BMJ Open

# Application of a data-driven XGBoost model for the prediction of COVID-19 in the USA: a time-series study

Zheng-gang Fang,[1] Shu-qin Yang,[1] Cai-xia Lv,[1] Shu-yi An,[2] Wei Wu [1]

**To cite:** Fang Z, Yang S, Lv C, *et al*. Application of a data-driven XGBoost model for the prediction of COVID-19 in the USA: a time-series study. *BMJ Open* 2022;**12**:e056685. doi:10.1136/bmjopen-2021-056685

► Prepublication history and additional supplemental material for this paper are available online. To view these files, please visit the journal online (http://dx.doi.org/10.1136/bmjopen-2021-056685).

Received 24 August 2021
Accepted 20 June 2022

[1] Department of Epidemiology, China Medical University, Shenyang, China
[2] Department of Social Medicine and Health, Liaoning Provincial Center for Disease Control and Prevention, Shenyang, China

**Correspondence to**
Dr Wei Wu; wuwei@cmu.edu.cn

## ABSTRACT

**Objective** The COVID-19 outbreak was first reported in Wuhan, China, and has been acknowledged as a pandemic due to its rapid spread worldwide. Predicting the trend of COVID-19 is of great significance for its prevention. A comparison between the autoregressive integrated moving average (ARIMA) model and the eXtreme Gradient Boosting (XGBoost) model was conducted to determine which was more accurate for anticipating the occurrence of COVID-19 in the USA.

**Design** Time-series study.

**Setting** The USA was the setting for this study.

**Main outcome measures** Three accuracy metrics, mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE), were applied to evaluate the performance of the two models.

**Results** In our study, for the training set and the validation set, the MAE, RMSE and MAPE of the XGBoost model were less than those of the ARIMA model.

**Conclusions** The XGBoost model can help improve prediction of COVID-19 cases in the USA over the ARIMA model.

## STRENGTHS AND LIMITATIONS OF THIS STUDY

⇒ This study used the autoregressive integrated moving average and eXtreme Gradient Boosting (XGBoost) models to predict cases of COVID-19 in the USA.

⇒ Data on vaccination in the USA were introduced into the XGBoost model.

⇒ The seasonality of data was considered in both models.

⇒ The study period was relatively small and should be expanded to better reflect the future development of COVID-19 in the USA.

⇒ The XGBoost model was built based on prevaccination-induced herd immunity. Therefore, as the cases of more transmissible variants increase, the accuracy of prediction may decline.

## INTRODUCTION

First detected in Wuhan, China, and subsequently spread to all over the world, COVID-19 (http://COVID-19.who.int/) promises to be a defining global health event of the 21 century and has posed a severe and growing threat to public health.[1 2] Immediately after the first case in the USA was identified on 20 January 2020, COVID-19 cases increased exponentially until 11 July 2021, on that date were 33 595 701 cases and 598 442 deaths.[3] The majority of cases experience mild-to-moderate respiratory illness, but even death has resulted.[4] The common symptoms resulting from COVID-19 infection appear to be wide, encompassing fever, cough, fatigue and sore throat.[5 6] The clinical features of most patients are fever, and some have dyspnoea and extensive pneumonia infiltrates on CT scan of the chest.[7 8]

Given the uncertainty around decisions on the accurate time of the emergence and disappearance of the disease, it has been an increasingly important area of study in short-term forecasting to create better plans and more appropriate responses. Time-series analysis is beneficial for understanding the association of variables by using different models and obtaining more accurate predictions. The autoregressive integrated moving average (ARIMA) model by Box and Jenkins is the most common analytical method in data science. It is used for processing not only stationary but also non-stationary time series and is even applicable to seasonal time series.[9] However, infectious diseases are affected by many factors, and their time series usually do not conform to a linear function. Therefore, the Box-Jenkins based ARIMA model is insufficient to handle non-linear situations well. In contrast, the eXtreme Gradient Boosting (XGBoost) model is a flexible machine learning method capable of dealing with the non-linearity of time series through its strong self-learning ability.

The incidence of COVID-19 has varied greatly among countries,[10] and it has been noted that vaccination may play a key role in the containment of the COVID-19 pandemic.[11 12] Vaccines against COVID-19 now used in the USA have demonstrated high effectiveness.[13] Therefore, effective vaccines against COVID-19 will be essential to lowering morbidity and mortality. Nevertheless, to date, no researchers have included vaccinated individuals in the XGBoost model to forecast the incidence of COVID-19.

In this study, ARIMA and XGBoost models were developed to fit and forecast COVID-19 in the USA. In addition, we determined which of those models is a better predictor of COVID-19 in the USA by comparing the fit and forecast accuracies of the two models.

## METHODS

### Data sources

Data on COVID-19 cases[3] and vaccination[13] in the USA were collected from the website of the Centers for Disease Control and Prevention of the USA (https://COVID-19.cdc.gov). The daily data on COVID-19 in the USA from 13 December 2020 to 30 June 2021 were split into training (13 December 2020 to 16 June 2021) and validation sets (17 June 2021 to 30 June 2021). The models were established on training data and tested on the validation set.

### Seasonal ARIMA model

ARIMA models have often been used for the prediction of infectious diseases, such as dengue,[14] Hemorrhagic fever with renal syndrome (HFRS)[15] and malaria.[16] Considering time trends, periodic changes and random fluctuations, it has become a common model in data science. ARIMA is optimal for data containing trend, cyclicity and seasonality.[17] In our study, an ARIMA (p, d, q) (P, D, Q) [S] model was built, in which p represents the autoregression (AR) order, d the difference order and q the moving average (MA) order. S denotes the period of the seasonal trend and P, D and Q are the seasonal terms for the seasonal ARIMA. Parameters (P, D, Q) and (p, d, q) are determined according to the partial autocorrelation function (PACF) and autocorrelation function (ACF). Parameter S is chosen by the periodic length of seasonality. The seasonal model can be presented as follows:

$$Y_t = T_t + S_t + R_t$$

where $T_t$, $S_t$ and $R_t$ denote the tendency, seasonal effect and random effects, respectively. By differencing, we stabilised the time series. An augmented Dickey-Fuller (ADF) test is used to confirm this stabilisation. The corrected Akaike's information criterion (AICc) informs us of the goodness of fit of the ARIMA model. The model with the minimum value will be regarded as optimal. Finally, the Ljung-Box test was used to examine whether the residual sequences were white noise.

### XGBoost model

The XGBoost model is a decision tree-based machine learning algorithm that is widely used in data science. By using an internal algorithm that combines the results from multiple individual trees, we can yield accurate predictions.[18] Simultaneously, the model shows the ranking of input features. Moreover, XGBoost can help us obtain a stronger classifier from other classifiers and has other benefits, such as avoiding overfitting, effectively dealing with missing values and reducing running time by parallel and distributed calculation.[19] The objective function of the XGBoost model is as follow:

$$Y^k = \sum_{i=1}^{n} l\left(\left(y_i, y_i^{k-1}\right) + f_k\left(x_i\right)\right) + \Omega\left(f_k\right)$$

where $n$ denotes the number of training data, $x_i$ and $y_i$ are the feature vector and its label at the $i^{th}$ instance, $y_i^{k-1}$ represents the prediction of the $i^{th}$ instance at the $t-1^{th}$ iteration, $l$ is a loss function that calculates the difference between the label and the final forecast plus the new tree output, $f_k$ denotes a new tree that classifies the $i^{th}$ instance with $x_i$, and denotes the regularisation term that penalises the complexity of the new tree.[20] In the process of building the XGBoost model, the lag terms in the data are the input items, which are used for the prediction of data. Given the existence of a seasonal trend, we built seven lag terms (1-day to 7-day lag) as input items. To transform week variables to a common format, a one-hot encoding technique was used, which can convert categorical variables into numerical values in machine learning preprocessing. The week variable is used as a one-hot representation encoded into a matrix, whose columns correspond to the presence of Monday, Wednesday, Thursday, Friday, Saturday and Sunday. The matrix of the week variable is represented as follows:

$$\mathrm{W}_i = \begin{cases} 1 \text{ if week is } \left(\text{Mon, Wed, Thu, Fri, Sat, Sun}\right) \\ 0 \text{ otherwise} \end{cases}$$

We built a numerical variable from 1 to the number of observations to analyse the effect of the time trend. The hyperparameters, including SubsampRate, ColsampRate, Depth, MinChild and eta, should be adjusted to optimise the XGBoost model.

### Model selection

In our study, three accuracy metrics were applied to evaluate the performance of the models: mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE), as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right|$$

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

$$\mathrm{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - y_i}{y_i}\right|$$

In these equations, $n$, $y_i$ and $y_i$ are the number of observations, the forecasted value, and the actual value, respectively. MAE is the mean of the absolute prediction error, which represents the MAEs between the actual and the prediction. RMSE is the square root of the average squared error, which is frequently used to evaluate the difference between the prediction and the actual value. MAPE represents the mean error between the actual and the prediction in percentage form, which computes the

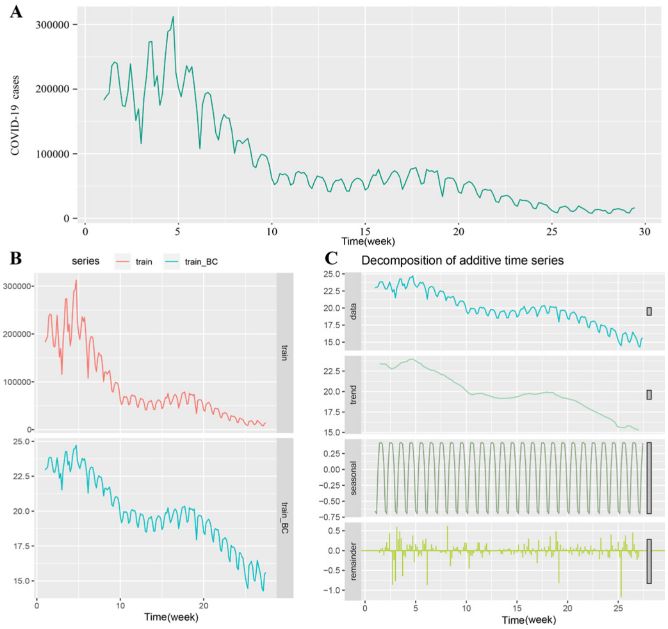

**Figure 1** (A) Daily cases of COVID-19 in the USA from 13 December 2020 to 30 June 2021. (B) Contrast of the primary and transformed series of COVID-19. (C) Decomposition of the transformed series of COVID-19.

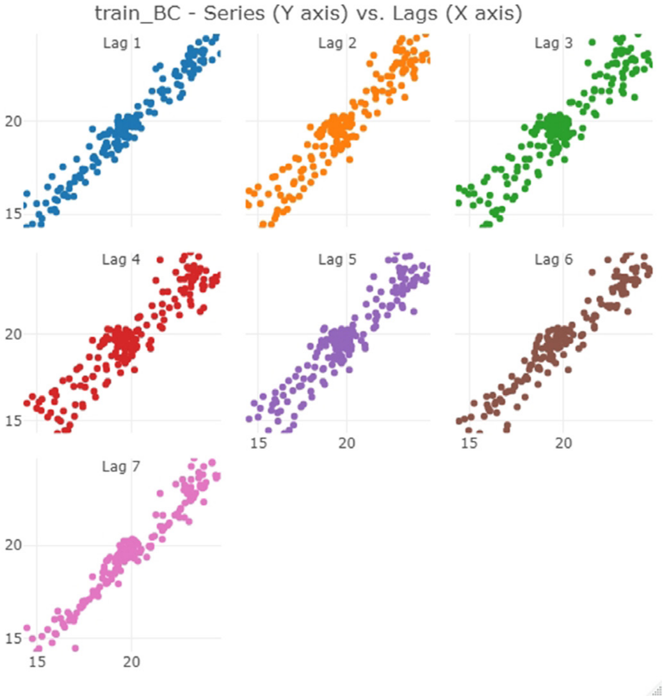

**Figure 2** Difference correlations in the first seven lags.

average absolute percent difference between the actual and the prediction. As MAPE, RMSE and MAE approach zero, the prediction results are considered more accurate.

### Data analysis

In our study, all data were processed in R V.4.1.0 software. We used the xts, TSstudio and tseries packages to analyse of data and the ggplots2 and dygraphs packages to draw diagrams. The proposed models were established via forecast and xgboost packages (see R codes in online supplemental material 1).

### Patient and public involvement

No patients were involved.

## RESULTS

### Characteristics of COVID-19 cases

As of 11 July 2021, the total number of COVID-19 cases had reached 33 595 701 in the USA. According to the plot of daily cases, a study period was chosen from 13 December 2020 to 30 June 2021. First, it was certain to make the series become stationary. The time-series graph, given in figure 1A, shows that the data have a downward trend and fluctuate greatly, and the ADF test also confirms its non-stationarity. By Box-Cox transformation, the original data became more stationary with less fluctuation (figure 1B),[21] and we then decomposed it. The Box-Cox transformation data, seasonal trend, time trend and remainder are shown in figure 1C. The diagrams show that there is a seasonal pattern and a trend. Moreover, we drew the relationship between the transformed and lag series (figure 2). To stabilise the time series, seasonal and regular differencing were applied. We conducted first-order and seven-order differencing (seasonal differencing) to address the instability caused by time trends and seasonal factors.

### Forecasting the cases of COVID-19 by the seasonal ARIMA model

After first-order and seasonal differencing, the COVID-19 data transformed by Box-Cox transformation became stationary (figure 3), and the ADF test also supported stationarity (t=−5.6143, p<0.01). This result showed us that the parameters d and D are 1 and 1 in the seasonal ARIMA model.

The plots of ACF (figure 4A) and PACF (figure 4B) showed the temporal dependence of COVID-19 cases, and thus, we tried to build a seasonal ARIMA model with nonseasonal (p, d, q) and seasonal (P, D, Q) parameters. After differencing, the peak values (lag 1, 4, 7 and 14) in figure 4A indicated that the maximum q and Q values should be set to 4 and 2, respectively. At the same time, significant peak values at lags 1, 2 and 4, and 7, 14, 21 and 28 are observed in figure 4B, and thus, the maximum p and P values should be 4 and 4, respectively. Then, we found the model with the lowest AICc value via the auto.arima function. Finally, the optimal model was ARIMA $(0,1,1)(0,1,1)_7$ (table 1), and the Ljung-Box test indicated that the residual series was white noise (p=0.6325). The time plot of the residuals, the corresponding ACF and the histogram also checked that residuals from the model were white noise. (figure 5). The ARIMA $(0,1,1)(0,1,1)_7$ model performed well in the fit and forecasting of COVID-19 cases. The details are given in figure 6A.

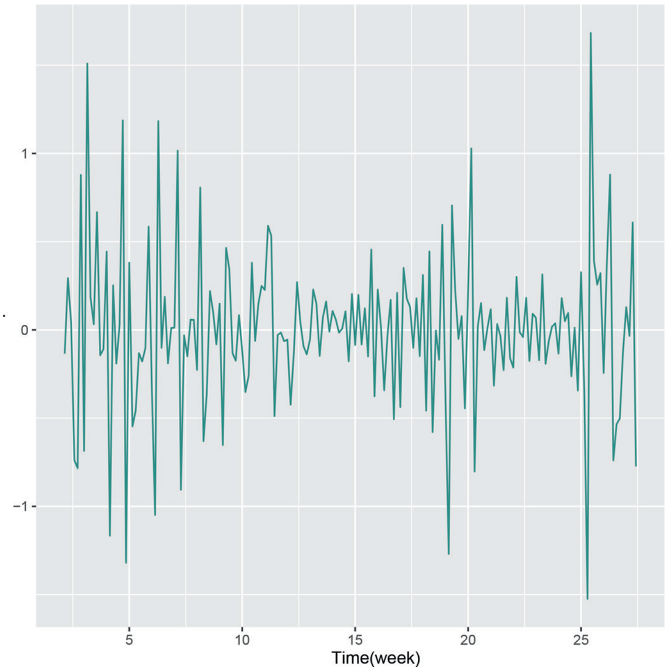

**Figure 3** Cases of COVID-19 in the USA after transformation and differences.

### Forecasting the cases of COVID-19 by the XGBoost model

In the application of the XGBoost model, the value of hyperparameters is essentially important. We consistently built models via preset bounds for hyperparameters, and then we obtained the best one in the final training with 168 rounds. The hyperparameters of the optimal model were: SubSampRate=0.5, ColSampRate=0.2, Depth=4, MinChild=2 and eta=0.07. The fit and forecast results of the optimal model are shown in figure 6B.

### Models comparison

For the ARIMA $(0,1,1)\,(0,1,1)_7$ model, we lost 8 observations in the training set after differencing, and only 162 observations were used for analysis. For the XGBoost model, we built seven lag terms (1-day to 7-day lag) as input terms because of the existence of seasonal trends. Accordingly, only 163 observations remained for analysis. The fit and forecast information of the two models are illustrated in table 2. In the training set and the validation set, compared with the seasonal ARIMA model, the XGBoost model had smaller values of MAE, RMSE and MAPE. It should be noted that the performance of the test set in the XGBoost model outweighed that of the validation set in the seasonal ARIMA model. For the XGBoost model, the MAPE values of the training and validation sets (4.046% and 7.892%) were excellent.

## DISCUSSION

In this paper, we developed two models (seasonal ARIMA and XGBoost) and used past data on daily cases of COVID-19 to predict 14 days ahead in the USA. The fit and prediction accuracies of the proposed models were assessed by three criteria. The model results show that

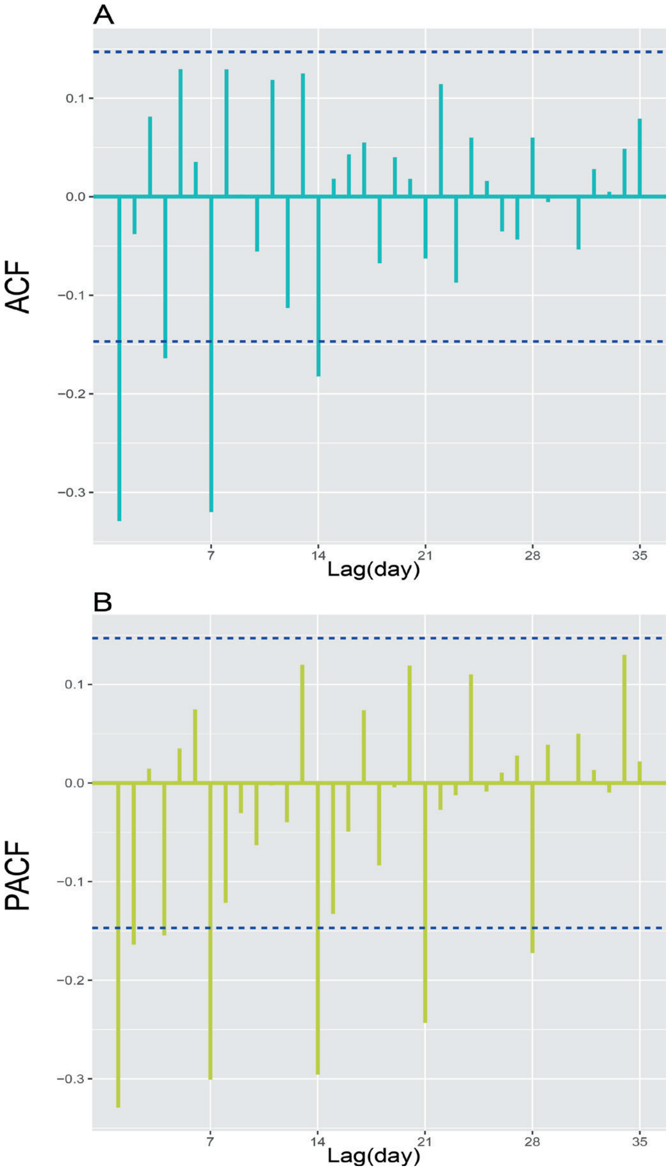

**Figure 4** (A) Autocorrelation function (ACF) and (B) partial autocorrelation function (PACF) diagrams for cases of COVID-19 in the USA after transformation and differences.

**Table 1** Parameters of the ARIMA (0,1,1) (0,1,1)$_7$ model

| **Series: train<br>ARIMA (0,1,1) (0,1,1)$_7$** | | | |
|---|---|---|---|
| Coefficients | | ma1 | sma1 |
| | | −0.391 | −0.917 |
| | SE | −0.070 | 0.067 |
| CIs of coefficients | | 2.5% | 97.5% |
| | ma1 | −0.528 | −0.253 |
| | sma1 | −1.048 | −0.785 |
| AICc | | 128.920 | |

AICc, Akaike’s information criterion; ARIMA, autoregressive integrated moving average.

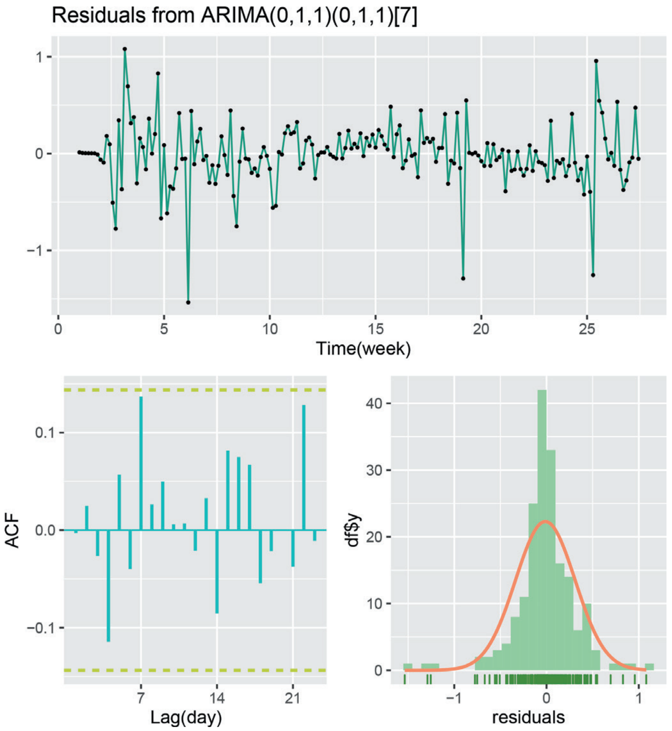

**Figure 5** The combination of residuals, the corresponding autocorrelation function (ACF) diagram, and the histogram for the autoregressive integrated moving average (ARIMA) (0,1,1)(0,1,1)$_7$ model.

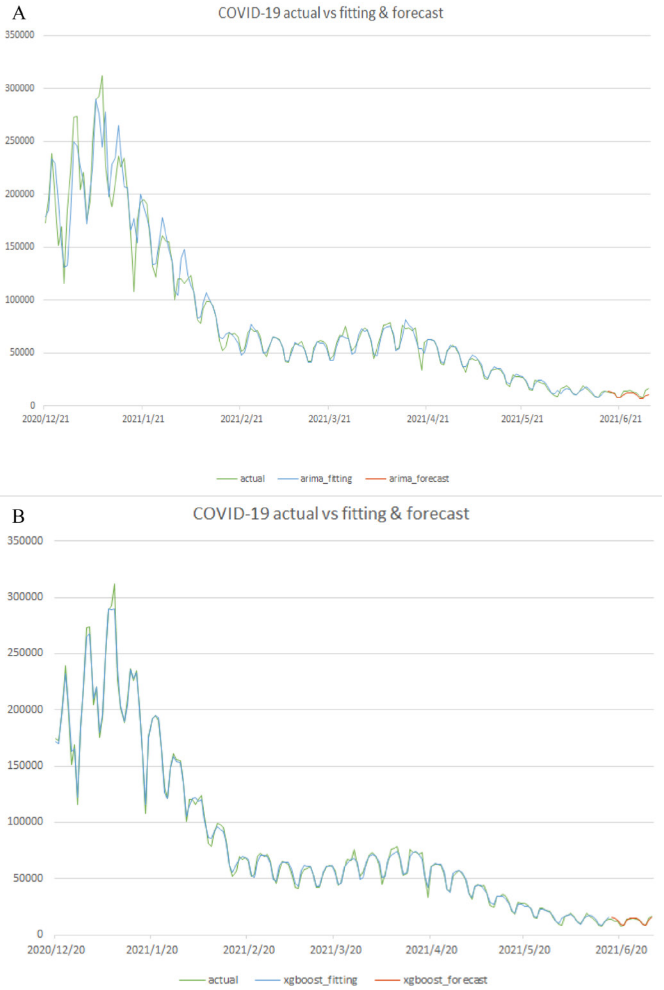

**Figure 6** Fit and forecast results of (A) autoregressive integrated moving average (ARIMA) (0,1,1)(0,1,1)$_7$ and (B) eXtreme Gradient Boosting (XGBoost) models.

the XGBoost model has better fit and better forecast COVID-19 cases in the USA. The prediction of cases of COVID-19 can help the government and the public take precautionary measures to control the further spread of COVID-19.

The ARIMA model is commonly used for the prediction of time-series data, and it can show autocorrelations in data. The XGBoost model is a decision tree-based machine learning model, by which we can uncover the non-linearity in the time series of COVID-19 cases. Accordingly, our models not only retain the irregular trend of the COVID-19 data but also capture the incidental fluctuation. The ARIMA model combines AR with the MA, which is beneficial for capturing the characteristics of data in nature and making a more exact forecast. The seasonal ARIMA model has been among the most significant predictors for seasonal forecasts of time series.[15 22 23] Normally, the loss of data happens more often with more differences. In our study, we only used the data on the daily number of COVID-19 cases to build the ARIMA model. We first conducted a first-order difference while we found that the data did not become stationary. We conducted a seasonal difference in the next step, and the result was good. Finally, the ARIMA $(0,1,1)\,(0,1,1)_7$ model was selected as the optimal model with the minimum AICc. From the results of the ARIMA $(0,1,1)\,(0,1,1)_7$ model, we can conclude that the model precisely reflects the seasonality in the data on COVID-19 cases. Nevertheless, owing to the non-linearity of the data, the MAE, RMSE and MAPE in the validation set were not good.

Starting the experimental evaluation with the seasonal ARIMA, we then applied the XGBoost model to further analyse the time series in the USA. In current COVID-19 research, the effectiveness of vaccines against COVID-19 has been confirmed. Once vaccines have been approved for use in individuals, sufficient and effective vaccines will help build herd immunity among people.[24–26] From the variable importance graph (figure 7) for the XGBoost model, we also see that the significance scores of vaccine variables (fully vaccinated and at least one dose vaccinated) rank in the second and fifth positions. As a result, vaccines have played an important role in the spread of COVID-19 in the USA. Vaccinations have been administered in countries on different dates. As of 11 July 2021, more than 158 million people were fully vaccinated and 183 million had at least one dose against COVID-19 in the USA. Based on the afore-mentioned evidence, in addition to the data on the daily number of COVID-19 cases, we also collected the vaccination data to build the XGBoost model. The vaccination data included the daily

Table 2 Performance of the ARIMA (0,1,1) $(0,1,1)_7$ and XGBoost model

| | Training set | | | Test set | | |
|---|---|---|---|---|---|---|
| Model | MAE | RMSE | MAPE (%) | MAE | RMSE | MAPE (%) |
| ARIMA (0,1,1) $(0,1,1)_7$ | 7061.536 | 13 517.664 | 7.996 | 2083.571 | 2633.424 | 15.884 |
| XGBoost | 2331.134 | 3500.331 | 4.046 | 962.357 | 1209.984 | 7.892 |

ARIMA, autoregressive integrated moving average; MAE, mean absolute error; MAPE, mean absolute percentage error; RMSE, root mean square error; XGBoost, eXtreme Gradient Boosting.

cumulative number of fully vaccinated and those with at least one dose. The XGBoost model has already been carried out in studies to predict the trend in COVID-19.[18 19 27–34] Luo *et al*[19] used the long short-term memory and XGBoost models in the prediction of COVID-19 in the USA and assessed the ranking of features via the XGBoost model. Khan *et al*[31] aimed to predict the mortality rate in confirmed COVID-19 patients from 146 countries employing the XGBoost model. Ahamad *et al*[34] developed several machine learning algorithms and discovered that the XGBoost model could precisely predict COVID-19 trends and simultaneously select features associated with them for all ages. In this paper, the XGBoost model is better than the seasonal ARIMA model based on the fit and forecast results, which is probably because vaccine variables were considered. The forecasting results showed that the MAEs of the seasonal ARIMA and XGBoost models were 2083.571 and 962.357, respectively. The RMSE values were 2633.424 and 1209.984, respectively. The MAPE (%) values were 15.884 and 7.892, respectively. Additionally, the accuracy metric values for the training data (2331.134, 3500.331, 4.016) and the validation data (962.357, 1209.984, 7.892) are quite small. As shown in table 2. This finding also suggests the high accuracy of the XGBoost model in the fit and forecast of COVID-19. However, new variants ravaging the USA are raising worries about the effectiveness of currently administered vaccines.[35 36] The XGBoost model is built based on prevaccination-induced herd immunity in the USA. Therefore, as the cases of more transmissible variants increase, the accuracy of prediction may decrease.

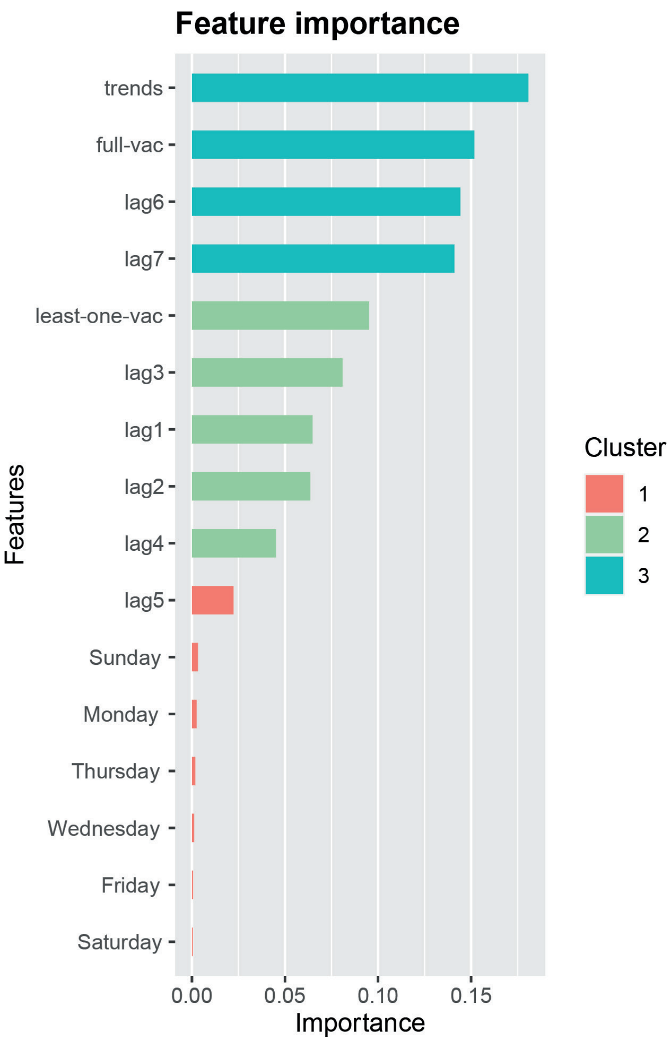

Figure 7 Feature importance for COVID-19 cases in the USA.

The time series of epidemics are always characterised by instability and volatility. Therefore, differencing and transformation are required to render them stationary. The ARIMA model is inapplicable to processing data that cannot be converted into stationary data, whereas the XGBoost model can dismiss it. Hence, compared with the traditional ARIMA model, the XGBoost model will achieve a broader application in practice. However, we first developed a seasonal ARIMA. According to the principle of this model, we used the past data on daily cases of COVID-19 to predict 14 days ahead by using the forecast function in the forecast package. The one-step ahead prediction method was performed in the XGBoost model. One-step ahead prediction uses actual past data to obtain a 1-day prediction. For example, actual data before and at time t as the model inputs to forecast the daily cases at time t+1, and actual data before and at time t+1 are used as the model inputs to forecast the daily cases at time t+2. According to the one-step prediction, we obtain the 14-day forecasting values. To a certain extent, the ARIMA model is more useful in real-world applications because it can forecast over a longer period. The XGBoost model can only use one-step ahead prediction, especially when impact factors are used as inputs of the model. New data are needed to rebuild the model to better reflect the future development of COVID-19 in the USA. This prediction of cases of COVID-19 by the models can help the government make effective measures and policies to deal with COVID-19.

## CONCLUSIONS

Based on data from COVID-19 cases in the USA, we developed the XGBoost and seasonal ARIMA models, by which we conducted a 14-day, out-of -sample prediction. We obtained the fit and forecast results and compared the performance of the two models with the MAE, RMSE and MAPE values. We concluded that the XGBoost model leads to a notable improvement in the fit and prediction accuracy.

**Contributors** Z-gF designed and drafted the manuscript. S-qY and C-xL participated in the data collection. Z-gF and WW participated in the data analysis. S-yA and WW made critically revised the manuscript. The final manuscript has been read and approved by all the authors. WW is responsible for the overall content as the guarantor.

**Funding** This study was sponsored by the National Natural Science Foundation of China (grant no. 81202254), the Health and Medical Big Data Research Project of China Medical University (grant no. HMB201903105) and the Science Foundation of Liaoning Provincial Department of Education (LJKQZ2021027).

**Competing interests** None declared.

**Patient and public involvement** Patients and/or the public were not involved in the design, or conduct, or reporting, or dissemination plans of this research.

**Patient consent for publication** Not applicable.

**Ethics approval** Not applicable.

**Provenance and peer review** Not commissioned; externally peer reviewed.

**Data availability statement** Data are available in a public, open access repository. Data are available in a public, open access repository. The data that support the findings of this study are openly available from the website of the Centers for Disease Control and Prevention of the United States at https://covid.cdc.gov.

**Supplemental material** This content has been supplied by the author(s). It has not been vetted by BMJ Publishing Group Limited (BMJ) and may not have been peer-reviewed. Any opinions or recommendations discussed are solely those of the author(s) and are not endorsed by BMJ. BMJ disclaims all liability and responsibility arising from any reliance placed on the content. Where the content includes any translated material, BMJ does not warrant the accuracy and reliability of the translations (including but not limited to local regulations, clinical guidelines, terminology, drug names and drug dosages), and is not responsible for any error and/or omissions arising from translation and adaptation or otherwise.

**ORCID iD**

Wei Wu http://orcid.org/0000-0001-5535-1682

## REFERENCES

1 Wang C, Horby PW, Hayden FG, *et al*. A novel coronavirus outbreak of global health concern. *Lancet* 2020;395:470–3.
2 Guan W-J, Ni Z-Y, Hu Y, *et al*. Clinical characteristics of coronavirus disease 2019 in China. *N Engl J Med* 2020;382:1708–20.
3 Centers for Disease Control and Prevention. Data Table for Daily Case Trends - The United States. COVID Data Tracker, 11 July, 2021. Available: https://covid.cdc.gov/covid-data-tracker/#trends_dailycases
4 Zhou F, Yu T, Du R, *et al*. Clinical course and risk factors for mortality of adult inpatients with COVID-19 in Wuhan, China: a retrospective cohort study. *Lancet* 2020;395:1054–62.
5 Wang Y, Wang Y, Chen Y, *et al*. Unique epidemiological and clinical features of the emerging 2019 novel coronavirus pneumonia (COVID-19) implicate special control measures. *J Med Virol* 2020;92:568–76.
6 Pedersen SF, Ho Y-C. SARS-CoV-2: a storm is raging. *J Clin Invest* 2020;130:2202–5.
7 Jin Y, Yang H, Ji W, *et al*. Virology, epidemiology, pathogenesis, and control of COVID-19. *Viruses* 2020;12:372.
8 Wiersinga WJ, Rhodes A, Cheng AC, *et al*. Pathophysiology, transmission, diagnosis, and treatment of coronavirus disease 2019 (COVID-19). *JAMA* 2020;324:782–93.
9 Singh S, Parmar KS, Kumar J, *et al*. Development of new hybrid model of discrete wavelet decomposition and autoregressive integrated moving average (ARIMA) models in application to one month forecast the casualties cases of COVID-19. *Chaos Solitons Fractals* 2020;135:109866.
10 CDC COVID-19 Response Team. Geographic Differences in COVID-19 Cases, Deaths, and Incidence - United States, February 12-April 7, 2020. *MMWR Morb Mortal Wkly Rep* 2020;69:465–71.
11 Aslam S, Adler E, Mekeel K, *et al*. Clinical effectiveness of COVID-19 vaccination in solid organ transplant recipients. *Transplant Infectious Disease* 2021;23.
12 Yengil E, Onlen Y, Ozer C, *et al*. Effectiveness of booster measles-mumps-rubella vaccination in lower COVID-19 infection rates: a retrospective cohort study in Turkish adults. *Int J Gen Med* 2021;14:1757–62.
13 Centers for Disease Control and Prevention. Trends in number of COVID-19 vaccinations in the US. COVID data Tracker, 11 July, 2021. Available: https://covid.cdc.gov/covid-data-tracker/#vaccination-trends
14 Gharbi M, Quenel P, Gustave J, *et al*. Time series analysis of dengue incidence in Guadeloupe, French West Indies: Forecasting models using climate variables as predictors. *BMC Infect Dis* 2011;11:166.
15 Ye G-H, Alim M, Guan P, *et al*. Improving the precision of modeling the incidence of hemorrhagic fever with renal syndrome in mainland China with an ensemble machine learning approach. *PLoS One* 2021;16:e0248597.
16 Midekisa A, Senay G, Henebry GM, *et al*. Remote sensing-based time series models for malaria early warning in the highlands of Ethiopia. *Malar J* 2012;11:165.
17 Ceylan Z. Estimation of COVID-19 prevalence in Italy, Spain, and France. *Sci Total Environ* 2020;729:138817.
18 Mehta M, Julaiti J, Griffin P, *et al*. Early stage machine Learning–Based prediction of US County vulnerability to the COVID-19 pandemic: machine learning approach. *JMIR Public Health Surveill* 2020;6:e19446–87.
19 Luo J, Zhang Z, Fu Y, *et al*. Time series prediction of COVID-19 transmission in America using LSTM and XGBoost algorithms. *Results Phys* 2021;27:104462–62.
20 Nishio M, Nishizawa M, Sugiyama O, *et al*. Computer-Aided diagnosis of lung nodule using gradient tree boosting and Bayesian optimization. *PLoS One* 2018;13:e0195875.
21 Curran-Everett D. Explorations in statistics: the log transformation. *Adv Physiol Educ* 2018;42:343–7.
22 Yousaf M, Zahir S, Riaz M, *et al*. Statistical analysis of forecasting COVID-19 for upcoming month in Pakistan. *Chaos Solitons Fractals* 2020;138:109926.
23 Wu W, An S-Y, Guan P, *et al*. Time series analysis of human brucellosis in mainland China by using Elman and Jordan recurrent neural networks. *BMC Infect Dis* 2019;19:11.
24 Cihan P. Forecasting fully vaccinated people against COVID-19 and examining future vaccination rate for herd immunity in the US, Asia, Europe, Africa, South America, and the world. *Appl Soft Comput* 2021;111:107708–08.
25 Omer SB, Yildirim I, Forman HP. Herd immunity and implications for SARS-CoV-2 control. *JAMA* 2020;324:2095–6.
26 Quinonez E, Vahed M, Hashemi Shahraki A, *et al*. Structural analysis of the novel variants of SARS-CoV-2 and forecasting in North America. *Viruses* 2021;13. doi:10.3390/v13050930. [Epub ahead of print: 17 05 2021].
27 Wang K, Zuo P, Liu Y, *et al*. Clinical and laboratory predictors of in-hospital mortality in patients with coronavirus Disease-2019: a cohort study in Wuhan, China. *Clin Infect Dis* 2020;71:2079–88.
28 Vaid A, Somani S, Russak AJ, *et al*. Machine learning to predict mortality and critical events in a cohort of patients with COVID-19 in New York City: model development and validation. *J Med Internet Res* 2020;22:e24018.
29 Wang JM, Liu W, Chen X. Predictive modeling of morbidity and mortality in COVID-19 hospitalized patients and its clinical implications. *medRxiv* 2020.

30 Ma X, Ng M, Xu S, *et al*. Development and validation of prognosis model of mortality risk in patients with COVID-19. *Epidemiol Infect* 2020;148:e168. doi:10.1017/S0950268820001727

31 Khan IU, Aslam N, Aljabri M, *et al*. Computational Intelligence-Based model for mortality rate prediction in COVID-19 patients. *Int J Environ Res Public Health* 2021;18. doi:10.3390/ijerph18126429. [Epub ahead of print: 14 06 2021].

32 Karthikeyan A, Garg A, Vinod PK, *et al*. Machine learning based clinical decision support system for early COVID-19 mortality prediction. *Front Public Health* 2021;9:626697.

33 Bertsimas D, Lukin G, Mingardi L, *et al*. COVID-19 mortality risk assessment: an international multi-center study. *PLoS One* 2020;15:e0243262.

34 Ahamad MM, Aktar S, Rashed-Al-Mahfuz M, *et al*. A machine learning model to identify early stage symptoms of SARS-Cov-2 infected patients. *Expert Syst Appl* 2020;160:113661.

35 Washington NL, Gangavarapu K, Zeller M, *et al*. Genomic epidemiology identifies emergence and rapid transmission of SARS-CoV-2 B.1.1.7 in the United States. *medRxiv* 2021. doi:10.1101/2021.02.06.21251159. [Epub ahead of print: 07 Feb 2021].

36 Firestone MJ, Lorentz AJ, Meyer S, *et al*. First Identified Cases of SARS-CoV-2 Variant P.1 in the United States - Minnesota, January 2021. *MMWR Morb Mortal Wkly Rep* 2021;70:346–7.

```
library(TSstudio)
library(forecast)
library(tseries)
library(ggplot2)
library(dplyr)
library(plotly)
library(gridExtra)
library(lubridate)
library(xgboost)
library(Metrics)
library(Ckmeans.1d.dp)
library(tidyverse)
library(magrittr)
library(dygraphs)
library(xts)
library(latticeExtra)
#clearing database
rm(list=ls())

setwd("C:/Users/lxb/Desktop/COVID19/")
############################################################

raw.data.confirmed <- read.csv('./cdc/data_table_for_daily_case_trends__united_states.csv')
US_confirmed_df <- data.frame(date = seq.Date(from = as.Date('2020-01-22'), length.out =
dim(raw.data.confirmed)[1],by = 'day'),
                              count = raw.data.confirmed$New.Cases)
USA_df<-US_confirmed_df[c(327:(length(US_confirmed_df$count)-9)),]


vaccinations <- read.csv('./cdc/trends_in_number_of_covid19_vaccinations_in_the_us.csv')
vaccinations <- vaccinations[1:200,]
vaccinations <- vaccinations[c('People.with.at.least.One.Dose.Cumulative',
'People.Fully.Vaccinated.Cumulative')]

vaccinations$date<-USA_df$date
vaccinations<-rename(vaccinations, least_one_vac =
People.with.at.least.One.Dose.Cumulative)
vaccinations<-rename(vaccinations, full_vac = People.Fully.Vaccinated.Cumulative)


###############################################xgboost######################
#############
# function to take a vector and create a matrix of itself and lagged values
lagv <- function(x, maxlag, keeporig = TRUE){
  if(!is.vector(x) & !is.ts(x)){
    stop("x must be a vector or time series")
  }
  x <- as.vector(x)
  n <- length(x)
  z <- matrix(0, nrow = (n - maxlag), ncol = maxlag + 1)
  for(i in 1:ncol(z)){
    z[ , i] <- x[(maxlag + 2 - i):(n + 1 - i)]
  }
  varname <- "x"
  colnames(z) <- c(varname, paste0(varname, "_lag", 1:maxlag))
  if(!keeporig){
    z <- z[ ,-1]
  }
  return(z)
}


#####

USA_df$season <- factor(weekdays(USA_df$date), ordered=F)
USA_ts<-ts(USA_df$count)

USA_df <- USA_df[-c(1:7),]

lagx <- lagv(USA_ts,7,keeporig = F)
USA_data <- cbind(USA_df, lagx)

USA_data_ts<-USA_ts[-c(1:7)] %>% ts()

# Some models get one hot encoding of seasons
USA_data$z <- 1
seasons <- model.matrix(z ~ season, data = USA_data)[ ,-1]
USA_data <- cbind(USA_data, seasons)

USA_data[c('date', 'season', 'z')] <- NULL

USA_data$trends <- 1:dim(USA_data)[1]


USA_data <- cbind(USA_data, vaccinations[8:dim(vaccinations)[1],])


#####
```

```
input <- USA_data[,-1]
output <- USA_data[,1]


h <- 14


train_input <- input[1:(dim(USA_data)[1]-h),]
train_input <- data.matrix(train_input)
test_input <- input[(dim(USA_data)[1]-h+1):dim(USA_data)[1],]
test_input <- data.matrix(test_input)
train_output <- output[1:(dim(USA_data)[1]-h)]
test_output <- output[(dim(USA_data)[1]-h+1):dim(USA_data)[1]]


searchGridSubCol <- expand.grid(subsample = c(0.3,0.4,0.5,0.6),
                                colsample_bytree = c(0.1,0.2,0.3,0.4),
                                max_depth = c(4,5,6),
                                min_child = c(1,2),
                                eta = c(0.05,0.06,0.07,0.08))
ntrees <- 1000

rmseErrorsHyperparameters <- apply(searchGridSubCol, 1, function(parameterList){
  #Extract Parameters to test
  currentSubsampleRate <- parameterList[["subsample"]]
  currentColsampleRate <- parameterList[["colsample_bytree"]]
  currentDepth <- parameterList[["max_depth"]]
  currentEta <- parameterList[["eta"]]
  currentMinChild <- parameterList[["min_child"]]
  set.seed(1)
  xgboostModelCV <- xgb.cv(data = train_input,
                           label = train_output,
                           nrounds = ntrees,
                           nfold = 10,
                           showsd = F,
                           metrics = "rmse",
                           verbose = T,
                           eval_metric = "rmse",
                           objective = "reg:squarederror",
                           max_depth = currentDepth,
                           eta = currentEta,
                           subsample = currentSubsampleRate,
                                    colsample_bytree = currentColsampleRate,
                                    print_every_n = 10,
                                    min_child_weight = currentMinChild,
                                    booster = "gbtree",
                                    early_stopping_rounds = 10)

  xvalidationScores <- as.data.frame(xgboostModelCV$evaluation_log)
  rmse <- tail(xvalidationScores$test_rmse_mean, 1)
  trmse <- tail(xvalidationScores$train_rmse_mean,1)
  bestRounds <- xgboostModelCV$best_iteration
  output_param    <-    return(c(bestRounds,    rmse,    trmse,    currentSubsampleRate,
currentColsampleRate, currentDepth, currentEta, currentMinChild))})

output_param <- as.data.frame(t(rmseErrorsHyperparameters))
varnames <- c("bestRounds", "TestRMSE", "TrainRMSE", "SubSampRate", "ColSampRate",
"Depth", "eta", "MinChild")
names(output_param) <- varnames
output_param <- output_param[order(output_param["TestRMSE"]),]
head(output_param)

set.seed(1)
USA_xgboost <- xgboost(data = train_input,
                                label = train_output,
                                nrounds = output_param[1,]$bestRounds,
                                booster = "gbtree",
                                eta = output_param[1,]$eta,
                                max_depth = output_param[1,]$Depth,
                                min_child_weight = output_param[1,]$MinChild,
                                subsample = output_param[1,]$SubSampRate,
                                colsample_bytree = output_param[1,]$ColSampRate,
                                print_every_n = 100,
                                objective = "reg:squarederror",
                                verbose = TRUE,
                                eval_metric = "rmse"
)

USA_import <-xgb.importance(model = USA_xgboost)
xgb.ggplot.importance(USA_import)


####################one-step forecasts
fit_USA_xgboost <- predict(USA_xgboost, train_input)
fc_USA_xgboost <- predict(USA_xgboost, test_input)
```

```
split_USA <- ts_split(ts.obj = USA_data_ts, sample.out = h)
train_US <- split_USA$train
test_US <- split_USA$test

fit_result <- cbind(train_US, fit_USA_xgboost)
autoplot(fit_result,ylab = "")

forecast_result <- cbind(test_US,fc_USA_xgboost)
autoplot(forecast_result,ylab = "")


####################Accuary calculation

rmse(train_US,fit_USA_xgboost)
rmse(test_US,fc_USA_xgboost)

mae(train_US,fit_USA_xgboost)
mae(test_US,fc_USA_xgboost)

mape(train_US,fit_USA_xgboost)
mape(test_US,fc_USA_xgboost)

############################################################arima##########
########

USA_confirmed_df<-US_confirmed_df[c(327:(length(US_confirmed_df$count)-9)),]
US_confirmed_ts <- ts(USA_confirmed_df$count, start = c(1, 1), frequency = 7)
autoplot(US_confirmed_ts)

summary(US_confirmed_ts)
# Setting training and testing partitions
split_US_confirmed_ts <- ts_split(ts.obj = US_confirmed_ts, sample.out = h)
train <- split_US_confirmed_ts$train
test <- split_US_confirmed_ts$test

lambda <- BoxCox.lambda(train)
train_BC <- BoxCox(train, lambda)

autoplot(cbind(train,train_BC),facets = T,colour = T,ylab = "")

fit <- decompose(train_BC, type = 'additive')
autoplot(fit)
```

```
ts_lags(train_BC, lags = 1:7)
ts_lags(train_BC, lags = c(7, 14, 21, 28))
ts_seasonal(train_BC,type='normal')
ts_seasonal(train_BC,type='cycle')

ggAcf(train_BC, lag.max = 35)
ggPacf(train_BC, lag.max = 35)

col<-colorspace::sequential_hcl(n = 7, palette = "Mint")
train_BC %>% diff %>% autoplot(colour=col[3])
train_BC %>% diff %>% ggAcf(lag.max = 35)
train_BC %>% diff %>% ggPacf(lag.max = 35)

train_BC %>% diff %>% diff(lag =7) %>% autoplot(colour=col[3])
train_BC %>% diff %>% diff(lag =7) %>% ggAcf(lag.max = 35)
train_BC %>% diff %>% diff(lag =7) %>% ggPacf(lag.max = 35)

train_BC %>% diff %>% diff(lag =7) %>% adf.test

####################################################the                    auto.arima
function##########
US_auto_arima <- auto.arima(train,
                            d = 1,
                            D = 1,
                            max.p = 4,
                            max.q = 4,
                            max.P = 4,
                            max.Q = 2,
                            max.order = 14,
                            lambda = 'auto',
                            biasadj = TRUE,
                            stepwise = FALSE,
                            approximation = FALSE,
                            parallel = TRUE,
                            num.cores = NULL)

summary(US_auto_arima)
confint(US_auto_arima)
checkresiduals(US_auto_arima)

B <- NULL
for(i in 1:21){
  B[i] <- Box.test(residuals(US_auto_arima), lag = i, type = 'Ljung-Box')$p.value
}
plot(B, main = 'Ljung-Box test', ylab = 'p-value', xlab = 'lag', pch = 16, ylim = c(0,1)
     ,col=col[3])
abline(h = 0.05, lty = 2,col=col[3])

fc_US_arima <- forecast(US_auto_arima, biasadj = T, h = h)
test_forecast(actual=US_confirmed_ts, forecast.obj=fc_US_arima, train=train, test = test)
fc_US_arima$fitted[1:8] <- NA
round(forecast::accuracy(fc_US_arima,test),3)

#Fig_1(USA_confirmed_ts)
data_1 <- xts(x=USA_confirmed_df$count,order.by = USA_confirmed_df$date)
dygraph(data_1)

US_arima_excel<-data.frame(US_auto_arima[["x"]],fc_US_arima[["fitted"]])
write.csv(US_arima_excel,"C:/Users/lxb/Desktop/COVID19/US_ARIMA.csv")
US_xgboost_excel<-data.frame(US_auto_arima[["x"]][c(8:186)],fit_USA_xgboost)
write.csv(US_xgboost_excel,"C:/Users/lxb/Desktop/COVID19/US_XGBoost.csv")
```