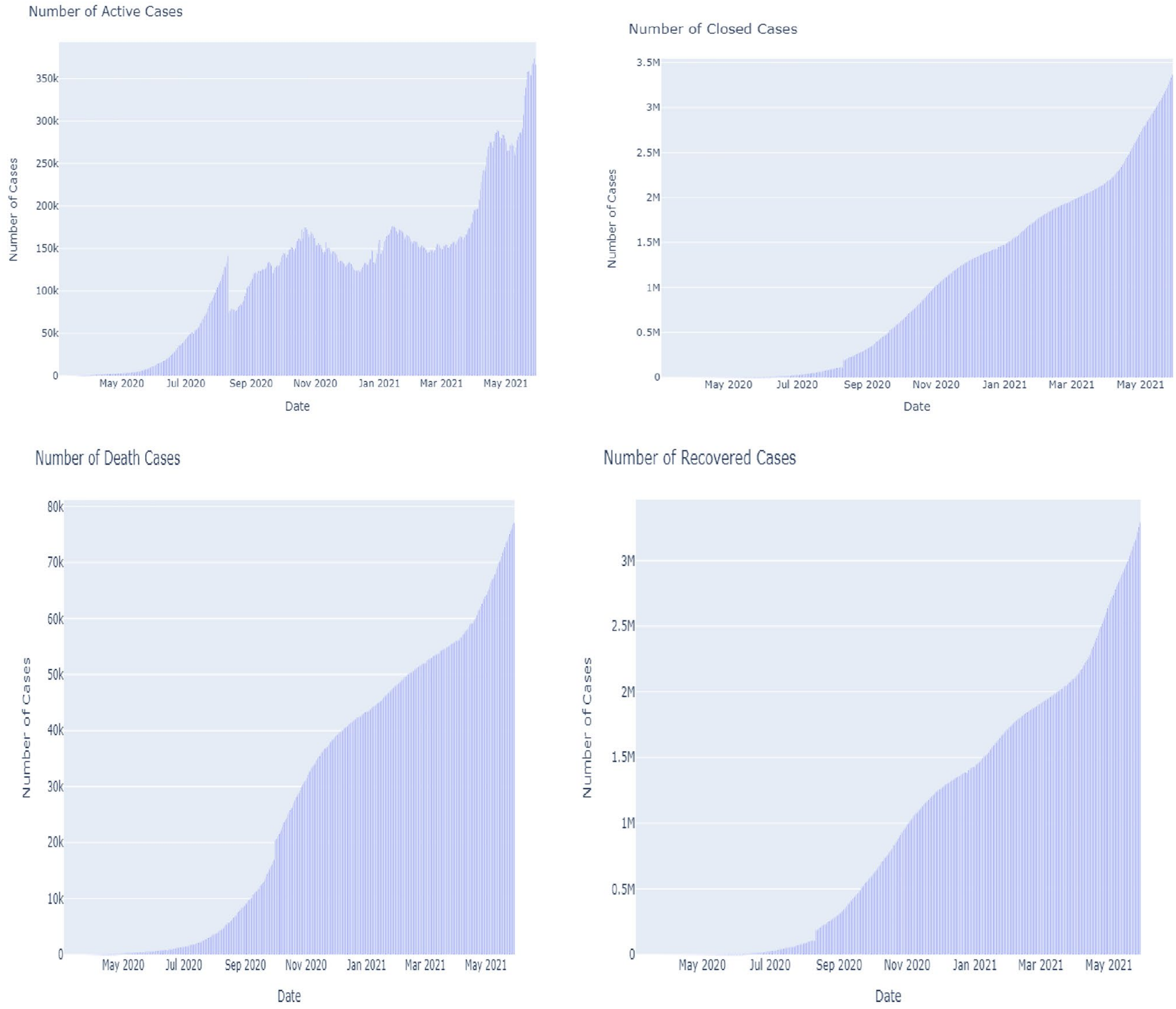


Fig. 6 Argentina's COVID-19 scenario

$$x^{'} = a + \frac{(x - \min(x))(b - a)}{\max(x) - \min(x)}. \tag{5}$$

After normalization, the data were split into two subsets: the training set, which would be used to assess machine learning methods, and the testing set, which would be used to evaluate deep learning techniques. It applies to issues involving classification or regression, as well as to any supervised learning technique. Following data partitioning, the first subset is utilized to fit the model; this is the training dataset. The second subset is used as an input element in the dataset supplied to the model, and predictions and comparisons to predicted values are performed. The test dataset is the second dataset. In a nutshell, the train data set is used to fit the machine learning model, while the test data set is used to verify the fit. The goal is to assess the performance of time series, machine learning, and deep learning models on new data. The most often used split percentages are as follows:

80% training, 20% testing.
67% training, 33% testing.
50% training, 50% testing.

Model Selection

Three sets of models have been used such as machine learning models (random forest regressor, decision tree regressor, K-nearest regressor, Kernel ridge regressor, XG Boost, RANSAC regressor, Linear regression, Lasso regression, Elastic Net regressor, Bayesian regressor, and Theilsen regressor), time series models (Facebook Prophet model and Holt model), and deep learning models (stacked long

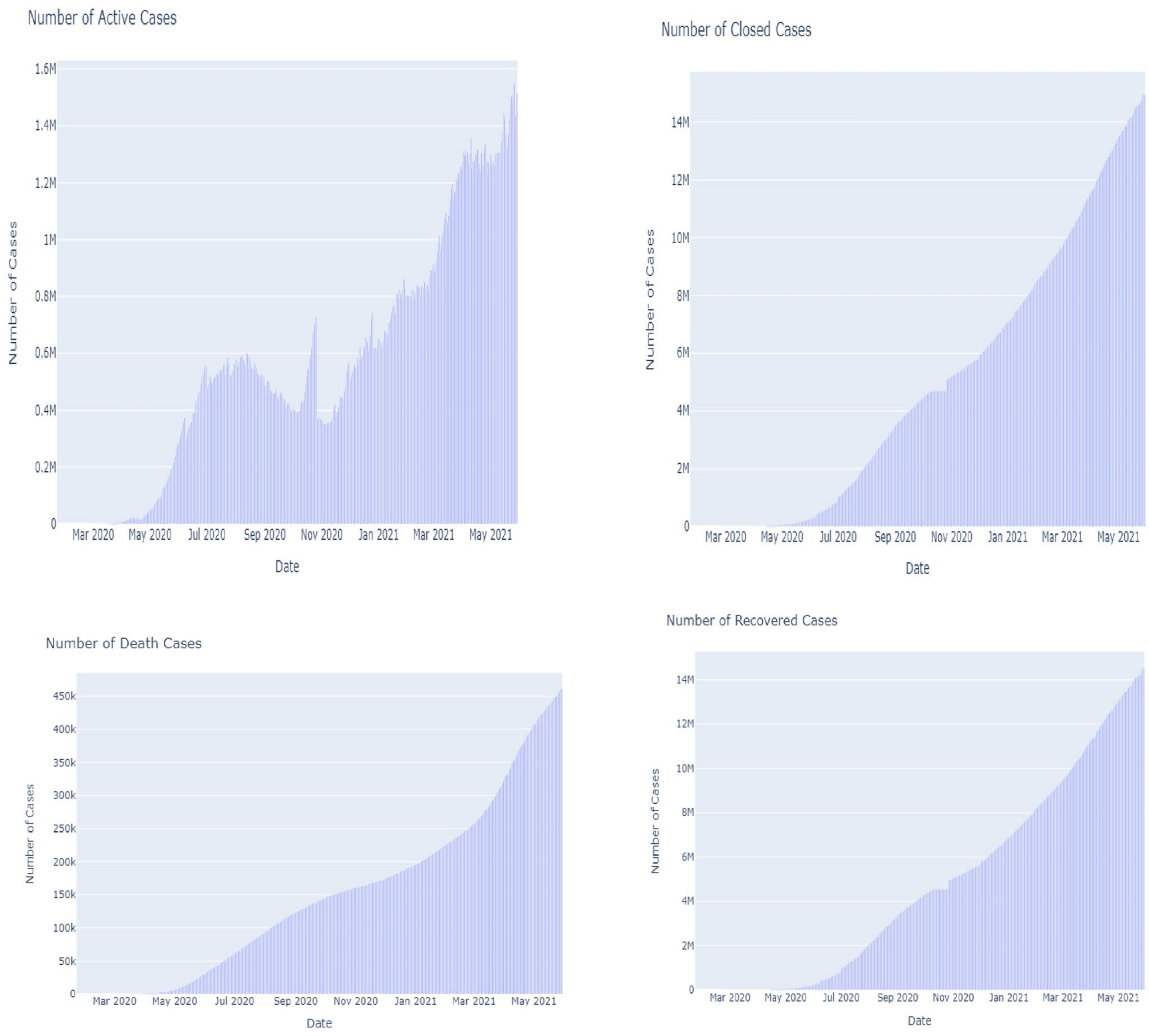


Fig. 7 Brazil's COVID-19 scenario

short-term memory and stacked gated recurrent unit) have been used to predict the confirmed cases, recovered cases. Death cases are discussed in this section.

Machine Learning Models *Random Forest Regressor*

Regression using random forest regression is a supervised learning approach that employs ensemble learning techniques to develop an accurate prediction model. During the training period, a random forest is created by several decision trees, and the output is the mean of the classes. A random forest regression model is robust and accurate and works primarily on nonlinear problems [31]. It can be calculated using Eqs. (6), (7)

$$\mathrm{RFfi}_i = \frac{\sum_{j\in \text{ all } trees} \mathrm{normfi}_{ij}}{T}, \tag{6}$$

where

$$\mathrm{normfi}_i = \frac{\mathrm{fi}_i}{\sum_{j\in \text{ all features}} \mathrm{fi}_i}, \tag{7}$$

RFfi sub(i) = the importance of feature i calculated from all trees in the random forest model. normfi sub(ij) = the normalized feature importance for i in tree j. T = total number of trees.

Decision Tree Regressor

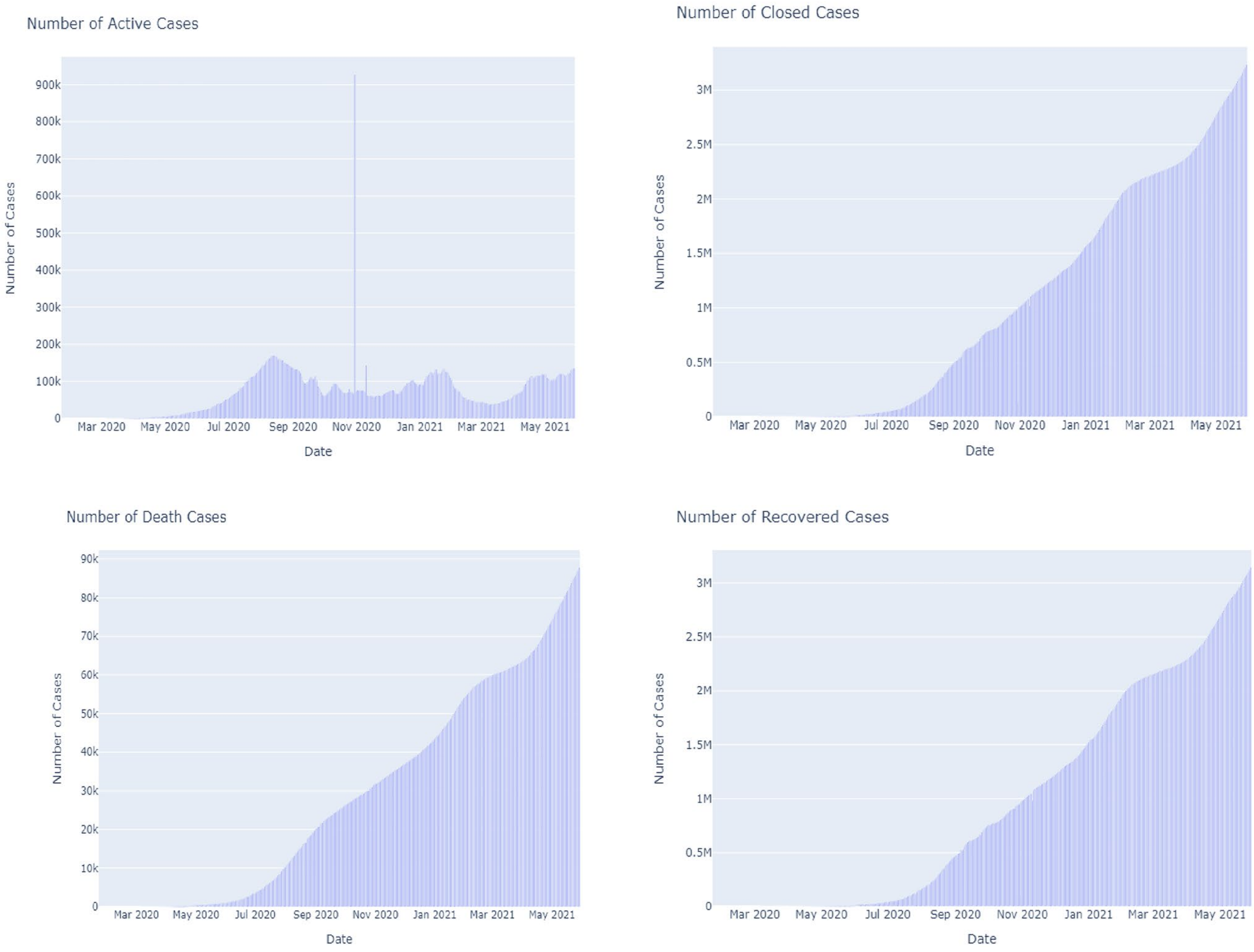


Fig. 8 Colombia's COVID-19 scenario

The decision tree algorithm is an example of a supervised learning algorithm. Regression and classification challenges may be solved using a decision tree, unlike other supervised learning techniques. To forecast the class or value of a target variable, use fundamental decision rules from previous data as building blocks for a training model that incorporates decision rules outside the training dataset (training data). At the tree's root, we forecast a class label for a record. When it comes to root attributes and record attributes, the values are compared. When we find the node with that particular value, we follow the branch corresponding to that value and go to the next node [32].

K Nearest Regressor

Non-parametric regression involves averaging nearby observations to determine if one or more independent variables are associated with a continuous result. For an analysis to be effective, the size of the neighborhood should be selected by the analyst. However, in some cases, it can be randomized to reduce the mean squared error. An algorithm that considers the K-nearest neighbor numerical objective is utilized to determine the average of the K target values. KNN regression and KNN classification both utilize the same distance functions [33]. KNN regression uses the same distance functions as KNN classification. The formulae to compute K-nearest regressor are shown in Eqs. (8)–(10)

$$\text{Euclidean formula}: \sqrt{\sum_{i=1}^{k}\left(x_i - y_i\right)^2}, \tag{8}$$

$$\text{Manhattan formula}: \sum_{i=1}^{k}\left|x_i - y_i\right|, \tag{9}$$

$$\text{Minkowski formula}: \left[\sum_{i=1}^{k}\left(\left|x_i - y_i\right|\right)^q\right]^{\frac{1}{q}}. \tag{10}$$

Kernel Ridge Regressor

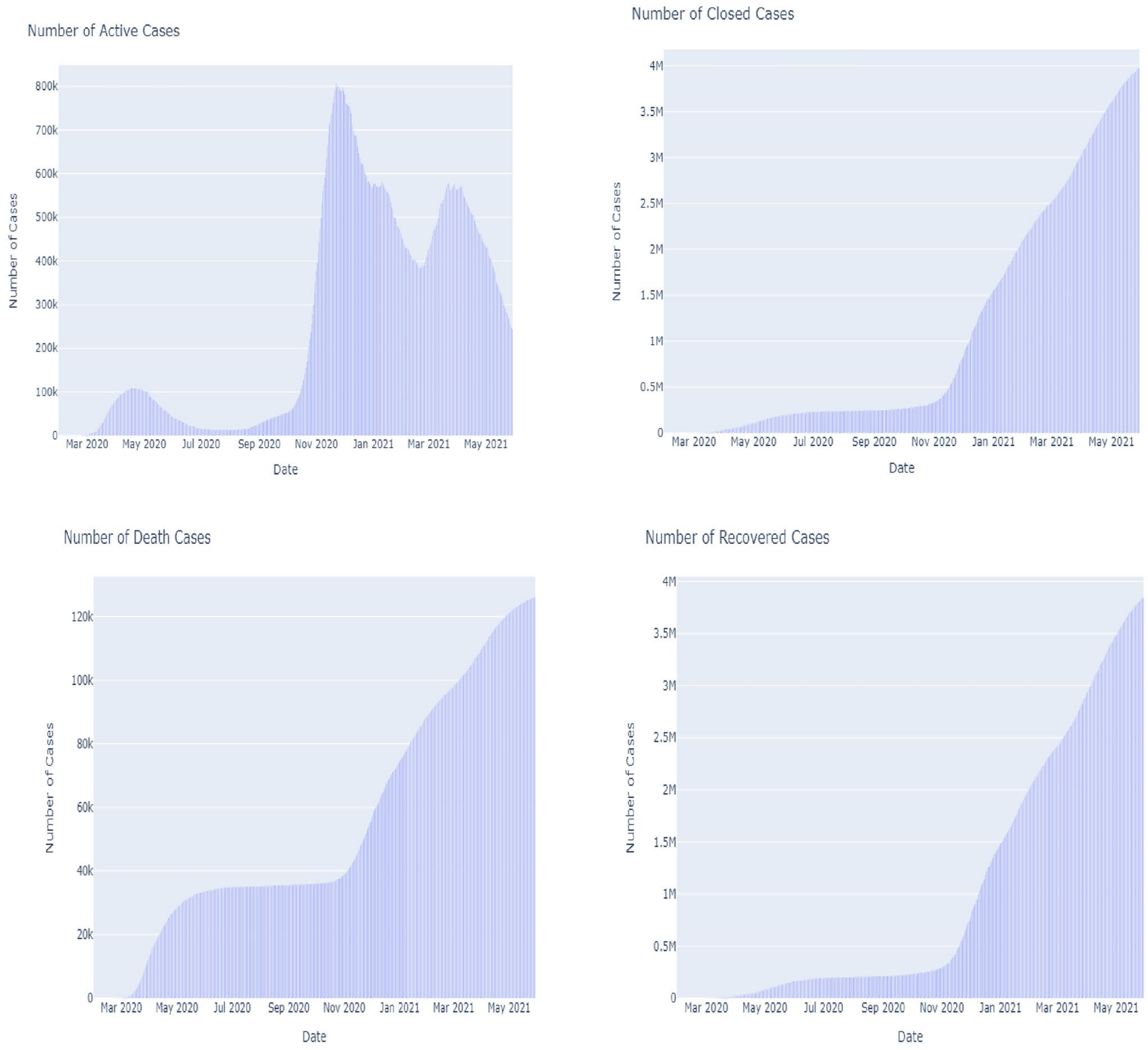


Fig. 9 Italy's COVID-19 scenario

Using the kernel method in combination with ridge regression creates a new regression technique called Kernel Ridge Regression (KRR). It is a type of ridge regression that is non-parametric. Our goal is to learn a function in the space defined by the kernel k using an approach known as minimization with optimization, and we define a squared loss with a squared norm regularization term. The kernel ridge regression does a linear function in the data space that is also proportional to the relevant kernel [34]. The equation can be written as Eq. (11)

$$\alpha = (K + \tau I)^{-1} y, \tag{11}$$

where K is the kernel matrix and α is the vector of weights in the space induced by the kernel.

XGBoost

A computer algorithm called XGBoost stands for "eXtreme Gradient Boosting." Supervised regression models are built using this method. XGBoost is a gradient boosted decision tree algorithm that is efficient and fast. XGBoost is a collection of software libraries with several user interfaces, such as the Command Line Interface (CLI), C++, Python,

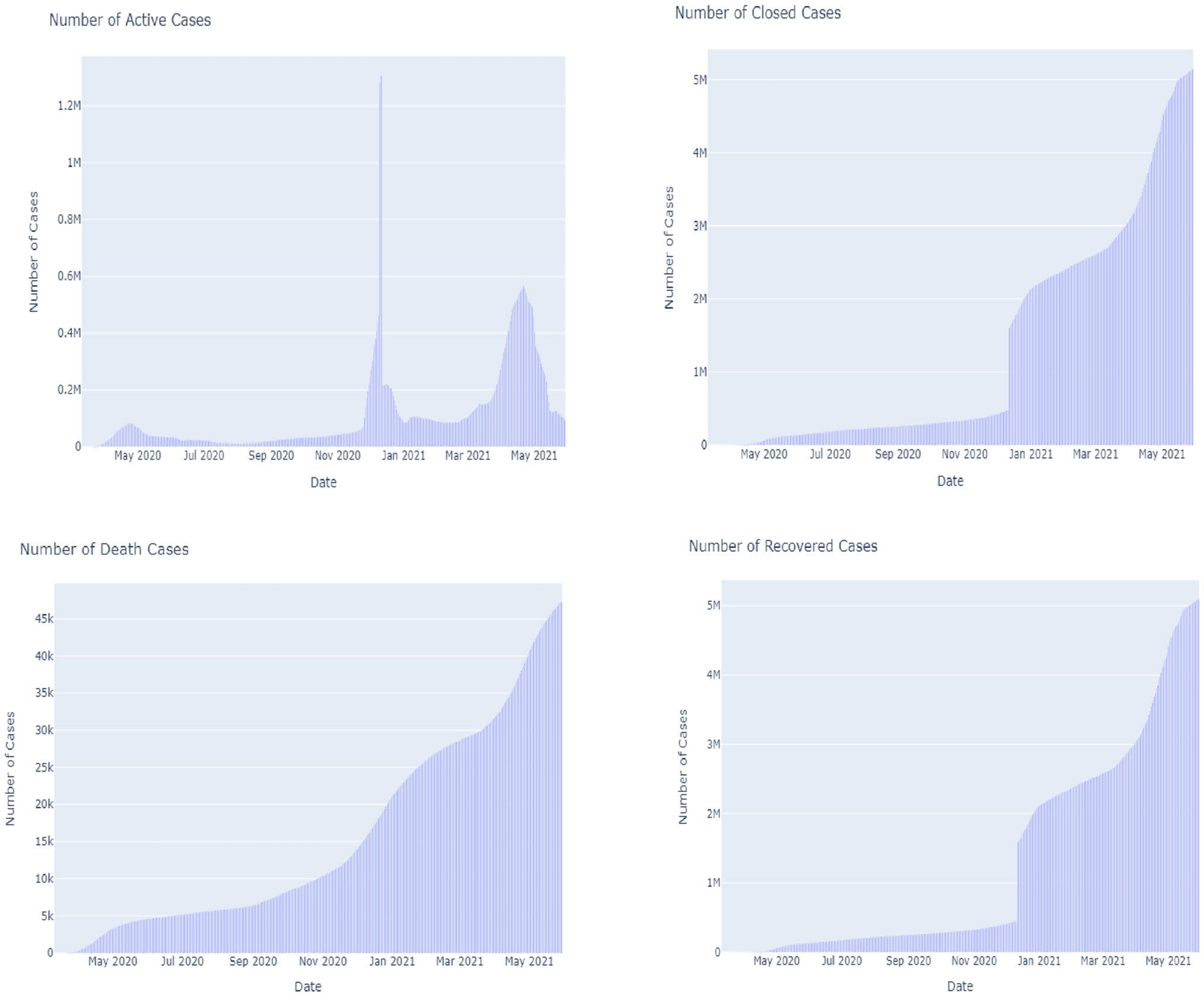


Fig. 10 Turkey's COVID-19 scenario

R, Java, and JVM interfaces. Three primary classes of boosting techniques are supported by XGBoost, including gradient boosting, stochastic gradient boosting, and regularized gradient boosting. The main reason to use XGBoost is to speed up model execution and gain project execution speed. Regression loss functions, such as linear and logistic, are most commonly used with XGBoost for regression issues [25]. The formula to compute it is shown in Eqs. (12, 13)

$$L(\emptyset) = \sum_{i} l\left(\hat{y}_i, y_i\right) + \sum_{k} \Omega\left(f_k\right), \tag{12}$$

where

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2, \tag{13}$$

y_i is a real value (label) known from the training data set.

RANSAC Regressor

RANSAC regressor is also known as the RANdom SAmple Consensus algorithm. It is an iterative algorithm used for the robust estimation of parameters by excluding the outliers in the training dataset. RANSAC is a nondeterministic algorithm as it produces a good result only with a certain probability. This method uses machine learning and random sampling of observable data to estimate model parameters in conjunction with a voting system. The RANSAC algorithm needs to be executed to perform RANSAC analysis. The following formula is used to determine the results of the RANSAC algorithm. It involves p, the probability that the RANSAC algorithm returns valuable results, and w, the likelihood of selecting an inlier on each point. Each time a single point is selected, there is a probability of picking an inlier. The possibility of choosing an inlier on every single point is called w. The chance of picking an inlier each time a single

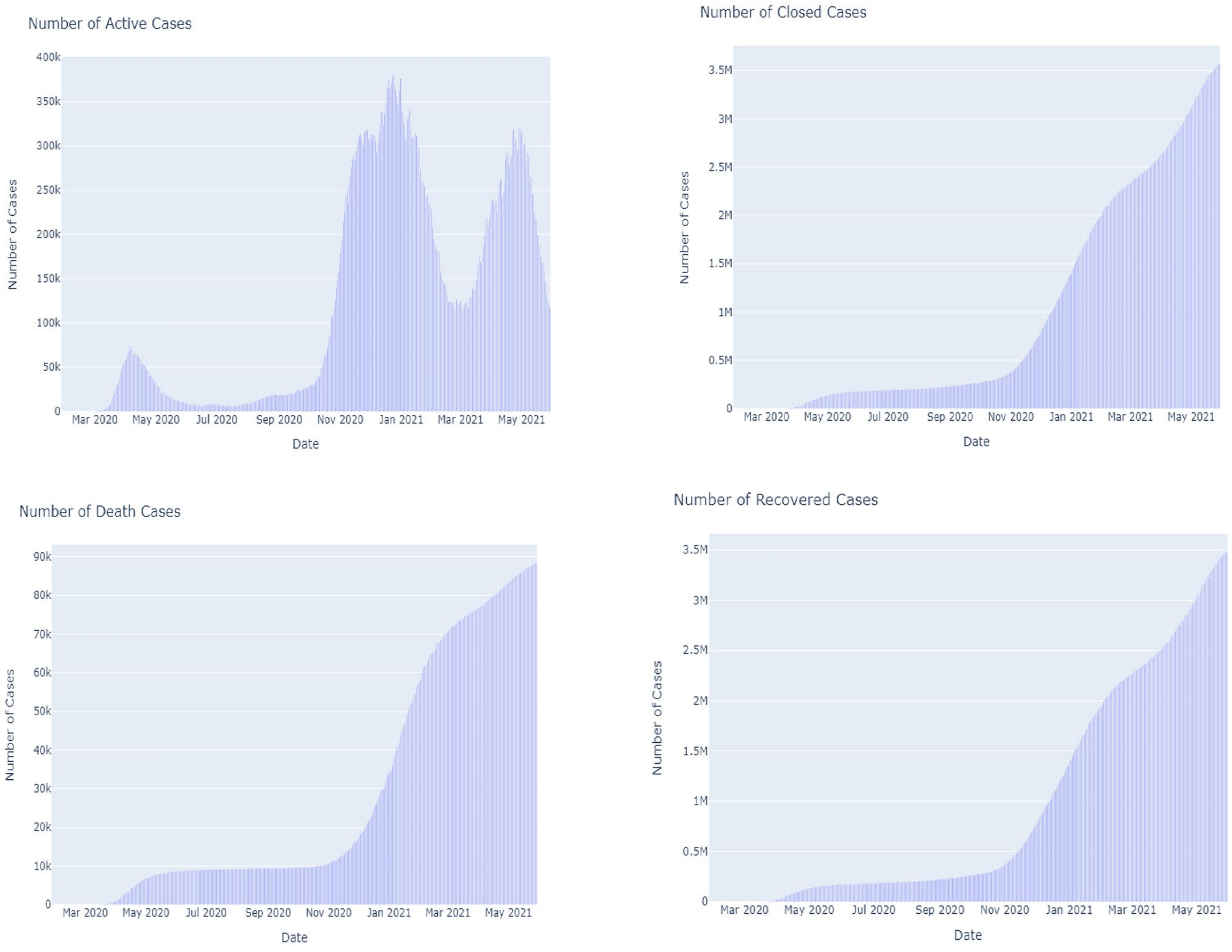


Fig. 11 Germany's COVID-19 scenario

point is set is called *wn*. Then, 1-*wn* is the probability that at least one of the *n* points is an outlier. Finally, *k* is the number of iterations [35]. The likelihood that the algorithm never selects a set of *n* inlier points is shown by Eqs. (14), (15)

$$(1 - p) = (1 - w^n)^k, \tag{14}$$

and after taking the logarithm of both sides, Eq. (14) becomes

$$k = \frac{\log(1 - p)}{\log(1 - w^n)}. \tag{15}$$

Linear Regression

A machine learning approach that uses supervised learning, known as Linear Regression Analysis (LRA), is a supervised learning algorithm. The model may be trained to predict the outcome of data using a given set of factors using a linear regression method. In quantitative sciences, linear regression is typically used to indicate a quantitative response from the predictor variable. It is intended to show how an independent variable affects the goal prediction value. In forecasting, it is used to determine how variables are related [36]. Linear regression can be written as by Eqs. (16)–(18)

$$y = a + bx, \tag{16}$$

where a and b are given by the formulae

$$b(\text{slope}) = \frac{n\sum xy - (\sum x)(\sum y)}{n\sum x^2 - (\sum x)^2}, \tag{17}$$

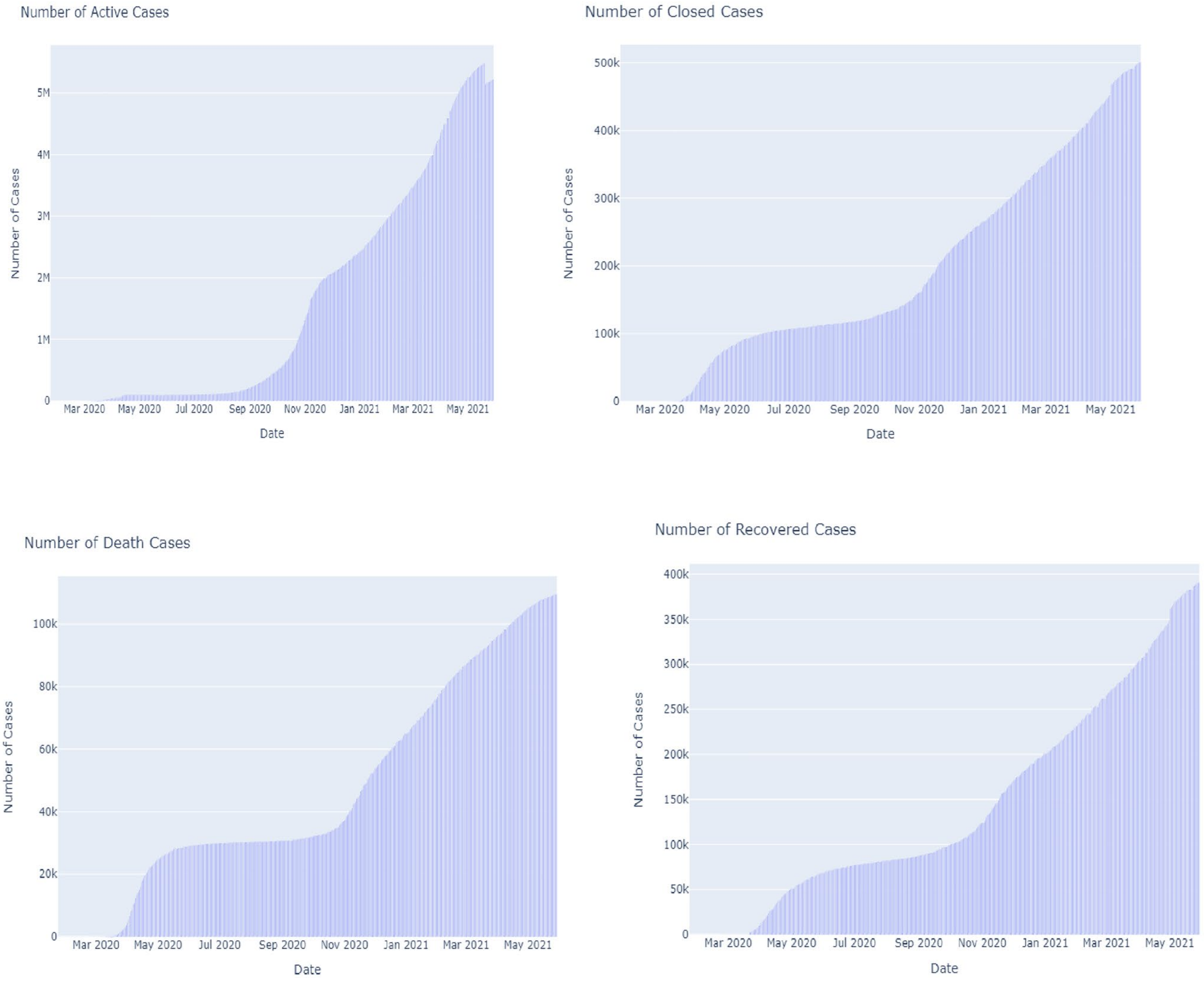


Fig. 12 France's COVID-19 scenario

$$a(\text{intercept}) = \frac{n\sum y - b\left(\sum x\right)}{n}. \tag{18}$$

Here, x and y are two independent and dependent variables, respectively, on the regression line, b is slope of line, and a is an intercept of the line.

Lasso Regression

The "LASSO" stands for Least Absolute Shrinkage and Selection **O**perator, which is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. The lasso technique promotes the use of sparse, basic models (i.e., models with fewer parameters). This form of regression is well suited for models with a high degree of multicollinearity or automating some aspects of model selection, such as variable selection/parameter removal, as necessary [37]. The mathematical equation of lasso regression is shown in Eq. (19)

$$\text{Min}_{\beta} L_1 = \sum_{i=1}^{n}\left(y_i - \sum_{j} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\left|\beta_j\right|. \tag{19}$$

For simplicity, let $p = 1$ and $\beta_i = \beta$. Now, Eq. (19) becomes Eq. (20)

$$L_1 = (y - x\beta)^2 + \lambda|\beta| = y^2 - 2xy\beta + x^2\beta^2 + \lambda|\beta|, \tag{20}$$

where λ represents the amount of shrinkage.

Elastic Net Regressor

Elastic net linear regression regularizes regression models using both the lasso and ridge methods as shown in Eq. (21). By learning from the inadequacies of both lasso and ridge regression approaches, the methodology integrates both to enhance the regularization of statistical models. The elastic net technique overcomes the drawbacks of the lasso method, namely that it only requires a few samples for high-dimensional data. The flexible net technique allows for the addition of "n" variables until saturation is reached. When the variables are highly linked groups, lasso tends to pick one variable from each group and disregard the others completely. To overcome the constraints of a lasso, the elastic net incorporates a quadratic expression in the penalty, which becomes ridge regression when employed alone. The first step is determining the ridge regression coefficients, followed by a lasso sort of coefficient shrinkage [38]. In a nutshell

$$\text{ENR} = \text{Lasso Regression} + \text{Ridge Regression}, \text{ where,} \tag{21}$$

$$\text{Lasso Regression} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \left(mx_i + z\right)\right)^2 + \lambda\sum_{i=1}^{p}(mx_i + z), \tag{22}$$

$$\text{Ridge Regression} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \left(mx_i + z\right)\right)^2 + \lambda\sum_{i=1}^{p}\left(mx_i + z\right)^2. \tag{23}$$

Using both Eqs. (22), (23), we get Eq. (24)

$$\text{ENR} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \left(mx_i + z\right)\right)^2 + \lambda\sum_{i=1}^{p}\left(mx_i + z\right)^2 + \lambda\sum_{i=1}^{p}(mx_i + z). \tag{24}$$

Bayesian Regressor

Bayesian Regressor is a regression approach that uses Bayesian inference to do statistical analysis. This method enables a natural process to persist in the presence of limited or poorly dispersed data. It generates predictions based on the posterior probability of all feasible regression weights. With Bayesian Linear Regression, the aim is not to choose the "best" model parameter but to estimate the distribution of model parameters [39]. It is demonstrated by Eq. (25)

$$P(\beta|y, X) = \frac{P(y|\beta, X) \times P(\beta|X)}{P(y|X)}. \tag{25}$$

Here, $P(\beta|\text{y}, X)$ is the posterior probability distribution of the model parameters given the inputs and outputs. This is equal to the likelihood of the data, $P(\beta|\text{y}, X)$, multiplied by the prior probability of the parameters and divided by a normalization constant.

Theilsen Regressor

Theilsen regressor is a non-parametric statistic where a line is fitted to sampled points in the plane by selecting the median of the lines connecting pairs of points. Theilsen regression is a fast algorithm that is insensitive to outliers. Additionally, it has been referred to as the most widely used non-parametric approach for estimating a linear trend. The two-dimensional point Theilsen regression x_i, y_i is the median m of the slopes $\frac{(y_j - y_i)}{(x_j - x_i)}$ based on all pairwise sampling locations [40]. Once the slope m has been determined, we can find a line from sample points by setting the y intercept b to be the median of the values $y_i - mx_i$. A variant to Theilsen regression can be calculated using Eq. (26)

$$r_{\text{TS}}(x, y) = \text{sign}\left(m_{\text{TS}}(y, x)\right) \cdot \sqrt{m_{\text{TS}}(y, x) \cdot m_{\text{TS}}(x, y)}. \tag{26}$$

Time Series Models *Facebook Prophet*

A forecasting approach based on an additive model known as a prophet is used to correlate nonlinear trends with seasonal and holiday impacts as well as yearly, weekly, and daily patterns. Time series with strong seasonal influences and extensive historical data spanning many seasons do well with this approach. The Prophet works well with outliers, which makes it resistant to data and trend shifts. The time series model is built on a prophet, and it is fast, fully automated, and very exact. The trend, seasonality, and holidays form our time series model, which we break down into three key components: trend, seasonality, and holidays [24]. They are merged in Eq. (27) as follows:

$$y(t) = g(t) + s(t) + h(t) + \in t, \tag{27}$$

$g(t)$: For modeling non-periodic changes in time series, a piecewise linear or logistic growth curve is used. $s(t)$: changes on a regular basis (e.g., weekly/yearly seasonality). $h(t)$: The impact of vacations (supplied by the user) on individuals with irregular schedules. εt: The error term is used to account for any unforeseen changes that the model does not account for.

Holt Model

The Holt model is a well-known technique for predicting data with a trend. Holt's model consists of three distinct equations that interact to create a final forecast. The first is a fundamental smoothing equation, often known as the level equation, which directly adjusts the previous smoothed value for the trend of the previous period. The trend is updated over time using the second equation, which expresses the trend as the difference between the previous two smoothed values. Finally, the final forecast is generated using the third equation. Holt's approach makes use of two parameters: one for global smoothing and another for the trend smoothing equation. Additionally, this technique is referred to as double exponential smoothing or trend-enhanced exponential smoothing [41]. It is computed using Eqs. (28)-(30)

$$\text{Level equation} = l_t = \alpha y_t + (1 - \alpha)\left(l_{t-1} + \text{Ø} b_{t-1}\right), \tag{28}$$

$$\text{Trend equation} = b_t = \beta^*\left(l_t - l_{t-1}\right) + (1 - \beta^*)b_{t-1}, \tag{29}$$

$$\text{Forecast equation} = \hat{y}_{t+h|t} = l_t + hb_t, \tag{30}$$

where l_t represents the estimation of the series' level at time t, b_t represents the estimation of the series' trend (slope) at time t, and α and $\beta*$ are the smoothing parameters for

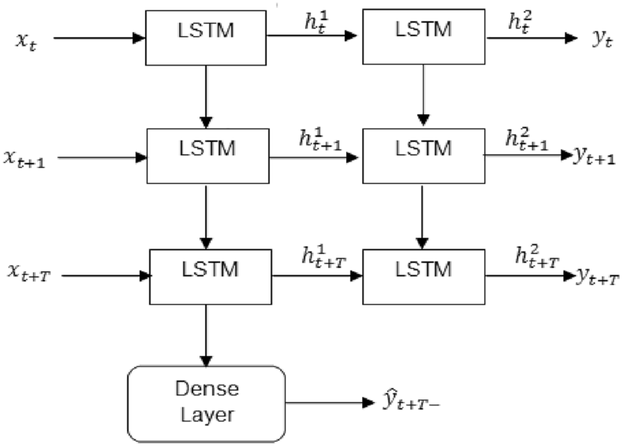


Fig. 13 Stacked LSTM architecture

the level, $0 \leq \alpha \leq 1$, and trend, $0 \leq \beta^* \leq 1$, respectively. l_t is a weighted average of observation y_t and the one-step-ahead training forecast for time t, denoted here by $l_t - 1 + b_t - 1$. b_t is a weighted average of the estimated trend at time t based on $l_t - l_t - 1$ and $b_t - 1$, the trend's earlier estimations, according to the trend equation. The prediction for the next h steps forward is equal to the most recent predicted level multiplied by h times the most recent estimated trend value. As a result, the predictions in terms of h are linear.

Deep Learning Models *Stacked LSTM*

Deep LSTM is another name for an LSTM that has a large number of LSTM layers. The model described in Fig. 13 is called a stacked LSTM, with several hidden LSTM layers layered on top of each other.

Assume i_t^l, f_t^l, o_t^l, c_t^l and h_t^l are the values of the input gate, forget gate, output gate, memory cell, and hidden state using Eqs. (31)–(35) at time t in the sequence and layer l, respectively. $x_{t,k}$ is the input of the system at time t at location k, whereas W_{xj} for $j \in \{i, f, o, c\}$ are the weights that connect the input, $x_t = [x_{t,1}, x_{t,2}, \ldots, x_{t,c}]^{\mathrm{T}}$ to the corresponding gates and the memory cell [42]

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right), \tag{31}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right), \tag{32}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \tag{33}$$

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o\right), \tag{34}$$

$$h_t = o_t \odot \tanh\left(c_t\right). \tag{35}$$

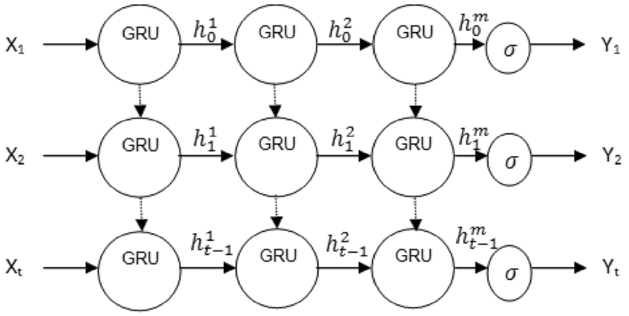


Fig. 14 Stacked GRU architecture

Stacked GRU

The simple model, GRU, is incapable of doing advanced feature extraction. On the other hand, the deep model, stacked GRU, is formed from several simple models, with the input of the first layer being the original data, as shown in Fig. 14.

Increased classifier performance may be realized by making use of time series data. Instead of considering whether the models are time-dependent, these individuals sidestep the trade-off between time and precision. As described in Eqs. (36)–(39) [43], the central GRU unit receives the output of the top GRU unit's hidden layer. A sigmoid layer is added to the preceding layer's hidden layer to accomplish the ultimate result in Eq. (40)

$$z_t^i = \sigma\left(W_z^i \cdot \left[h_{t-1}^i, h_t^{i-1}\right]\right), \tag{36}$$

$$r_t^i = \sigma(W_r^i \cdot [h_{t-1}^i, h_t^{i-1}]), \tag{37}$$

$$h_t^{\sim i} = \tanh(W^i \cdot [r_t^i \odot h_{t-1}^i, h_t^{i-1}]), \tag{38}$$

$$h_t^i = z_t^i \odot h_{t-1}^i + \left(1 - z_t^i\right) \odot h_t^{i-1}. \tag{39}$$

Here, z represents the update gate, r is the reset gate which is used to control the direction of the data stream at time t, h_{t-1} is the output of the hidden layer, and $\tilde{h}_t$ is the output of candidate hidden layer at time t

$$\tilde{y}_{\text{last}} = \sigma\left(W_{\text{o}}^n h_{\text{o}}^n + b_{\text{o}}^n\right). \tag{40}$$

Here, $\tilde{y}_{\text{last}}$ is the predicted label at the last moment, W_{o}^n is the weight of the output layer, and b_{o}^n is the bias of the n-th GRU unit.

Evaluative Parameters

RMSE The usual technique of quantifying the error of a model in quantitative data is the root-mean-square error. It is defined by an Eq. (41). By identifying the error, the dataset reveals how distant each data point is from a regression line, and the root-mean-square error quantifies how concentrated each data point is around the line of best fit [44]

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{\left(\hat{y}_i - y_i\right)^2}{n}}, \tag{41}$$

$\hat{y}_i$ are predicted vales, y_i are observed values, and n is the number of observations.

R^2 Score The statistician's coefficient is a model's ability to predict or explain a result in a regression setting. R^2 Score is a percentage used to quantify the amount of variance in the dependent variable that can be predicted using linear regression and the predictor variable (independent variable) [44]. It is shown by Eq. (42)

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}; \tag{42}$$

RSS is the sum of squares of residuals and TSS is the total sum of squares

Result Analysis

Different machine learning models, time series, and deep learning algorithms were used to calculate the RMSE and R^2 values, features extracted in the form of confirmed cases, death cases. They recovered points of ten different countries, such as India, USA, Russia, Argentina, Brazil, Colombia, Italy, Turkey, Germany, and France.

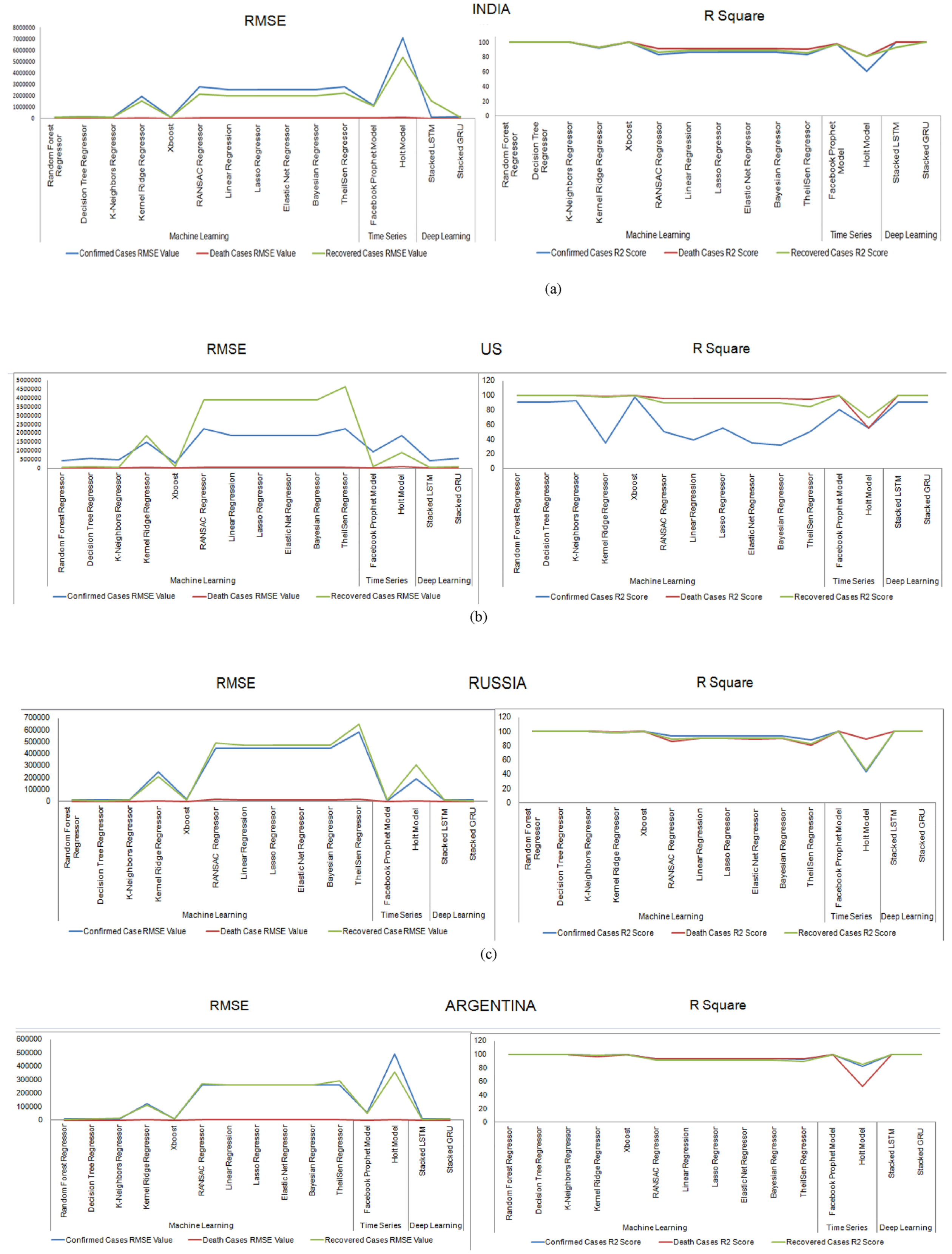
In Fig. 15, we can see the range of different algorithms for calculating the root-mean-square error and R square values of confirmed cases, death cases, and recovered cases of ten other countries. Hence, to show the best algorithm out of these three techniques, three scenarios have been taken to elaborate the values of root-mean-square error and R^2.

Scenario 1: Predict RMSE and R Square Value Using Machine Learning Models

We have used 11 algorithms in machine learning models, such as Random Forest Regressor, Decision Tree Regressor, K neighbor Regressor, Kernel Ridge Regressor, XBoost, RANSAC Linear Regression, Lasso Regression, Elastic Net Regressor, Bayesian Regressor, and Theilsen Regressor. Out of all these algorithms, Random Forest Regressor has obtained the minor root-mean-square error value for confirmed, death, and recovered cases of India by 68,302, 813, and 64,494, Italy by 7447,256 and 8283, and France by 14,391, 243, and 763, respectively, as well as *R* Square achieved by it for all the three cases of these countries, is 99.9%. On the other hand, for the US, the lowest RMSE and highest *R* square value for confirmed cases have been achieved by XBoost by 290,098 and 97.5, respectively. For death and recovered points, random forest regressor reached the lowest RMSE and highest R square value by 1159, 53,667, and 99.9, respectively. For Russia, Brazil, and Colombia, random forest regressor achieved the highest R square value of 99.9 for all the three cases and the lowest RMSE matters by (9113, 196), (8682, 846), and (8020, 241) for confirmed and death cases, respectively, while as decision tree regressor in recovered cases by 8124,29,276 and 10,027, respectively. For Argentina, the random forest regressor achieved the highest R square value of 99.9 and the lowest RMSE value of 7976 and 7438 for confirmed and recovered cases. In contrast, the decision tree regressor scored a 182 RMSE value in terms of death cases. For Turkey, Random Forest Regressor has achieved 13,539, 100 root-mean-square error values for confirmed and instances of death, while X Boost has achieved 16,465 root-mean-square error for recovered patients with 99.9 R Square. In the end, for Germany, K Neighbor Regressor has reached 11,977 and 221 root-mean-square error values for confirmed and death cases, while the random forest regressor achieved the least RMSE for recovered instances 6753.

Scenario 2: Predict RMSE and R Square Value Using Time Series Models

We have used two algorithms in time series models, i.e., Facebook Prophet Model and Holt Model. Out of these two models, Facebook Prophet Model has played an essential role by providing the lowest root-mean-square error value for confirmed, death, and recovered cases of India by 1,112,918, 12,524, and 1,061,511, the US by 922,620,2530, and 80,401, Russia by 5262,156, and 12196, Argentina by 55,118, 1048, and 51,794, Brazil by 24,606, 2174, and 38,904, Colombia by 39,239, 1208, and 63,090, Italy by 41,057, 582, and 15,202, Turkey by 111,271, 645, and 137,165, Germany by 24,606, 191, and 31,245, and France by 85,910, 361, and 2442, respectively. Moreover, Facebook Prophet Model also achieved the highest *R* Square value for confirmed, death, and recovered cases of India by 97.2, 97.5, and 96.8, the US by 80.25, 99.9, Argentina, Brazil, Italy, Germany by 99.9, Colombia by 99.8, 99.7, and 99.5, Turkey by 99.5, 99.7, and 99.1, and France by 99.8 and 99.9, respectively.


INDIA

RMSE

R Square

(a)

US

RMSE

R Square

(b)

RUSSIA

RMSE

R Square

(c)

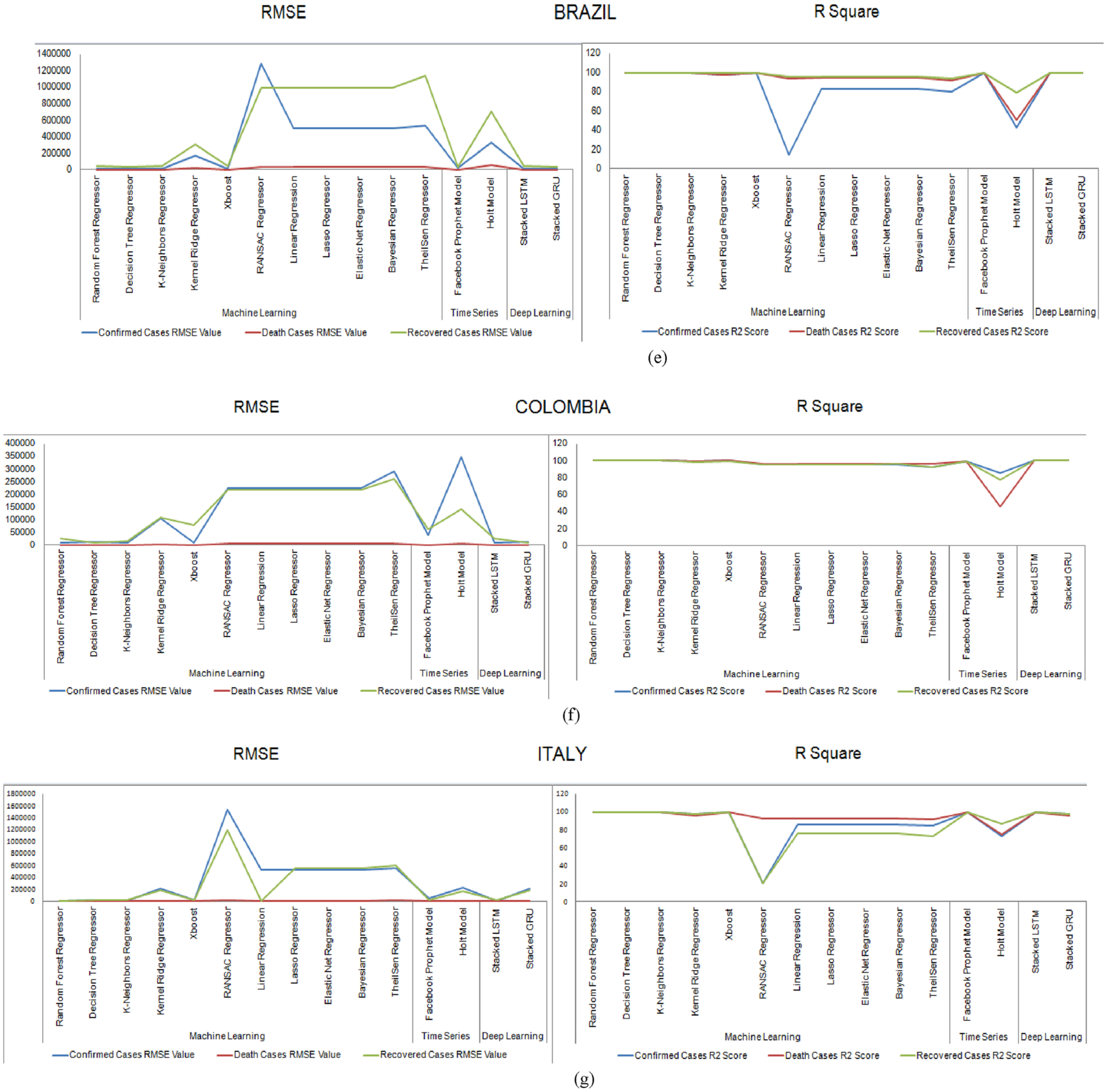
ARGENTINA

RMSE

R Square

(d)

Fig. 15 Models based analysis for confirmed, death, and recovered cases of **a** India, **b** US, **c** Russia, **d** Argentina, **e** Brazil, **f** Colombia, **g** Italy, **h**, Turkey, **i** Germany, and **j** France



(e)

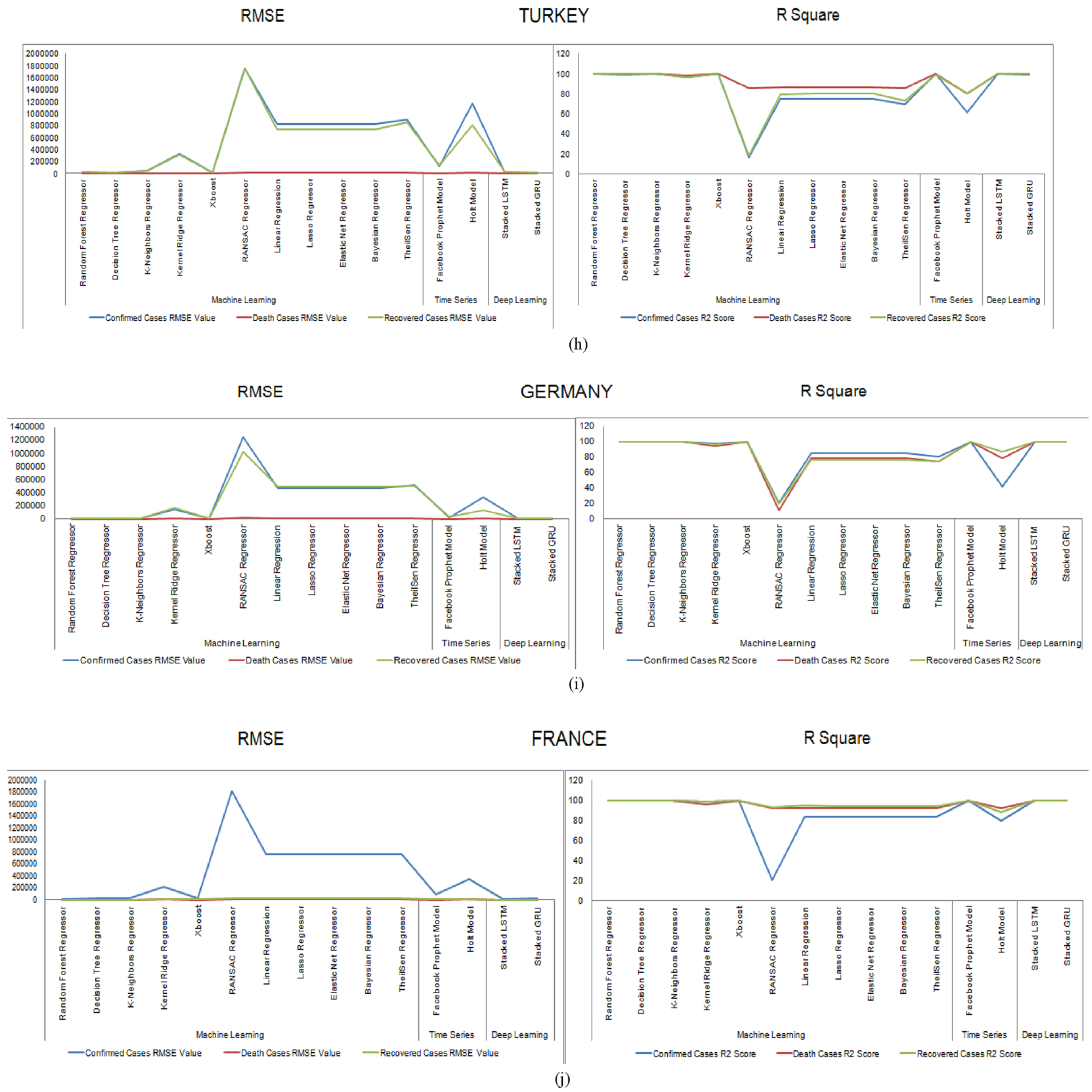
(f)

(g)

Fig. 15 (continued)

Scenario 3: Predict RMSE and R Square Value Using Deep Learning Models

We have used two algorithms in deep learning models, i.e., Stacked Long Short-Term Memory and Stacked Gated Recurrent Unit. On analyzing both the algorithms, it has been seen that Stacked Long Short-Term Memory has achieved the lowest root-mean-square error value and 99.9 R^2 value for confirmed cases, death cases, and recovered patients of US by 418,343, 1160, and 53,669, Italy by 8835, 328, and 11,256, and France by 14,389, 240, and 762, respectively. It has also obtained the lowest root-mean-square value for confirmed and death cases of India by 68,303, and 814, Russia by 9113 and 196, Brazil 8682 and 846, Turkey by 13,539, and 100, Colombia by 8020 and 241, respectively, while as the root-mean-square error value for the recovered cases of all these countries has been achieved by Stacked Gated Recurrent Unit by 65,760, 8124, 29,276, 17,462, and 10,027, respectively. Stacked Gated Recurrent Unit has gotten the least root-mean-square error value for all the three cases, such as confirmed, death, and recovered points of Germany by



(h)

(i)

(j)

Fig. 15 (continued)

11,977, 221, and 9533, respectively with 99.9 R^2. In the case of Argentina, Stacked LSTM showed the highest *R* square value by 99.9 and the lowest RMSE value by 7974 and 7433 for confirmed and recovered instances, respectively, while as in the death case, Stacked GRU scored the lowest RMSE value by 182.

Table 3 is summed up the results by showcasing the best model out of all applied machine learning, time series, and deep learning models for ten different countries. Besides,

Table 3 Country-wise RMSE and *R* square values

Countries	Confirmed cases			Death cases			Recovered cases		
	RMSE	MSE	R^2	RMSE	MSE	R^2	RMSE	MSE	R^2
India	68,302	261.34	98.4	813	28.51	99.9	64,494	253.96	99.7
USA	290,098	538.60	98.5	1159	34.04	98.4	53,667	231.66	98.6
Russia	9113	95.46	98.9	196	14	99.2	8124	90.13	98.9
Argentina	7974	89.29	99.3	182	13.49	98.8	7433	86.21	99.4
Brazil	8682	93.17	99.0	846	29.13	97.5	29,276	171.10	99.5
Colombia	8020	89.55	97.8	241	15.52	99.2	10,027	100.13	97.9
Italy	7447	86.29	99.2	256	16	99.5	8283	91.01	97.5
Turkey	13,539	116.35	96.4	100	10	97.7	17,462	132.14	99.9
Germany	11,977	109.43	97.7	191	13.82	99.6	6753	82.17	99.9
France	14,389	119.95	99.3	240	15.49	99.9	762	27.60	99.9

root-mean-square error, and *R* square, another parameter has been added, i.e., mean square error (MSE) value to test the performance of the model for three different cases, i.e., confirmed, death, and recovered, which the author will refer to as case studies throughout this paper.

Many machine learning models, such as Facebook Prophet model and stacked long short-term memory, and the random forest regressor model from confirmed, death, and recovered cases, have been found to have achieved the lowest root-mean-square error value. In contrast, the Facebook Prophet model and stacked long short-term memory had the highest *R* Square value for the cases of ten countries. It has been shown that the bulk of these calculations (for confirmed, death, and recovered cases) were done using random forest regressor and stacked long short-term memory. Moreover, time series models, machine learning models, and deep learning models were also applied to predict confirmed, death, and recovered cases for ten different countries for *random 5 days* on *separate datasets*. All results are given in Tables 4, 5, and 6, respectively.

The time forecasting prediction will help the COVID warriors estimate their country's COVID affected rate. They will provide vaccinations to the respective government agencies and protect the people from this dreadful disease. Necessary steps will also be taken to ensure the mitigation of financial, mental, and physical loss done to this devastating pandemic. Based on these future assumptions, the countries mentioned above will continuously improve to defeat this unseen enemy. In addition to this, the results are also compared in Table 7 with the existing techniques on the basis of their mean R^2 score for multiple dataset of confirmed COVID cases.

Conclusion and Future Scope

In this work, the active, recovered, closed, and death cases from March 2020 to May 2021 of ten different countries, which includes India, the United States of America, Russia, Argentina, Brazil, Colombia, Italy, Turkey, Germany, and France, were pre-processed and later graphically depicted to examine the pattern and find missing values. Further, the data have been scaled using a MinMax scaler to extract and normalize the features to acquire an accurate prediction rate. Various machine learning, time series, and deep learning models, such as Random Forest Regressor, Decision Tree Regressor, K Nearest Regressor, Lasso Regressor, Linear Regressor, Bayesian Regressor, Theilsen Regressor, Kernel Ridge Regressor, RANSAC Regressor, XG Boost, Elastic Net Regressor, Facebook Prophet Model, Holt Model, and Stacked Long Short-Term Memory, and Stacked Gated Recurrent Memory, had been applied to forecast the confirmed, death, and recovered COVID-19 cases. At last, all the models were evaluated and tested using root-mean-error square and R square values to predict COVID-19 cases for the aforementioned ten different countries, and during implementation, it was discovered that the random forest regressor and stacked long short-term memory produced the majority of the best values for all the three cases, i.e., confirmed, death, and recovered.

The research is entirely based on statistical data and methodology; hence, the results generated will help these countries to take all essential safeguards before becoming

Table 4 Prediction of confirmed cases

Models	India	US	Russia	Argentina	Brazil	Colombia	Italy	Turkey	Germany	France
Random forest regressor	27,800,239.85	33,242,792	4,990,377.92	3,679,365	3,677,138.14	3,328,153.34	4,210,680.23	5,223,509	3,681,290.1	33,242,792
	27,800,239.85	33,242,792	4,990,377.92	3,679,365	3,677,138.14	3,328,153.34	4,210,680.23	5,223,509	3,681,290.1	33,242,792
	27,800,239.85	33,242,792	4,990,377.92	3,679,365	3,677,138.14	3,328,153.34	4,210,680.23	5,223,509	3,681,290.1	33,242,792
	27,800,239.85	33,242,792	4,990,377.92	3,679,365	3,677,138.14	3,328,153.34	4,210,680.23	5,223,509	3,681,290.1	33,242,792
	27,800,239.85	33,242,792	4,990,377.92	3,679,365	3,677,138.14	3,328,153.34	4,210,680.23	5,223,509	3,681,290.1	33,242,792
Facebook Prophet model	25,489,630.74	34,368,570	5,103,636.45	3,652,250	3,336,632.71	3,336,632.71	4,595,002.18	5,869,178	3,931,678.31	34,368,570
	25,647,356.8	34,429,330	5,112,575.85	3,670,628	3,349,399.23	3,349,399.23	4,611,381.59	5,912,395	3,948,639.15	34,429,330
	25,809,552.6	34,491,854	5,121,715.96	3,688,887	3,361,507.91	3,361,507.91	4,628,232.7	5,943,202	3,964,065.27	34,491,854
	25,971,872.92	34,548,965	5,130,710.05	3,705,250	3,373,982.73	3,373,982.73	4,644,341.77	5,973,329	3,978,063.45	34,548,965
	26,133,621.05	34,601,134	5,139,831.52	3,715,388	3,383,649.47	3,383,649.47	4,662,536.7	6,007,521	3,990,134.14	34,601,134
Stacked LSTM	27,894,800	33,213,357	4,995,613	3,680,159	3,342,567	3,342,567	4,213,055	5,223,499	3,684,672	33,213,357
	27,894,800	33,213,357	4,995,613	3,680,159	3,342,567	3,342,567	4,213,055	5,223,499	3,684,672	33,213,357
	27,894,800	33,213,357	4,995,613	3,680,159	3,342,567	3,342,567	4,213,055	5,223,499	3,684,672	33,213,357
	27,894,800	33,213,357	4,995,613	3,680,159	3,342,567	3,342,567	4,213,055	5,223,499	3,684,672	33,213,357
	27,894,800	33,213,357	4,995,613	3,680,159	3,342,567	3,342,567	4,213,055	5,223,499	3,684,672	33,213,357

Table 5 Prediction of death cases

Models	India	US	Russia	Argentina	Brazil	Colombia	Italy	Turkey	Germany	France
Random forest regressor	323,004.1	593,614.2	117,244.3	74,704.57	457,627.8	87,362.86	125,824.4	47,010.95	88,361.46	109,462.97
	323,004.1	593,614.2	117,244.3	74,704.57	457,627.8	87,362.86	125,824.4	47,010.95	88,361.46	109,462.97
	323,004.1	593,614.2	117,244.3	74,704.57	457,627.8	87,362.86	125,824.4	47,010.95	88,361.46	109,462.97
	323,004.1	593,614.2	117,244.3	74,704.57	457,627.8	87,362.86	125,824.4	47,010.95	88,361.46	109,462.97
	323,004.1	593,614.2	117,244.3	74,704.57	457,627.8	87,362.86	125,824.4	47,010.95	88,361.46	109,462.97
Facebook Prophet model	277,360.7	612,482.7	123,245.2	74,815.4	494,072.2	85,952.42	133,067.2	48,809.13	91,174.55	114,710.014
	278,779.8	613,708.3	123,639.8	75,126.5	496,726.4	86,254.16	133,421.9	49,029.8	91,422.54	114,970.418
	280,166.3	614,774.8	124,033.7	75,393.79	499,253.7	86,547.8	133,774.9	49,251.07	91,630.43	115,330.092
	281,549.9	615,607.7	124,418.3	75,596.27	501,646.2	86,842.44	134,103.1	49,470.37	91,758.49	115,511.485
	281,973.3	616,132.4	124,770.7	75,693.13	503,703.2	87,065.62	134,427.5	49,675.14	91,866.75	115,714.851
Stacked LSTM	325,972	593,606.5	102,855.5	75,056	457,808.8	77,694.1	125,410.2	46,721	88,413	109,304.4
	325,972	593,606.5	103,112.3	75,056	457,808.8	77,900.73	125,410.2	46,721	88,413	109,304.4
	325,972	593,606.5	103,369.2	75,056	457,808.8	78,107.36	125,410.2	46,721	88,413	109,304.4
	325,972	593,606.5	103,626	75,056	457,808.8	78,313.98	125,410.2	46,721	88,413	109,304.4
	325,972	593,606.5	103,882.9	75,056	457,808.8	78,520.61	125,410.2	46,721	88,413	109,304.4

Table 6 Prediction of recovered cases

Models	India	US	Russia	Argentina	Brazil	Colombia	Italy	Turkey	Germany	France
Random forest regressor	25,150,904	0	4,611,085	3,257,327	14,468,820	87,362.86	3,820,700	5,074,250	3,473,021	390,369.2
	25,150,904	0	4,611,085	3,257,327	14,468,820	87,362.86	3,820,700	5,074,250	3,473,021	390,369.2
	25,150,904	0	4,611,085	3,257,327	14,468,820	87,362.86	3,820,700	5,074,250	3,473,021	390,369.2
	25,150,904	0	4,611,085	3,257,327	14,468,820	87,362.86	3,820,700	5,074,250	3,473,021	390,369.2
	25,150,904	0	4,611,085	3,257,327	14,468,820	87,362.86	3,820,700	5,074,250	3,473,021	390,369.2
Facebook Prophet model	21,419,678	− 886,944	4,749,234	3,207,345	15,225,300	85,952.42	4,102,121	5,453,062	3,555,966	408,450.5
	21,540,505	− 887,276	4,759,648	3,222,251	15,283,819	86,254.16	4,119,468	5,481,622	3,569,605	410,053.8
	21,660,881	− 887,865	4,769,929	3,237,150	15,340,460	86,547.8	4,136,929	5,510,250	3,582,671	411,591.8
	21,783,507	− 886,061	4,779,956	3,252,218	15,394,384	86,842.44	4,153,016	5,555,886	3,594,548	412,709.5
	21,823,609	− 901,726	4,787,971	3,262,384	15,443,484	87,065.62	4,168,495	5,584,158	3,606,100	413,950.9
Stacked LSTM	25,150,904	1,064,335	4,616,422	3,288,467	14,471,076	77,694.1	2,935,741	3,696,329	3,453,918	390,369.2
	25,150,904	1,064,737	4,616,422	3,288,467	14,471,076	77,900.73	2,943,479	3,706,802	3,453,918	390,369.2
	25,150,904	1,065,139	4,616,422	3,288,467	14,471,076	78,107.36	2,951,218	3,717,274	3,453,918	390,369.2
	25,150,904	1,065,540	4,616,422	3,288,467	14,471,076	78,313.98	2,958,956	3,727,746	3,453,918	390,369.2
	25,150,904	1,065,942	4,616,422	3,288,467	14,471,076	78,520.61	2,966,694	3,738,219	3,453,918	390,369.2

Table 7 Comparison with the existing techniques

References	Dataset	Techniques	Mean R^2 values (%)
[45]	Real time dataset	Regression, cloud computing	92.2
[46]	Data collected from Our World in Data	Machine learning, cloud computing	98
[47]	Data collected from Saudi ministry of health	Non linear autoregressive artificial neural networks	98.7
[48]	WHO’s official data	Adaptive network based fuzzy interference system	97.63
[49]	Data collected from January 23, 2020 to June 17 2020	Random forest model	95.9
Our Study	Data collected from January 22, 2020 to May 29, 2021	Random forest regressor, Stacked LSTM	98.8

enslaved by the terrible COVID-19 sickness. Furthermore, an assessment of the complete economic failure in many sectors during the decrease of COVID-19 should be planned to assist countries in reviving their loss.

Funding Not applicable.

Availability of Data and Materials Not applicable.

Code Availability Not applicable.

Declarations

Conflicts of interest The authors declare no conflict of interest.

Ethical approval All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki Declaration and its later amendments or comparable ethical standards.

Consent for participants Informed consent was obtained from all individual participants included in the study.

References

1. Shastri S, Singh K, Kumar S, Kour P, Mansotra V. Time series forecasting of COVID-19 using deep learning models: India-USA comparative case study. Chaos Solit Fract. 2020;140: 110227. https://doi.org/10.1016/j.chaos.2020.110227.
2. Papastefanopoulos V, Linardatos P, Kotsiantis S. COVID-19: a comparison of time series methods to forecast percentage of active cases per population. Appl Sci (Switzerland). 2020;10(11):1–15. https://doi.org/10.3390/app10113880.
3. Chimmula VKR, Zhang L. Time series forecasting of COVID-19 transmission in Canada using LSTM networks. Chaos Solit Fract. 2020. https://doi.org/10.1016/j.chaos.2020.109864.
4. Toğaçar M, Ergen B, Cömert Z. COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches. Comput Biol Med. 2020. https://doi.org/10.1016/j.compbiomed.2020.103805.
5. Arshadi A, Webb J, Salem M, Cruz E, Calad-Thomson S, Ghadirian N, Collins J, Diez-Cecilia E, Kelly B, Goodarzi H, Yuan JS. Artificial intelligence for covid-19 drug discovery and vaccine development. Front Artif Intell. 2020;3(August):1–13. https://doi.org/10.3389/frai.2020.00065.
6. Elaziz A, Hosny M, Salah A, Darwish MM, Lu S, Sahlol AT. New machine learning method for image based diagnosis of COVID-19. PLoS ONE. 2020. https://doi.org/10.1371/journal.pone.0235187.
7. Alimadadi A, Aryal S, Manandhar I, Munroe PB, Joe B, Cheng X. Artificial intelligence and machine learning to fight COVID-19. Physiol Genom. 2020;52(4):200–2. https://doi.org/10.1152/physiolgenomics.00029.2020.
8. Alazab M, Awajan A, Mesleh A, Abraham A, Jatana V, Alhyari S. COVID-19 prediction and detection using deep learning. Int J Comput Inf Syst Ind Manag Appl. 2020;12(April):168–81.
9. Alakus TB, Turkoglu I. Comparison of deep learning approaches to predict COVID-19 infection. Chaos Solit Fract. 2020;140: 110120. https://doi.org/10.1016/j.chaos.2020.110120.
10. Punn NS, Sonbhadra SK, Agarwal S. COVID-19 epidemic analysis using machine learning and deep learning algorithms. MedRxiv. 2020. https://doi.org/10.1101/2020.04.08.20057679.
11. Wang S, Kang B, Ma J, Zeng X, Xiao M, Guo J, Cai M, Yang J, Li Y, Meng X, Xu B. A deep learning algorithm using CT images to screen for Corona virus disease (COVID-19). Eur Radiol. 2021. https://doi.org/10.1007/s00330-021-07715-1.
12. Bandyopadhyay D, Akhtar T, Hajra A, et al. COVID-19 pandemic: cardiovascular complications and future implications. Am J Cardiovasc Drugs. 2020;20:311–24. https://doi.org/10.1007/s40256-020-00420-2.
13. Ghoshal B, Tucker A. Estimating uncertainty and interpretability in deep learning for coronavirus (COVID-19) detection. 2020. pp. 1–14. http://arxiv.org/abs/2003.10769
14. Ismael AM, Şengür A. Deep learning approaches for COVID-19 detection based on chest X-ray images. Expert Syst Appl. 2021;164: 114054. https://doi.org/10.1016/j.eswa.2020.114054.
15. Panwar H, Gupta PK, Siddiqui MK, Morales-Menendez R, Singh V. Application of deep learning for fast detection of COVID-19 in X-rays using nCOVnet. Chaos Solit Fract. 2020;138: 109944. https://doi.org/10.1016/j.chaos.2020.109944.
16. Muhammad LJ, Islam MM, Usman SS, et al. Predictive data mining models for novel coronavirus (COVID-19) infected patients' recovery. SN Comput Sci. 2020;1:206. https://doi.org/10.1007/s42979-020-00216-w.
17. Wang S, Zha Y, Li W, Wu Q, Li X, Niu M, Wang M, Qiu X, Li H, Yu H, Gong W, Bai Y, Li L, Zhu Y, Wang L, Tian J. A fully automatic deep learning system for COVID-19 diagnostic and prognostic analysis. Eur Respir J. 2020;56(2):2000775. https://doi.org/10.1183/13993003.00775-2020.
18. Tamhane R, Mulge S. Prediction of COVID-19 outbreak using machine learning. Int Res J Eng Technol. 2020;7(5):5699–702.
19. Pajankar A. Data visualization with numpy and matplotlib. In: Practical python data visualization. Berkeley: Apress; 2021. https://doi.org/10.1007/978-1-4842-6455-3_5.
20. Waskom M. Seaborn: statistical data visualization. J Open Source Softw. 2021;6:1–4.
21. Chumachenko D, Chumachenko T, Meniailov I, Pyrohov P, Kuzin I, Rodyna R. On-seasline data processing, simulation and forecasting of the coronavirus die (COVID-19) propagation in ukraine based on machine learning approach. In: Babichev S, Peleshko D, Vynokurova O, editors. Data stream mining & processing. DSMP 2020. Communications in computer and information science, vol. 1158. Cham: Springer; 2020. https://doi.org/10.1007/978-3-030-61656-4_25.
22. Singh M, Jakhar AK, Pandey S. Sentiment analysis on the impact of coronavirus in social life using the BERT model. Soc Netw Anal Min. 2021;11:33. https://doi.org/10.1007/s13278-021-00737-z.
23. Varoquaux G, Buitinck L, Louppe G, Grisel O, Pedregosa F, Mueller A. Scikit-learn: machine learning without learning the machinery. GetMobile Mob Comput Commun. 2015;19(1):29–33. https://doi.org/10.1145/2786984.2786995.
24. Yadav D, Maheshwari H, Chandra U, Sharma A. COVID-19 analysis by using machine and deep learning. In: Chakraborty C, Banerjee A, Garg L, Rodrigues JJPC, editors. Internet of medical things for smart healthcare studies in big data, vol. 80. Singapore: Springer; 2020. https://doi.org/10.1007/978-981-15-8097-0_2.
25. Khakharia A, Shah V, Jain S, et al. Outbreak prediction of COVID-19 for dense and populated countries using machine learning. Ann Data Sci. 2021;8:1–19. https://doi.org/10.1007/s40745-020-00314-9.
26. Albanese D, Visintainer R, Merler S, Riccadonna S, Jurman G, Furlanello C. mlpy: machine learning python. Math Soft. 2012;1–4.
27. Bologheanu R, Maleczek M, Laxar D, et al. Outcomes of non-COVID-19 critically ill patients during the COVID-19

pandemic. Wien Klin Wochenschr. 2021. https://doi.org/10.1007/s00508-021-01857-4.

28. Hancock JT, Khoshgoftaar TM. CatBoost for big data: an interdisciplinary review. J Big Data. 2020;7:94. https://doi.org/10.1186/s40537-020-00369-8.
29. Kairon P, Bhattacharyya S. COVID-19 outbreak prediction using quantum neural networks. In: Bhattacharyya S, Dutta P, Datta K, editors. Intelligence enabled research. Advances in intelligent systems and computing, vol. 1279. Singapore: Springer; 2021. https://doi.org/10.1007/978-981-15-9290-4_12.
30. Consonni M, Telesca A, Dalla Bella E, et al. Amyotrophic lateral sclerosis patients' and caregivers' distress and loneliness during COVID-19 lockdown. J Neurol. 2021;268:420–3. https://doi.org/10.1007/s00415-020-10080-6.
31. Brinati D, Campagner A, Ferrari D, et al. Detection of COVID-19 infection from routine blood exams with machine learning: a feasibility study. J Med Syst. 2020;44:135. https://doi.org/10.1007/s10916-020-01597-4.
32. Khanday AMUD, Rabani ST, Khan QR, et al. Machine learning based approaches for detecting COVID-19 using clinical text data. Int J Inf Tecnol. 2020;12:731–9. https://doi.org/10.1007/s41870-020-00495-9.
33. Kwekha-Rashid AS, Abduljabbar HN, Alhayani B. Coronavirus disease (COVID-19) cases analysis using machine-learning applications. Appl Nanosci. 2021. https://doi.org/10.1007/s13204-021-01868-7.
34. Ebner L, Funke-Chambour M, von Garnier C, et al. Imaging in the aftermath of COVID-19: what to expect. Eur Radiol. 2021;31:4390–2. https://doi.org/10.1007/s00330-020-07465-6.
35. Ma Z, Li H, Fang W, Liu Q, Zhou B, Bu Z. A cloud-edge-terminal collaborative system for temperature measurement in COVID-19 prevention. In: IEEE INFOCOM 2021—IEEE conference on computer communications workshops (INFOCOM WKSHPS), 2021, pp. 1–6. https://doi.org/10.1109/INFOCOMWKSHPS51825.2021.9484616.
36. Senapati A, Nag A, Mondal A, et al. A novel framework for COVID-19 case prediction through piecewise regression in India. Int J Inf Tecnol. 2021;13:41–8. https://doi.org/10.1007/s41870-020-00552-3.
37. Bhardwaj P, Bhandari G, Kumar Y, et al. An investigational approach for the prediction of gastric cancer using artificial intelligence techniques: a systematic review. Arch Computat Methods Eng. 2022. https://doi.org/10.1007/s11831-022-09737-438.
38. Kumar Y, Patel NP, Koul A, Gupta A. Early prediction of neonatal jaundice using artificial intelligence techniques. In: 2nd International conference on innovative practices in technology and management (ICIPTM). 2022. pp. 222–226. https://doi.org/10.1109/ICIPTM54933.2022.9753884.
39. Gupta A, Koul A, Kumar Y. Pancreatic cancer detection using machine and deep learning techniques. In: 2nd International conference on innovative practices in technology and management (ICIPTM), 2022, pp. 151–155. https://doi.org/10.1109/ICIPTM54933.2022.9754010.
40. Shoaib M, Salahudin H, Hammad M, et al. Performance evaluation of soft computing approaches for forecasting COVID-19 pandemic cases. Sn Comput Sci. 2021;2:372. https://doi.org/10.1007/s42979-021-00764-9.
41. Kumar Y, Gupta S, Gupta A. Study of machine and deep learning classifications for IOT enabled healthcare devices. In: International Conference on Technological Advancements and Innovations (ICTAI). 2021. pp. 212–217. https://doi.org/10.1109/ICTAI53825.2021.9673437.
42. Kohli R, Garg A, Phutela S, Kumar Y, Jain S. An improvised model for securing cloud-based E-healthcare systems. In: Marques G, Bhoi AK, Albuquerque VHCD, Hareesha KS, editors. IoT in healthcare and ambient assisted living studies in computational intelligence, vol. 933. Singapore: Springer; 2021. https://doi.org/10.1007/978-981-15-9897-5_14.
43. Kumar Y, Gupta S. Deep transfer learning approaches to predict glaucoma, cataract, choroidal neovascularization, diabetic macular edema, drusen and healthy eyes: an experimental review. Arch Computat Methods Eng. 2022. https://doi.org/10.1007/s11831-022-09807-7.
44. Singh H, Bawa S. Predicting COVID-19 statistics using machine learning regression model: Li-MuLi-Poly. Multimedia Syst. 2021. https://doi.org/10.1007/s00530-021-00798-2.
45. Andreas A, Mavromoustakis CX, Mastorakis G, Mumtaz S, Batalla JM, Pallis E. Modified machine learning Techique for curve fitting on regression models for COVID-19 projections. In: 2020 IEEE 25th international workshop on computer aided modeling and design of communication links and networks (CAMAD). 2020. IEEE. pp. 1–6.
46. Tuli S, Tuli S, Tuli R, Gill SS. Predicting the growth and trend of COVID-19 pandemic using machine learning and cloud computing. Internet Things. 2020;11: 100222.
47. Elsheikh AH, Saba AI, Abd Elaziz M, Lu S, Shanmugan S, Muthuramalingam T, et al. Deep learning-based forecasting model for COVID-19 outbreak in Saudi Arabia. Process Saf Environ Prot. 2021;149:223–33.
48. Zivkovic M, Bacanin N, Venkatachalam K, Nayyar A, Djordjevic A, Strumberger I, Al-Turjman F. COVID-19 cases prediction by using hybrid machine learning and beetle antennae search approach. Sustain Cit Soc. 2021;66: 102669.
49. Yeşilkanat CM. Spatio-temporal estimation of the daily cases of COVID-19 in worldwide using random forest machine learning algorithm. Chaos Solit Fract. 2020;140: 110210.

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