

Outbreak prediction of COVID-19 for dense and populated countries using machine learning

Received: date / Accepted: date

Abstract The Coronavirus Disease-2019 (COVID-19) pandemic continues to have a devastating effect on the health and well-being of the global population. A continued rise in the number of patients testing positive for COVID-19 has created a lot of stress on governing bodies across the globe and they are finding it difficult to tackle the situation. We have developed an outbreak prediction system for COVID-19 for the top 10 highly and densely populated countries. The proposed prediction models forecast the count of new cases likely to arise for successive 5 days using 9 different machine learning algorithms. A set of models for predicting the rise in new cases, having an average accuracy of $87.9\% \pm 3.9\%$ was developed for 10 high population and high density countries. The highest accuracy of 99.93% was achieved for Ethiopia using Auto-Regressive Moving Average (ARMA) averaged over the next 5 days. The proposed prediction models used by us can help stakeholders to be prepared in advance for any sudden rise in outbreak to ensure optimal management of available resources.

Keywords COVID-19 outbreak prediction · COVID-19 · Machine learning

1 Introduction

The SARS-CoV-2 coronavirus disease (COVID-19) disease originated in Wuhan, China sometime during December 2019. Within a month, more than ten thousand people were infected and hundreds died [1]. The initial outbreak caused several deaths, as the medical systems were not capable of handling many seriously ill patients. Till July 23, 2020 there were 631,680 deaths [2] reported across the world due to this pandemic. In a rapidly evolving pandemic, improper analysis and predictions of the number of patients results in an inefficient distribution of medical resources. Limited medical facilities and mismanagement of resource allocation can lead to additional severe cases and a decline

Address(es) of author(s) should be given

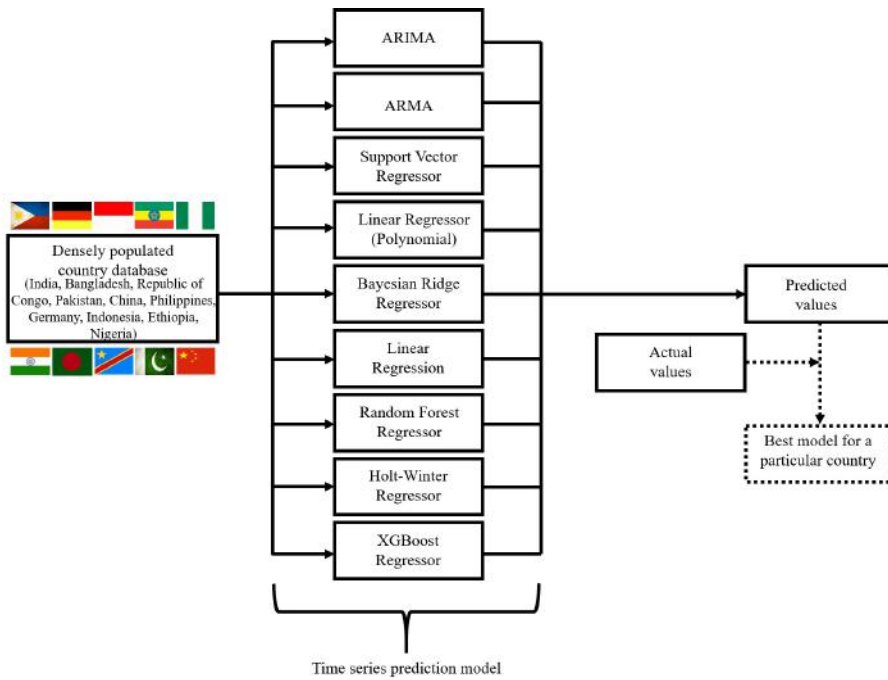


Fig. 1 Proposed system flow diagram. The data on the spread of COVID-19 in the top 10 densely populated countries, viz., India, Bangladesh, the Democratic Republic of Congo, Pakistan, China, Philippines, Germany, Indonesia, Ethiopia, and Nigeria were analyzed. The data for all the countries was fed into 9 different machine learning algorithms to predict the count of new cases for the next 5 days. These predicted values were compared with the actual values that were found and the accuracy was calculated. The best outbreak prediction model was selected for each country depending on the accuracy values obtained.

in recovery rates. To cope with this situation, predicting the new cases which will arise in the future is very important. This can ensure optimal allocation of medical resources in the affected regions.

Data science in the predictive domain is an emerging field. In this study we have incorporated the principles of data science [3] for the prediction of COVID-19 progression. The outbreak of COVID-19 seriously challenges every government with regard to the capacity and management of public health systems in facing the catastrophic emergency [4]. The prediction model can help hospitals and healthcare management to properly allocate resources, thereby reducing the pressure and allowing the situation to be handled with relative ease.

We developed and tested 9 different predictive algorithms for 10 countries. It was noticed that the pattern of growth in the number of cases varied from country to country. The basic approach for the predictions was to train the models based on the dataset provided, but these models were not sufficiently accurate, as they were trained on only one class of dataset. As a result, the models were unable to accurately predict the number of new cases and, conse-

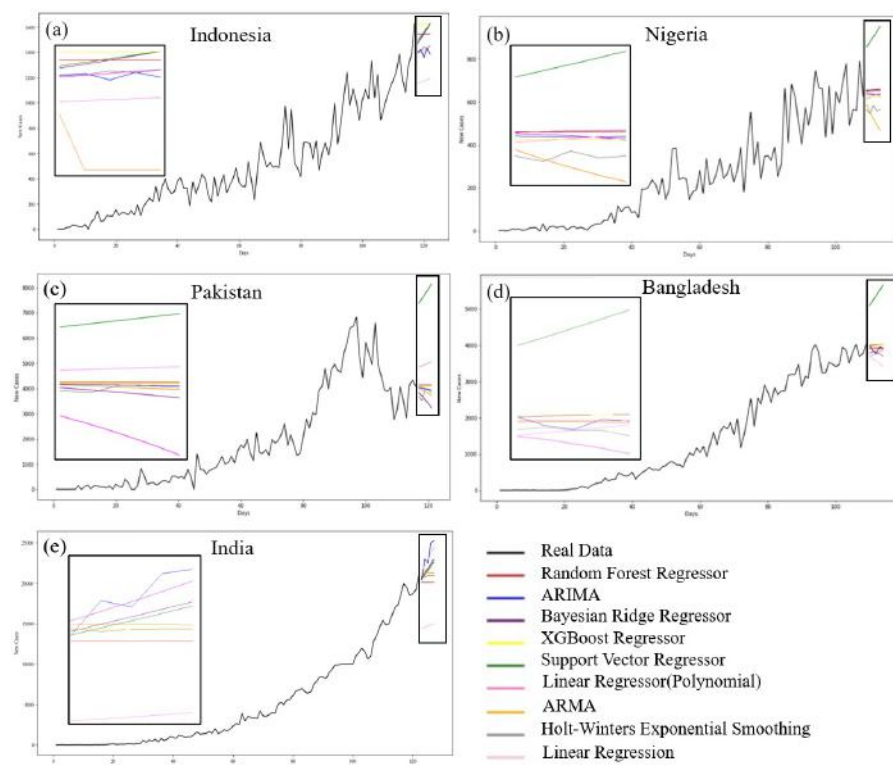


Fig. 2 Prediction plots for the number of COVID-19 patients that would rise in the next 5 days for some countries where an exponential increase in the curve is expected or the rise in the cases would remain constant. Various machine learning models were deployed for predicting the outbreak. The black line shows the actual data, whereas the other colors represent the predictions obtained using the different ML algorithms. The SVR model is inefficient for most of the countries, whereas the ARIMA model gave comparatively better results. The predictions for the countries can be seen more clearly from the snippets. (a) The prediction plot for Indonesia indicates a rise in the curve as predicted by most of the models. ARIMA shows a decline in the cases, whereas the ARMA model indicates a rise in the curve. (b) Prediction plot for Nigeria. Apart from the ARMA model, all the other models predicted an increase in the curve. (c) Prediction plot for Pakistan. SVR indicates a sharp increase in the curve, whereas the other models show a constant rise in the number of cases. (d) Prediction plot for Bangladesh. All the models indicate a constant rise in the cases, whereas the SVR model shows an abrupt increase in the curve, indicating its inefficiency for predicting the outbreak. (e) Prediction plot for India. The cases in India will increase exponentially as predicted by the models, whereas the LRP model predicted the decline for India.

quently, the existing techniques failed to utilize the resources in an optimized way [5]. Insufficient training data is also one of the reasons for the models to have low accuracy.

We tried 9 different standard machine learning (ML) algorithms for predicting the number of patients for the next 5 days. After getting a decent accuracy of 85 %, we implemented these algorithms on datasets of different

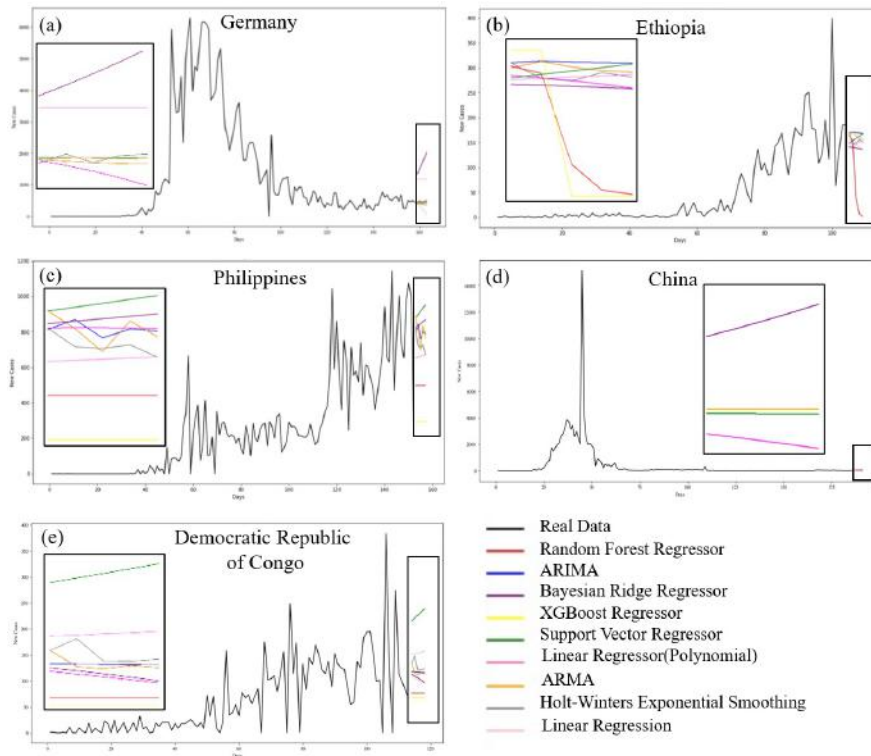


Fig. 3 Prediction for the next 5 days of the number of patients in different countries where cases are likely to decrease in the coming days using 9 different machine learning algorithms. The black line represents the real data obtained and the rest all colors show predictions using different ML models. The predictions for the countries can be seen more clearly from the snippets. (a) Prediction plot for Germany. The ARIMA and ARMA models indicate that the count will remain constant for the coming days, whereas XGB shows a decrease in the cases. (b) Prediction plot for Ethiopia. RGB shows a rapid decline in the number of cases, whereas the ARMA model shows a slight decrease in the curve. (c) Prediction plot for the Philippines. All the algorithms were inefficient in predicting the highly uneven number of cases seen in the country. (d) Prediction plot for China. Training the dataset with some specific values, a few algorithms such as LRP and BRR gave inappropriate results (e) Prediction plot for the Democratic Republic of Congo. SVR and LRP show an increase in the number of cases.

countries. We selected 10 countries with the highest population and the highest density for our work. By using the data of these countries, we trained standard prediction models using multiple ML algorithms and obtained different accuracy for each of the models for different countries. The different models gave high accuracy for different countries. However, there were variations in accuracy because of the different trends of change in COVID-19 patients for different countries.

The system flow diagram is shown in Figure 1. The 9 different machine learning (ML) algorithms used were Auto-Regressive Moving Average (ARMA), Auto-Regressive Integrated Moving Average (ARIMA), Support Vector Re-

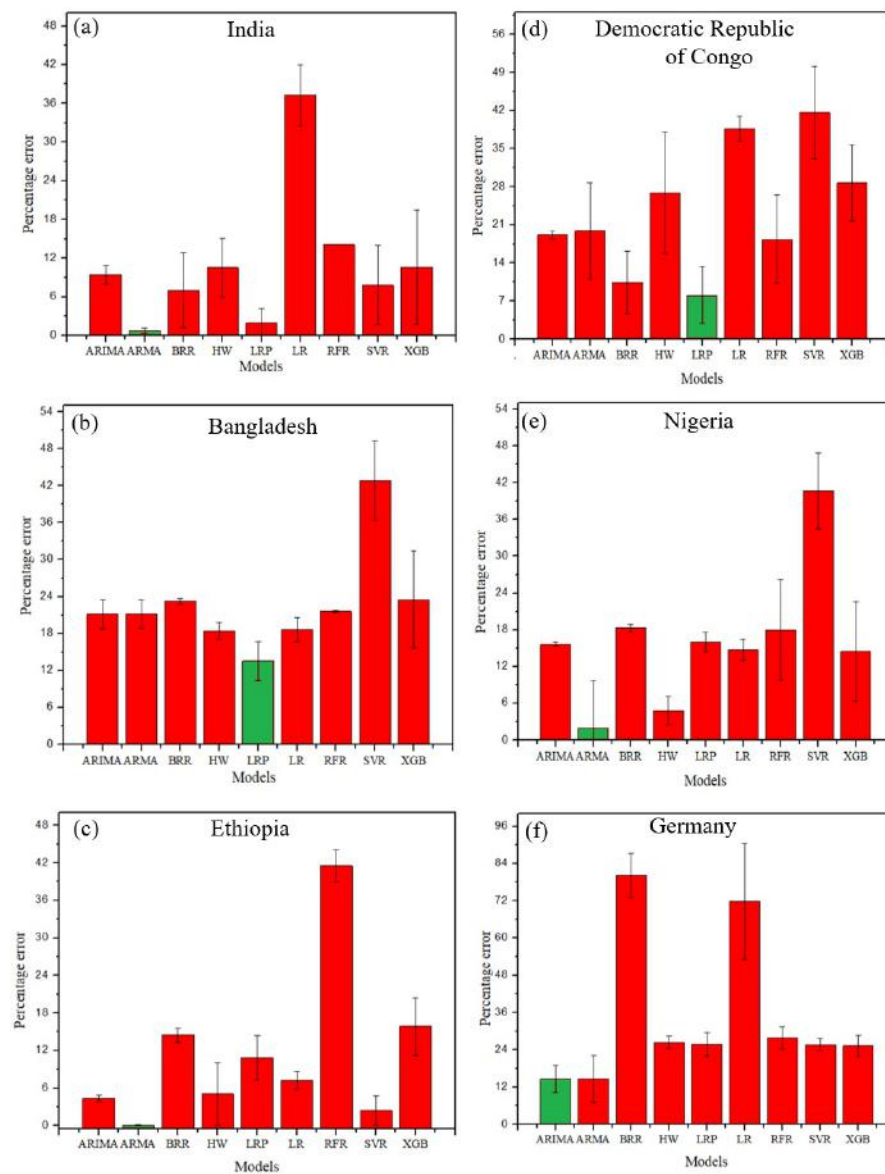


Fig. 4 Bar graphs depicting the error percentage and error bar for 5-day prediction by using 9 different ML models. The green color bar indicates the model with the least percentage error, i.e., highest percentage accuracy. (a) For India, the ARMA model gave the highest accuracy with an error bar of 0.42. (b) For Bangladesh, the LRP model gave the highest accuracy with an error bar of 3.18 as compared to other models. (c) The ARMA model showed the least percentage error in comparison to other models for Ethiopia. (d) The LRP model gave the least percentage error for the Democratic Republic of Congo. (e) In the case of Nigeria, the ARMA model gave the least percentage error, although the error bar had a value of 7.76. (f) For Germany, the ARIMA model gave the least percentage error, while models like BRR and LR gave percentage error of more than 50 %

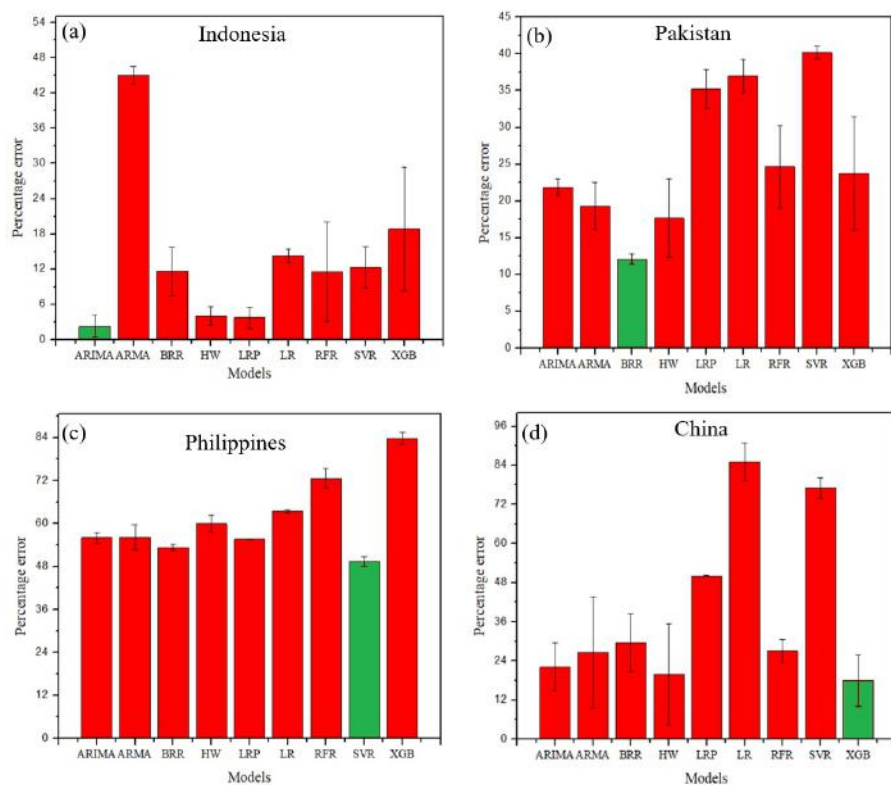


Fig. 5 Bar Graphs depicting the error percentage and error bar for 5 days prediction using nine different ML models. The green color bar indicates the model with the least percentage error i.e. highest percentage accuracy. (a) For Indonesia, the ARIMA model gave the least percentage error whereas it was seen that ARMA gave the highest percentage error as compared to other models. (b) For Pakistan, the BRR model proved best for prediction. (c) For the Philippines, none of the models gave accuracy that was expected. The percentage error of all models was more than 40 %. (d) For China, the XGB model proved the best for prediction while models like LR and SVR gave an error percentage of more than 70 %.

gressor (SVR), Linear Regressor (polynomial) (LRP), Bayesian Ridge Regression (BRR), Linear Regression (LR), Random Forest Regressor (RFG), Holt-Winter Exponential Smoothing (HW), and Extreme gradient Boost Regressor (XGB).

2 Literature review

Multiple research works have been carried out to predict the outbreak of COVID-19. On the dataset of patients from STEMI, different classifiers used for predictions were, namely, Logistic Regression, LogitBoost, Decision Tree, NBC, Neural Networks, and the two versions of Bayesian Network Classifiers [6]. Kumar *et al.* [7] used the ARIMA model for predicting the outbreak in

the top 15 European countries. Tuli *et al.* [8] proposed an ML model that can run continuously on Cloud Data Centers (CDCs) for accurate spread prediction and proactive development of strategic response by the government and citizens. Robust Weibull models fitted well on their dataset rather than baseline Gaussian models. Petropoulos *et al.* [9] introduced an objective approach to predict continuation of COVID-19 by live forecasting. They produce ten-days-ahead point forecasts and prediction intervals. A susceptible-exposed-infectious-recovered (SEIR) metapopulation model was used to simulate the epidemics across all major cities in China, with 95 % credible intervals [10]. Yang *et al.* [11] used the modified SEIR model to derive the epidemic curve. They used an artificial intelligence (AI) approach, trained on the 2003 SARS data, to predict the epidemic. Bhatnagar *et al.* [12] created a mathematical model for predicting the spread of COVID-19 in countries using various types of parameters and tested their model on real data of countries.

A segmented Poisson model was incorporated by the power law and the exponential law as proposed by Zhang *et al.* [13] to study the COVID-19 outbreaks in six major western countries. Maier *et al.* [14] have introduced a parsimonious model that captures the infected individuals and also population-wide isolation practices in response to containment policies. Li *et al.* [15] studied the transmission process of COVID-19. It used forward prediction and backward inference of the epidemic situation, and the relevant analysis helped relevant countries to make more appropriate decisions. Tomar *et al.* [16] have used data-driven estimation methods like long short-term memory (LSTM) and curve fitting for prediction of the number of COVID-19 cases in India 30 days in advance and the effect of preventive measures like social isolation and lockdown on the spread of COVID-19. Kumar *et al.* [17] have applied cluster analysis, to classify real groups of infectious disease of COVID-19 on a data set of different states and union territories in India, based on their high similarity to each other.

Wu *et al.* [10] forecasted the prediction for only the major cities of China, whereas Zhang *et al.* [13] predicted for six major western countries. On the other hand, the proposed methods forecast the count for 10 highly and densely populated countries. SEIR, Poisson, ARIMA, and exponential smoothing model were reported for COVID-19 count prediction. However, we have incorporated 9 different ML algorithms for the prediction and also trained our models with the data of over 100 days, which was 3 times more than reported in the literature.

3 Methodology

Since COVID-19 is spreading rapidly, we considered the top 10 countries with high population and high density for our outbreak prediction system. Since COVID-19 spreads majorly through human contact, it was imperative to consider only those countries with high density as well as high population. The dataset of the countries, namely, Bangladesh, India, China, Pakistan, Ger-

| Countries | Density (km2) | Population |
|------------------------------|---------------|------------|
| Bangladesh | 1116 | 164689383 |
| India | 420 | 1380004385 |
| China | 148 | 1439323776 |
| Pakistan | 250 | 220892340 |
| Germany | 235 | 83783942 |
| Nigeria | 223 | 206139589 |
| Ethiopia | 104 | 114963588 |
| Democratic Republic of Congo | 38 | 89561403 |
| Philippines | 320 | 109581078 |
| Indonesia | 144 | 273523615 |

Table 1 Top countries on the basis of population and density

many, Nigeria, Ethiopia, Democratic Republic of Congo, the Philippines, and Indonesia have been used. Initially, we identified a list of 20 most populated countries (Supplementary material S.10). Further, we obtained a list of countries with the highest population density (Supplementary material S.11). From these two lists, we identified the top 10 countries having the highest density as well as high population count (Table 1). We used 9 different machine learning algorithms for predicting the number of patients for the above-specified countries.

The train data to test data partition was 94 % and 6 %, respectively. The algorithms predicted the rise in the number of cases in the next 5 days for the countries specified in Table 1. The testing run-time for these algorithms varied between 2 to 5 seconds. The dataset was tuned by an iterative approach between the normalized value of zero and one. For the tuning of individual parameters, partial and full auto-correlation was used.

3.1 Auto-Regressive Moving Average (ARMA)

The ARMA model is simply the merger between AR(p) and MA(q) models, namely: the AR(p) model, which tries to explain the momentum and mean reversion effects often observed in trading markets (market participant effects) and the MA(q) model, which tries to capture the shock effects observed in white noise terms. These shock effects could be thought of as unexpected events affecting the observation process, e.g. surprise earnings, wars, attacks, etc. So first we loaded the dataset, and then divided it into a test set and a train set. We trained the model based on the train set and test set comprised of values for which we had to make predictions. Then we made an ARMA model that was trained on the training data. In ARMA, the values of p and q were put inside the order of the model. These values changed depending on what the model fitted the best. Values of p and q are normally taken up to 6. The values of p and q varied for different training datasets depending on the best fit.

Suppose a series a_0, a_1, \dots, a_t . To implement ARMA, we have to find the difference between data at different timestamps and make a new series alto-

gether. This difference that we take from the d parameter of the model. Let us represent the new time series as z_0, z_1, \dots, z_t . The newly formed time series is stationary and represented as $z_t = a_{(t+1)} - a_t$. Usually the value of d is taken as 0 or 1. The last value of the z series will be given by:

$$z_t = \sum_{i=0}^p (\phi_i z_{t-i}) + \sum_{j=0}^q (\theta_j \varepsilon_{t-j}) + \varepsilon_t \quad (1)$$

Now if we want to predict the value at k^{th} position, in future ie $k > t$, we have to get the answer in the original series means we need a value of a_k so we have to thus convert the z series into a series and it will be done as:

$$a_k = z_{k-1} + a_{k-1} = z_{k-2} + a_{k-2} = \sum_{i=1}^{k-l} z_{k-i} + a_l \quad (2)$$

3.2 Auto Regressive Integrated Moving Average (ARIMA)

ARIMA is a class of models that explains a given time series based on its past values, that is, its lags and the lagged forecast errors, so that the equation can be used to forecast future values. An ARIMA model is characterized by 3 terms: p , d , and q , where p is the order of the AR term, q is the order of the MA term and d is the number of differences required to make the time series stationary. So first we loaded a dataset, then divided into a test set and a train set. We trained the model based on the train set and test set comprised of the values for which we had to make predictions. Then we made an ARIMA model that was trained on the training data. The values of p , q , and d were put inside the order of the model. These values changed depending on what the model fitted the best. Values of p and q are normally taken up to 6 and d varied between 0 and 1. The values of p , d , and q varied for different training datasets depending on the best fit.

If there is given a time series l_0, l_1, \dots, l_t and we want to predict the last term that is l_t , then let the predicted last term be represented as \hat{l}_t . The actual last term will be given by:

$$l_t = \beta_0 + \sum_{i=0}^p (\beta_i l_{t-i}) + \sum_{j=0}^q (\phi_j \varepsilon_{t-j}) + \varepsilon_t \quad (3)$$

where $\sum_{i=0}^p (\beta_i l_{t-i})$ is Auto Regressive term, $\sum_{j=0}^q (\phi_j \varepsilon_{t-j})$ is Moving Average term and ε_t is Error lag. Now for predicting \hat{l}_t ,

$$\hat{l}_t = \beta_0 + \sum_{i=0}^p (\beta_i l_{t-i}) + \sum_{j=0}^q (\phi_j \varepsilon_{t-j}) \quad (4)$$

Only the Error lag term is not present. The values of p and q are determined by ACF and PACF, where ACF stands for Auto Correlation Function and PACF stands for Partial Auto-Correlation function.

3.3 Linear Regression (LR)

It is a statistical approach for modeling the relationship between a dependent variable and a given set of independent variables. All the values in the dataset were plotted. After plotting the points, we created the best-fit line. A best-fit line is the one that minimizes the error, i.e., it should have a minimum difference between the actual and predicted values. We found the slope of the line and also its y-intercept. After getting the equation of the line, we were able to predict the new values, which is the number of patients in an individual country. The expression for representing a line is given as $y=mx + c$, where 'm' is the slope. The formula for calculating the slope is

$$m = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sum(x - \bar{x})^2} \quad (5)$$

where \bar{x} and \bar{y} are the mean values.

3.4 Linear Regressor Polynomial (LRP)

We used the polynomial feature function provided by sci-kit learn library of machine learning, where we can increase the power of the input variable and then fit and transform it on any desired model. Firstly, we imported the necessary libraries. We then imported the polynomial and linear regression functions from sci-kit learn. We instantiated a polynomial feature function with degree=5 as a parameter. Then we fitted and transformed the input variable as well as the list of days for which we wanted to make the predictions. After that, we instantiated the linear regression model with parameters normalize=True, and fitintercept=False and then fitted the model using a new list made by applying polynomial features. Now we can use our model for predictions of COVID-19 cases on any particular day by using the list made by applying polynomial features to the list of days for which a prediction of COVID-19 cases is desired. Polynomial regression is a model based on a mixture of dependent and independent variables represented by m and y, respectively, and F_p is the polynomial function that tries to add variables of any power we need, which gives us the best results with the dataset taken.

$$y = \alpha_0 + \alpha_1 x^2 + \alpha_3 x^3 + \dots \alpha_n x^n \quad (6)$$

$$y = \alpha_0 + \sum_{i=1}^m \alpha_i x_i + F_p \quad (7)$$

Where,

m = number of independent variables,

y = dependent variable,

F_p = Polynomial function (It tries to add variables of any power we need)

3.5 Bayesian Ridge Polynomial Regressor (BRR)

Bayesian linear regression is an approach to linear regression in which the statistical analysis is undertaken within the context of Bayesian inference. Here also we have used a polynomial version of Bayesian Ridge Regressor to expose the important relationship between the input variables and target variables, which can further be used for prediction of COVID-19 cases on any random day. We used a randomized search that needs dictionaries of parameters having a different range of values as list of values and names of parameters as keys that we need to experiment with in our model and see which set of parameters gives the best results.

The model is defined with the parameters ω , α , and λ that are estimated jointly during the fit of the model, with the regularization parameters and λ being estimated by maximizing the log marginal likelihood. The initial value of the maximization procedure can be set with the hyperparameters α_{init} and λ_{init} . There are four more hyperparameters α_1 , α_2 , λ_1 , λ_2 , of the gamma prior distributions over α and λ . These are usually chosen to be non-informative.

Bayesian Ridge estimates a probabilistic model of the regression problem as described above. The prior for the coefficient ω is given by a spherical Gaussian:

$$p(\omega|\lambda) = N(\omega|0, \lambda^{-1}I_p) \quad (8)$$

The priors over α and λ are chosen to be gamma distributions, the conjugate prior for the precision of the Gaussian. The resulting model is called Bayesian Ridge Regression.

3.6 Support Vector Regressor (SVR)

SVR is a powerful algorithm that allows us to choose how tolerant we are of errors, both through an acceptable error margin(ϵ) and through tuning our tolerance of falling outside that acceptable error rate. Our original training dataset for every country was stated in a finite-dimensional state and so the sets to discriminate were not linearly separable in that space. To resolve this problem, our original finite-dimensional state was mapped into a higher-dimensional space. By doing this we could find the prediction of different countries in a non-linear approach. The model is defined as a comprehensive evaluation of the gram matrix along with the predictors $x(i)$ and $x(j)$. The gram matrix is a $n \times n$ dimensional matrix that contains the elements $g(i,j)$. The process comprises obtaining a non-linear SVM regression model by replacing the dot product of the predictors with a nonlinear kernel function comprising $G(x_1, x_2)$ as $\phi(x_1)$ and $\phi(x_2)$, where $\phi(x_1)$ comes out to be greater than $G(x_1, x_2)$ and $\phi(x_2)$ comes out to be less than the function modeled.

Some regression problems cannot be described adequately using a linear model. In such a case, the Lagrange dual formulation allows the previously-described technique to be extended to nonlinear functions (Primal formula). Obtain a nonlinear SVM regression model by replacing the dot product $x_1 x_2$

with a nonlinear kernel function $G(x_1, x_2) = \langle \varphi(x_1), \varphi(x_2) \rangle$, where $\varphi(x)$ is a transformation that maps x to a high-dimensional space. Statistics and Machine Learning Toolbox provides the following built-in semi-definite kernel functions.

The Kernel function of linear dot product is as shown:

$$G(x_j, x_k) = x_j' x_k \quad (9)$$

The Kernel function of Gaussian is:

$$G(x_j, x_k) = \exp(-|(x_i - x_k)^2|) \quad (10)$$

The Kernel function of Polynomial is:

$$G(x_j, x_k) = (1 + x_j' x_k)^q \quad (11)$$

, where q is in $\{2, 3, \dots\}$.

3.7 Random Forest Regressor (RFR)

We used RFR, as it fits several classifying decision trees on various sub-samples of our dataset and used averaging to improve the predictive accuracy and control over-fitting. The sub-sample size was controlled with the `maxsamples` parameter. We loaded the specific model into our training environment and initiated all the parameters to random values. We got the same result every time we ran the model on the given dataset. Then we fitted this model on the dataset so that we could easily predict the number of COVID-19 cases on any day using our trained model.

The Random forest regressor model comprises parameters such as the number of trees, the number of features represented by B and M , respectively. Here the values of B and M are less than or equal to the dimensional value d . $T(i)$ represents the tree at index i . The $tree(i)$ is constructed in such a way that at each node a random value from a subset of features is chosen considering splits on those features only.

$$D = ((x_1, y_1), \dots, (x_n, y_n)) \quad (12)$$

where,

D = Observed data point

The parameters are,

B = Number of trees,

M = Number of features,

x_i is d -dimensional vector,

$B, M \leq d$,

$T_i = treeT_i$

3.8 XGBoost Regressor (XGB)

XGBoost stands for "Extreme Gradient Boosting" and it is an implementation of gradient boosting trees algorithm. Firstly, we imported the necessary libraries and instantiated XGBoost with nestimators=1000 and fit the model. With this, we predicted the number of COVID-19 cases on any day we wanted. In these ways, we used this model to get the number of COVID-19 cases on any particular day using a dataset of actual COVID-19 cases used in training the model.

The model is defined as a comprehensive mix of training losses and regularization measures along with squared loss function summed up in an interval varying from 1 to n. The purpose of optimizing training loss is because of its assistance in predictive models, while regularization enhances the generalization of simpler models. Additive boosting is with γ and λ as hyperparameters. Approximation techniques such as Taylor approximation have been used in generating the model.

$$Obj = \sum_{i=1}^n \ell(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (13)$$

where

$$\sum_{i=1}^n \ell(y_i, \hat{y}_i) = TrainingLoss \quad (14)$$

$$\sum_{k=1}^K \Omega(f_k) = Regularization \quad (15)$$

$$L(\theta) = \sum (\hat{y}_i - y_i)^2 \quad (16)$$

where $L(\theta)$ is loss function and $\sum (\hat{y}_i - y_i)^2$ is the squared loss.

3.9 Holt-Winters Exponential Smoothing (HW)

For this model we considered the seasonality to be additive. This means that the forecasted value for each data element is the sum of the baseline, trend, and seasonality components. We use c to denote the frequency of the seasonality, i.e., the number of seasons in a particular period. With the additive method, the seasonal component is expressed in absolute terms in the scale of the observed series, and in the level equation the series is seasonally adjusted by subtracting the seasonal component. The value of periods also varies depending on the best fit and by analyzing the graph of training. So we loaded a dataset, and then divided it into a test set and a train set. We trained the model based on the train set and test set comprised of values for which we had to make predictions. Then we did exponential smoothing on the training dataset with seasonality as additive. This model consists of periods over which we want

exponential smoothing to take place. The value of periods varies depending on the best fit and by analyzing the graph of training.

The Holt-Winters seasonal method comprises the forecast equation and three smoothing equations, one for the level l_t , one for the trend b_t , and one for the seasonal component s_t , with corresponding smoothing parameters α, β and γ . Within each period, the seasonal component will add up to approximately zero, i.e., $\sum S_t = 0$ for a particular period. Mathematically, Holt Winters Additive Model is represented as:

Forecast = Estimated level + Trend + Seasonality at most recent time point

Series equation is represented as:

$$\hat{Y}_{t+h|t} = l_t + hb_t + S_{t+h-c(k+1)} \quad (17)$$

The series has Level (l_t), Trend (b_t) + *Seasonality*(s_t) with c seasons.

Level equation is represented as:

$$l_t = \alpha(y_t - S_{t-c}) + (1 - \alpha)(l_{t-1} + b_{t-1}) \quad (18)$$

The level equation shows a weighted average between the seasonally adjusted observation ($y_t - S_{t-c}$) and the non-seasonal forecast ($l_{t-1} + b_{t-1}$) for time t .

Trend equation is represented as:

$$b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} \quad (19)$$

The trend equation is identical to Holt's linear method.

Seasonality equation is represented as:

$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-c} \quad (20)$$

The seasonal equation shows a weighted average between the current seasonal index, ($y_t - l_{t-1} - b_{t-1}$), and the seasonal index of the same season c time periods ago. The values of α , β and γ usually range between 0 and 1

3.10 Hardware, Dataset and Software used

The models were trained on Windows 10 operating system with an 8th generation Intel i5 processor and 8 GB of RAM. The dataset was obtained from ourworldindata.org [18]. All the models were trained on Google Colaboratory, as well as Spyder using Python version 3.6.7 along with the assistance of libraries such as Numpy version 1.15, Matplotlib version 3.3.1, Pandas version 1.1.0, Scikit-learn version 0.23.1, XGBoost version 1.1.1, and Statsmodels version 0.10.2.

| | ARIMA | ARMA | BRR | HW | LRP | LR | RFR | SVR | XGB |
|-------------------------------------|-------------------------------|-------------------------------|-------------------------------|------------------|-------------------------------|------------------|-----------------|-------------------------------|-------------------------|
| India | 90.55 ± 1.46 | 99.26 ± 0.42 | 92.94 ± 5.78 | 89.45 ± 4.54 | 98.06 ± 2.21 | 62.75 ± 4.67 | 85.91 ± 3.41 | 92.16 ± 6.16 | 89.42 ± 8.9 |
| Bangladesh | 78.87 ± 2.33 | 78.82 ± 2.30 | 76.74 ± 0.50 | 81.59 ± 1.42 | 86.45 ± 3.18 | 81.36 ± 1.91 | 78.4 ± 0.14 | 57.16 ± 6.45 | 76.49 ± 7.9 |
| Ethiopia | 95.62 ± 0.53 | 99.93 ± 0.07 | 85.53 ± 1.12 | 94.94 ± 5.06 | 89.17 ± 3.53 | 92.74 ± 1.48 | 58.48 ± 2.6 | 97.59 ± 2.40 | 84.16 ± 4.6 |
| Indonesia | 97.72 ± 1.87 | 54.98 ± 1.53 | 88.38 ± 4.11 | 95.93 ± 1.55 | 96.26 ± 1.83 | 85.73 ± 1.10 | 88.44 ± 8.5 | 87.72 ± 3.57 | 81.17 ± 10.51 |
| Nigeria | 84.36 ± 0.35 | 98.06 ± 7.76 | 81.71 ± 0.55 | 95.17 ± 2.34 | 84.01 ± 1.63 | 85.27 ± 1.71 | 82.04 ± 8.2 | 59.35 ± 6.24 | 85.56 ± 8.1 |
| Pakistan | 78.16 ± 1.16 | 80.67 ± 3.19 | 87.91 ± 0.67 | 82.32 ± 5.32 | 64.78 ± 2.60 | 63.02 ± 2.26 | 75.35 ± 5.6 | 59.80 ± 0.86 | 76.25 ± 7.7 |
| Democratic Republic of Congo | 80.83 ± 0.67 | 80.08 ± 8.86 | 89.57 ± 5.82 | 73.07 ± 11.13 | 91.96 ± 5.22 | 61.38 ± 2.26 | 81.64 ± 8.1 | 58.37 ± 8.44 | 71.31 ± 7 |
| Germany | 85.39 ± 4.38 | 85.38 ± 7.46 | 19.85 ± 7.18 | 73.56 ± 7.93 | 74.24 ± 3.83 | 28.17 ± 18.64 | 72.11 ± 3.6 | 74.41 ± 1.97 | 74.77 ± 3.5 |
| China | 77.89 ± 7.29 | 73.45 ± 16.89 | 70.44 ± 8.93 | 80.10 ± 15.51 | 49.87 ± 0.092 | 14.94 ± 5.80 | 72.86 ± 3.5 | 22.97 ± 3.16 | 82 ± 8 |
| Philippines | 43.98 ± 1.48 | 43.89 ± 3.53 | 46.68 ± 0.86 | 40.04 ± 2.38 | 44.36 ± 0.09 | 36.51 ± 0.38 | 27.36 ± 2.7 | 50.54 ± 1.38 | 16.21 ± 1.6 |

Table 2 Accuracy values for all 10 countries using 9 ML algorithms

4 Results

As shown in Table 2, we used 9 different machine learning algorithms to predict the number of patients in 10 highly dense and populated countries. Among all the models for the various countries, we achieved the highest accuracy of 99.93 % for Ethiopia by using the ARMA model. ARIMA gave an accuracy of more than 85 % most of the time for almost all countries. Almost all the models gave an accuracy of more than 80 % at least for one of the 10 countries, except in the case of the Philippines.

We found different countries to have a different trend of increase or decrease in COVID-19 patients. Not every ML algorithm could give a very high accuracy for predicting the rise or fall in the cases for each country. Our results showed that for Bangladesh, the LRP model showed the highest accuracy of 86.45 %. For India, we got an accuracy of 99.26 % using the ARMA model. China had a prediction value of 82 % using the XGB model. For Pakistan, the accuracy was 87.91 % using the BRR model. For Nigeria, the accuracy was 98.06 % using the ARMA model. Democratic Republic of Congo showed the highest accuracy of 91.96 % by using the LRP model. Indonesia demonstrated the highest accuracy

| Method | Model Used | Accuracy (%) |
|---------------------------------|------------|--------------|
| Wu <i>et al.</i> [10] | SEIR | 95 |
| Chintalapudi <i>et al.</i> [19] | ARIMA | 93.75 |
| Poonia <i>et al.</i> [20] | ARIMA | 95 |
| Gupta <i>et al.</i> [21] | ARIMA | 70 |
| Our proposed method | ARMA | 99.93 |
| | ARIMA | 97.72 |
| | BRR | 92.94 |
| | HW | 95.93 |
| | LRP | 98.06 |
| | LR | 92.74 |
| | RFR | 88.44 |
| | SVR | 97.59 |
| | XGB | 89.42 |

Table 3 Comparison of methodologies reported in existing literature

of 97.72 % using the ARIMA model. For Germany, ARIMA gave an accuracy of 85.39 %. Using the SVR model, we got a prediction accuracy of 50.54 % for the Philippines.

Figure 4 shows bar graphs for different error percentages and their corresponding errors for the next 5-day predictions. In the case of India (Figure 4 (a)), Ethiopia (Figure 4 (c)), and Nigeria (Figure 4 (e)), the ARMA model gave the highest accuracy for the prediction as compared to the other models. For Bangladesh (Figure 4 (b)) and Democratic Republic of Congo (Figure 4 (d)), the LRP model proved to be effective, although the accuracy in the case of Bangladesh was low. In the case of Germany (Figure 4 (f)), the ARIMA model gave the least percentage error, while models like BRR and LR gave errors of more than 50 %. For Indonesia (Figure 5 (a)), the ARIMA model gave the least percentage error, while the ARMA model was highly inaccurate, with a very high error percentage. In the case of Pakistan (Figure 5 (b)), the BRR model yielded better accuracy, whereas for the Philippines (Figure 5 (c)), none of the models made accurate predictions. The percentage error of all models was more than 40 %. The XGB model proved best for prediction in the case of China (Figure 5 (d)). A range finder code was written that helps to improve accuracy. This code works on a range of predicted numbers from all 9 algorithms, rather than actual predictions by the individual algorithms. This combined approach helped us to improve accuracy by up to 8 %.

5 Discussions

It was not possible to get the results using all the 9 algorithms for each country as there were no specific trends observed. For the Philippines, we got a very low accuracy because a sudden drop of around 1,400 cases to 0 was seen and in the following day, the count increased by 4,500. Also because of the change in the government rules, the COVID-19 count of the country changed drastically. Due to this change, the proposed models were not able to make

predictions with high accuracy. According to our dataset for China, a particular day had approximately 2,000 patients and after that day the rise observed in the number of cases was approximately 13,000. The number of total patients was observed to be around 15,000. All of a sudden, the number dropped by 11,000 the next day. The declining phase started after the drop for about the next 100 days. Due to the peak value, our training dataset had to be changed. We considered only those values after the peak where a declining trend could be seen. To date, China shows a decline in the curve, and hence we considered only the decrement values. The slope for China is decreasing, and hence the values after the peak were considered. Also, after considering this, 2 out of the 9 models failed to show good accuracy for China. The raw data received from countries like China and Philippines were not correct, because the government policies changed on February 17, 2020 and July 6, 2020, respectively.

5.1 Comparison with other methods

As shown in Table 3, we have compared our methodology with the other methodologies reported in the literature. Most of the literature has used the ARIMA model for the outbreak prediction of COVID-19. We have used 9 different ML models for the prediction of COVID-19 on the top 10 densely and highly populated countries. We achieved the highest accuracy of 99.93 %, which was high as compared to the other methodologies reported. The highest accuracy was achieved by the ARMA model for Ethiopia. Poonia *et al.* [20] achieved the highest accuracy of 95 % for India using the ARIMA model forecasting. The ARIMA model for India achieved an accuracy of 90.55 %, which was high as compared to an accuracy of 70 % obtained by Gupta *et al.* [21] using the ARIMA model and Exponential smoothing. In comparison, the SEIR model implemented by Wu *et al.* [10] gave an accuracy of 95 % for prediction in Wuhan.

5.2 Future scope

Although the overall accuracy achieved was very good, we are still trying to implement prediction models using different algorithms that could give us higher accuracy. We are also planning to get a single standard model that can be used for any country, which may be a combination of different algorithms. We are planning to develop such ML algorithms that could give us an approximate duration of COVID-19 as a pandemic.

6 Conclusions

The study presented here outlines several technique of predicting new cases that would arise in a few days in the near future in any region during an expanding pandemic, so that there is the proper allocation of resources in those

regions for higher recovery rates. The ARMA model gave the highest accuracy for the prediction of COVID-19 cases for Ethiopia. From the results obtained on all the models for all countries, it was found that ARMA proved to be the best model for India and Nigeria. ARIMA was best for Indonesia and Germany. LRP for Bangladesh and Democratic Republic of Congo. And BRR, XGB, SVR proved best for Pakistan, China and the Philippines, respectively. We got an accuracy of more than 80 % for all the countries except the Philippines by any one of the 9 ML algorithms. The overall best model for the prediction was ARIMA. Generating high-accuracy prediction that could help in an optimized use of available resources along with pacing up the recovery graphs has been the main aim behind this exercise. These regions could potentially benefit from knowing the number of resources that they would need based on the predictions of the model. This model could help in lowering the cost of dealing with the pandemic and improve the recovery process in regions where it is deployed.

Compliance with Ethical Standards

Involvement of human participant and animals

This article does not contain any studies with animals or humans performed by any of the authors. All the necessary permissions were obtained from the Institute's Ethical Committee and concerned authorities.

Information about informed consent

No informed consent was required as the studies does not involve any human participant.

Data availability statement

All the data and codes used in this study, as well as, the supplementary material can be made available from the corresponding author, upon reasonable request.

Funding information

No funding was involved in the present work.

References

1. Y. Liu, Z. Gu, S. Xia, B. Shi, X.N. Zhou, Y. Shi, J. Liu (2020) What are the underlying transmission patterns of COVID-19 outbreak?—an age-specific social contact characterization, *EClinicalMedicine* p. 100354

2. A. Temesgen, A. Gurmesa, Y. Getchew (2018) Joint modeling of longitudinal cd4 count and time-to-death of hiv/tb co-infected patients: A case of Jimma university specialized hospital, *Annals of Data Science* **5**(4), 659
3. D.L. Olson, Y. Shi, Y. Shi (2007) *Introduction to business data mining*, vol. 10 (McGraw-Hill/Irwin Englewood Cliffs)
4. J. Li, K. Guo, E.H. Viedma, H. Lee, J. Liu, N. Zhong, L.F.A.M. Gomes, F.G. Filip, S.C. Fang (2020) M.S. Özdemir, et al., Culture vs policy: More global collaboration to effectively combat COVID-19, *The Innovation* p. 100023
5. Y. Shi, Y. Tian, G. Kou, Y. Peng, J. Li (2011) *Optimization based data mining: theory and applications* (Springer Science & Business Media)
6. J. Vomlel, H. Kruzík, P. Tuma, J. Precek, M. Hutýra (2012) Machine learning methods for mortality prediction in patients with st elevation myocardial infarction, *Proceedings of WUPES*, 204
7. P. Kumar, H. Kalita, S. Patariya, Y.D. Sharma, C. Nanda, M. Rani, J. Rahmani, A.S. Bhagavathula (2020) Forecasting the dynamics of COVID-19 pandemic in top 15 countries in April 2020: Arima model with machine learning approach, medRxiv
8. S. Tuli, S. Tuli, R. Tuli, S.S. Gill (2020) Predicting the growth and trend of COVID-19 pandemic using machine learning and cloud computing, *Internet of Things* p. 100222
9. F. Petropoulos, S. Makridakis (2020) Forecasting the novel coronavirus COVID-19, *PLoS one* **15**(3), e0231236
10. J.T. Wu, K. Leung, G.M. Leung (2020) Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: a modelling study, *The Lancet* **395**(10225), 689
11. Z. Yang, Z. Zeng, K. Wang, S.S. Wong, W. Liang, M. Zanin, P. Liu, X. Cao, Z. Gao, Z. Mai, et al. (2020) Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions, *Journal of Thoracic Disease* **12**(3), 165
12. M.R. Bhatnagar (2020) Covid-19: Mathematical modeling and predictions, submitted to ARXIV. Online available at: <http://web.iitd.ac.in/~manav/COVID.pdf>
13. X. Zhang, R. Ma, L. Wang (2020) Predicting turning point, duration and attack rate of COVID-19 outbreaks in major western countries, *Chaos, Solitons & Fractals* p. 109829
14. B.F. Maier, D. Brockmann (2020) Effective containment explains subexponential growth in recent confirmed COVID-19 cases in China, *Science* **368**(6492), 742
15. L. Li, Z. Yang, Z. Dang, C. Meng, J. Huang, H. Meng, D. Wang, G. Chen, J. Zhang, H. Peng, et al. (2020) Propagation analysis and prediction of the COVID-19, *Infectious Disease Modelling* **5**, 282
16. A. Tomar, N. Gupta (2020) Prediction for the spread of COVID-19 in India and effectiveness of preventive measures, *Science of The Total Environment* p. 138762
17. S. Kumar (2020) Monitoring novel corona virus (COVID-19) infections in India by cluster analysis, *Annals of Data Science* p. 1
18. E.O.O. Max Roser, Hannah Ritchie, J. Hasell (2020) Coronavirus pandemic (COVID-19), *Our World in Data*. <https://ourworldindata.org/coronavirus>
19. N. Chintalapudi, G. Battineni, F. Amenta (2020) COVID-19 disease outbreak forecasting of registered and recovered cases after sixty day lockdown in Italy: A data driven model approach, *Journal of Microbiology, Immunology and Infection*
20. N. Poonia, S. Azad (2020) Short-term forecasts of COVID-19 spread across Indian states until 1 may 2020, arXiv preprint arXiv:2004.13538
21. R. Gupta, S.K. Pal (2020) Trend analysis and forecasting of COVID-19 outbreak in India, medRxiv