Forecasting adversities of COVID-19 waves in India using intelligent computing

Arijit Chakraborty1 · Dipankar Das1 · Sajal Mitra2 · Debashis De3 · Anindya J. Pal4

Received: 16 May 2021 / Accepted: 14 September 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract
The second wave of the COVID-19 pandemic outburst triggered enormously all over India. This ill-fated and fatal brawl affected millions of Indian citizens, with many active and infected Indians struggling to recover from this deadly disease to date, leading to a grief situation. The present situation warrants developing a robust and sound forecasting model to evaluate the adversities of the epidemic with reasonable accuracy to assist officials in curbing this hazard. Consequently, we employed Auto-ARIMA, Auto-ETS, Auto-MLP, Auto-ELM, AM, MLP and proposed ELM methods for assessing accumulative infected COVID-19 individuals by the end of July 2021. We made 90 days of advanced forecasting, i.e., up to 24 July 2021, for the number of cumulative infected COVID-19 cases of India using all seven methods in 15 days’ intervals. We fine-tuned the hyper-parameters to enhance the prediction performance of these models and observed that the proposed ELM model offers satisfactory accuracy with MAPE of 5.01, and it rendered better accuracy than the other six models. To comprehend the dataset’s nature, five features are extracted. The resulting feature values encouraged further investigation of the models for an updated dataset, where the proposed model provides encouraging results.

Keywords COVID-19 · Extreme learning · Machine learning · MAPE · RMSE

1 Introduction
The novel Coronavirus (nCoV)-induced diseases, i.e., COVID-19, were first identified and confirmed in China [1–3]. This contagious outbreak propagated very fast and adversely affected 222 countries worldwide [4]. The Severe Acute Respiratory Syndrome Coronavirus-2, i.e., (SARS-CoV-2), is the seventh pathogen in the Coronavirus genus, with a sequence likeness to its progenitor SARS-CoV of 79.6% [3, 5]. The majority of COVID-19 transmission in humans is through tiny droplets having a size between 5 µm < size < 10 µm, or through aerosols (size ≤ 5 µm) induced by close physical contacts [6–10]. Typical symptoms of COVID-19 infections are fever, cough, tiredness, chest pain, dyspnoea, and sore throat [11]. The COVID-19 disease struck more than 150 million people globally, including a fatality of more than 3 million peoples [4]. In contrast, more than 138 million people also reported having recovered from this disease [4].

In late January 2020, COVID-19 received its first confirmation in India. Following two successive incidents of COVID-19 were identified again in February 2020 [12]. After that, this speedy outbreak was disseminated all over India and converted into a pandemic condition. This critical situation adversely touched India and its billion people with indoor imprisonment, social distancing, countrywide lockdowns, travel limitations and restricted outdoor movement [13]. In India, the rise and impact of this deadly infection...
affected more than 23 million peoples. Presently, numerous active COVID-19 cases exist in the country [4]. The limitations of healthcare infrastructure with a large population size present a complex and challenging scenario in India that warrants proper analysis and accurate evaluation of this pandemic disease. In this context, Mathematical or statistical time-based modeling, therefore, becomes necessary for appropriate forecasting of COVID-19 in India. Such forecasting could support the strategy makers, government agencies, and healthcare workforces to organize the future strategy and planning prudently.

A commonly recognized model for projecting and assessing a particular time series established on anterior values, i.e., lags and lagged errors for understanding future trends, is auto-regressive integrated moving average (ARIMA) [14]. This model is effective in sketching autocorrelations present in the dataset [15].

The Abbasov–Mamedova (AM) model is a broadly acknowledged Fuzzy time series model with the edge over other traditional time-series forecasting models. It can forecast out of boundary values, i.e., exterior to the min–max limit of the original dataset [16]. The AM method was initially employed in predicting Azerbaijan’s population [17].

A sequel to a neural-net is multi-layer perceptron (MLP), comprising three layers, namely, input, hidden, and output, to accomplish classification or prediction tasks. The backpropagation learning algorithm is used to train the neurons. The MLP algorithm is often employed for solving linearly non-separable problems [18].

Exponential Smoothing State space methods are a modern systematic framework that combines a spectrum of univariate time series projection techniques [19].

The level, trend (T), and seasonal (S) components coupled with an Error (E) signify the ETS model [20]. As a prediction, ETS computes the weighted mean of each item in the time-series dataset for forecasting. Unlike basic moving average approaches, this method uses weights dependent on the fixed smoothing parameter, which progressively shrinks with time [21].

In their work, [22] proposed the concept of Extreme learning machines (ELM), which is conceptually a feedforward neural network having one or more layers of hidden nodes. ELM is typically deployed to address classification, clustering, regression, compression, sparse approximation, and feature learning problems. The ELM algorithm exhibits generalization performance at a breakneck learning pace, eliminating the requirement of gradient-based backpropagation to achieve a faster learning speed. The ELM algorithm replaces traditional single-layer feed-forward neural net iterations with the Moore–Penrose generalized matrix inverse [23]. ELM’s hidden layer node weights and biases are set at random, thereby saving time on tuning via backpropagation. As a result, it gains information and learns faster than traditional neural networks.

Thus, it is very much relevant to develop a robust and cogent investigation for estimating the COVID-19 misfortunes in India to achieve solid and practical decisions. Due to the intricacy and irrationality of such outbreaks, no one strategy can be effective; we employed the Auto-ARIMA, Auto-ETS, Auto-MLP, Auto-ELM, AM, MLP and proposed ELM methods. We compared the results of these seven forecasting models to pinpoint the one that would accurately anticipate India’s future up until July 24, 2021.

1.1 Motivations and contributions

Many researchers have designed and applied various time-series and statistical-based models for forecasting the COVID-19 pandemic for the upcoming progression. At present, statistical and machine learning (ML)-based time-series methods, such as Linear Regression, polynomial neural networks (PNN), support vector machine (SVM), ARIMA, ELM, ETS, Prophet, Hybridized Deep Learning, and Fuzzy methods considered to forecast several pandemic diseases including the COVID-19 [24–43].

However, these approaches’ effectiveness is usually lessened due to overhead incurred in training these models and hyper-parameter tuning. Moreover, the erratum and incomprehensibility in measuring COVID-19 adversities by employing a solitary strategy can be deleterious, which inspired us to include Auto-ARIMA, Auto-ETS, Auto-MLP, Auto-ELM, AM, MLP and proposed ELM methodologies to develop and propose the best candidate model for forecasting the COVID-19 cumulative infected cases. Some significant offerings of the present study are as follows:

- Our approach was evaluated with six distinct statistical-based time series and ML models to estimate the 461 days of cumulative cases from 21 January 2020 to 25 April 2021.
- To calculate the optimal forecasting outcomes, we fine-tuned hyper-parameters of individual models. We inspected the performance result based on root mean square error (RMSE), mean absolute percentage error (MAPE), and Theil U statistics in the testing phase for further analysis.
- The analysis of advance forecasting of consecutive six stages, each with 15 days’ interval, i.e., a total of 90 days’ advance forecasting done using all seven methods based on the optimal values of their hyper-parameters to recognize the best-fit candidate.
We achieved advanced forecasting of three months, i.e., 26 April 2021 to 24 July 2021, in estimating the no. of cumulative infected cases of COVID-19 in India with reasonable accuracy.

Our proposed ELM method exhibited a favorable outcome in forecasting a highly imprecise pandemic trend in India with a low MAPE of 5. Furthermore, the proposed model exhibited 26% more accuracy than the MLP method proposed in our previous work of [44].

The proposed model validated using an updated dataset up to 28 June 2022.

1.2 Dataset

We collected the cumulative data of the cases of Covid-19 infection in India from 21 January 2020 to 25 April 2021, i.e., 461 days from the World Health Organization (WHO) [6]. The complete set of data was divided into three subsets, where the first subset with data of 400 days, i.e., from 21 January to 23 February 2021 for the training set, the second subset consisting 30 days’ data, i.e., 24 February to 25 March 2021 considered for the validation set. The remaining data of 31 days data, i.e., from 26 March to 25 April 2021, constituted the test set. The forecasting results are divided into six subsets, each with 15 days’ interval. Table 1 lists the date-wise dataset partitions utilized for training, validation, and testing the models.

2 Background study

Alzahrani et al. [45] proposed an ARIMA-based forecasting to examine COVID-19 cases in Saudi Arabia. In a similar work, authors of [14] used ARIMA to indicate the total no. of cumulative COVID-19 pandemic incidents in various countries. Similarly, [46] considered the ARIMA and Holt-Winters-based techniques for assessing COVID-19 cases in India.

In their work, authors of [47] analyzed the adaptability of fuzzy time series (FTS), artificial neural network (ANN), and ARIMA strategies to estimate the no. of newly infected COVID-19 cases and fatalities in India, wherein the FTS approach outperformed the ANN model. They examined data of 107 days and forecasted the outbreak adversities for the next seven days. The authors of [48] employed the FTS and ARIMA models for forecasting pandemic explosions in India and concluded that both the models yielded similar results proximate to the original dataset. By considering a dataset of seventy-seven days, they determined the COVID-19 infected, recovery, and death cases for the next seven days. In related work, authors of [49] considered the ARIMA and FTS models to forecast the COVID-19 infection cases in Egypt, South Africa, and Algeria wherein the ARIMA method harmonized with the data trajectory, whereas, in their work, the authors of [50] developed the ANN-based prediction approach to predict the COVID-19 total, active, and death cases in different states of India.

Gecili et al. [29] employed the ARIMA and cubic smoothing spline models to estimate the verified, mortal, and recovered COVID-19 cases in the USA and Italy. In [30], authors assessed the ARIMA model for predicting the COVID-19 confirmed patients in Korea. They predicted the next 14 days of the cumulative confirmed COVID-19 cases and evaluated their model based on RMSE, MAE, MAPE, and sum of square error (SSE) parameters with a 95% confidence interval. Chordia and Pawar [31] considered ARIMA and Prophet models for predicting the confirmed, death, and recovered COVID-19 cases for India’s five maximum affected states. The authors of [32] also considered the same ARIMA and Prophet models for forecasting the COVID-19 cases in Indonesia. Ganiny and Nisar [33] introduced a forecasting model to predict the pandemic situation in India 30 days ahead.

The authors of [51] developed an AM-based model to forecast Azerbaijan and Vietnam’s population. In another work, [52] employed AM method to forecast rice production in Vietnam with reasonable accuracy.

In [53], the authors applied the MLP approach for forecasting the COVID-19 outbreak. They collected COVID-19 time-series data of thirty countries and developed a model that performs effective six days ahead forecasts. In [54], the authors compared the performances of the MLP, ELM, ARIMA, NNETAR, Holt-Winter, Prophet, BSTS, TBATS, and hybrid models to estimate the amount of fatalities and confirmed COVID-19 instances in Iran. They performed thirty days ahead of forecasting using the models and concluded that MLP performed best for the number of confirmed COVID-19 cases.
cases, whereas Holt-Winter outperformed other models in forecasting the fatality.

In [39], the authors applied the ETS approach to predict the trend of acute hemorrhagic conjunctivitis in China, whereas in [40], the authors proposed an integrated approach based on ARIMA and ETS to forecast human brucellosis in China. In [41], the authors developed a forecast model for the S&P500 stock prices using ARIMA and ETS and reported that the ARIMA outperformed the ETS approach. Naim and Mahara [42] compared seven approaches in forecasting industrial natural gas consumption and perceived that the ARIMA performed better than the ETS. The authors of [43] employed a variant of ARIMA, namely SARIMA and ETS approaches to forecast the influence of COVID-19 on the ISL and RWI Container performance index.

In their work, the authors of [34] proposed ELM-based hybrid approach for forecasting the hydrological time-series data in China. The authors of [35, 36] also employed the ELM approach for developing a forecast model using hydrological time-series data. In another work, [37] employed the

---

**Fig. 1** Research outline

**Table 2** Forecast the COVID-19 outbreak using goodness for fit statistics

| Authors     | Method        | Disease   | Best fit candidate based on |
|-------------|---------------|-----------|----------------------------|
|             | Statistical   | Time series | COVID-19 | RMSE | MAPE | MAE |
| [14]        | ✔             | ✔         | ✔        | ✔    | ✔    | ✔   |
| [29]        | ✔             | ✔         | ✔        | ✔    | ✔    | ✔   |
| [30]        | ✔             | ✔         | ✔        | ✔    | ✔    | ✔   |
| [31]        | ✔             | ✔         | ✔        | ✔    | ✔    | ✔   |
| [32]        | ✔             | ✔         | ✔        | ✔    | ✔    | ✔   |
| [33]        | ✔             | ✔         | ✔        | ✔    | ✔    | ✔   |
| [45]        | ✔             | ✔         | ✔        | ✔    | ✔    | ✔   |
| [46]        | ✔             | ✔         | ✔        | ✔    | ✔    | ✔   |
| [47]        | ✔             | ✔         | ✔        | ✔    | ✔    | ✔   |
| [48]        | ✔             | ✔         | ✔        | ✔    | ✔    | ✔   |
| [49]        | ✔             | ✔         | ✔        | ✔    | ✔    | ✔   |
| [50]        | ✔             | ✔         | ✔        | ✔    | ✔    | ✔   |
| Our work    | ✔             | ✔         | ✔        | ✔    | ✔    | ✔   |
Forecasting adversities of COVID-19 waves in India using intelligent computing

Fig. 2 Architecture of MLP method

Table 3 The MLP model hyper-parameter search space

| Hyper-parameter | Hidden layer | No. of neurons per hidden layer |
|-----------------|--------------|--------------------------------|
| 1               | 1            | 10                             |
| 2               | 2            | 10/20                          |
| 3               | 3            | 10/20/30                       |
| 4               | 4            | 10/20/30/40                    |
| 5               | 4            | 10/20/30/30                    |
| 6               | 4            | 10/20/30/20                    |
| 7               | 4            | 10/20/30/10                    |

ELM method to develop a weighted-average-based ensemble model to forecast Spain’s electricity consumption. In [38], the authors compared the online sequential ELM and online recurrent ELM performances to forecast the passenger count of the New York time-series dataset.

A summary of diverse approaches employed for forecasting COVID-19 cases listed in Table 2.

3 Methodology

An outline of the present study for predicting the aggregate infected COVID-19 instances is portrayed in Fig. 1.

We employed Auto-ARIMA, Auto-ETS, Auto-MLP, Auto-ELM, MLP, AM-FTS and the proposed ELM approach for developing the forecast models. The MLP architecture recommended by [44] is shown in Fig. 2. The search space of the hyper-parameter of MLP architecture recommended in [44] is provided in Table 3.

The AM model’s hyper-parameter search space includes $C = \{0.0001, 0.0002, 0.0003, 0.0004, \ldots, 0.0020\}$, $w = \{2, 3, 4, 5, \ldots, 21\}$, and $n = \{3, 4, 5, 6, \ldots, 22\}$; where $C$ is a constant value, $w$ is a parameter, and $n$ is the number of intervals.

Fig. 3 AM model hyper-parameter tuning
The flowcharts of the techniques deployed for tuning hyper-parameters of the AM and MLP models are presented in Figs. 3 and 4, respectively.

We applied the elm() function of the nnfor package [55] to design the ELM neural network model using the dataset. The inputs to the proposed ELM are automatically selected from the auto-correlated univariate lag values of the time series. In the elm function, we set the lag argument equal to the identified auto-correlated lag values of the data. The setting of sel.lag argument to TRUE enables the function to select the significant lag values automatically. This automatic selection of the input vector is a nonparametric and iterative filter-based approach combining Euclidean distance and MLP proposed by [56].

The autocorrelation in the time series is identified using the acf() function [57]. The frequency of the time series is picked up automatically from the data. Initially, we applied a single hidden layer and obtained the optimized no. of neurons in the hidden layer through the hyper-parameter tuning. In Table 4, the suggested ELM model’s hyper-parameter search space is presented. The proposed ELM model’s architecture and its hyper-parameter tuning technique are shown in Figs. 5 and 6, respectively. We applied lasso with cross-validation to estimate the weights of the output layer. We trained twenty networks to produce the ensemble forecast. The forecasts are combined using the median operator.

We constructed the models using the training set and determined the accuracy of the validation set in hyper-parameter fine-tuning. Consequently, the final models are constructed utilizing the training set and validation set and the test set used for the forecast-accuracy computation. We compared the performances of the developed models using MAPE, RMSE, and Theil’s U statistics. RMSE measures the forecasting model’s expected and significant deviations for ascertaining the degree of error. MAPE outlines a ratio that assesses the forecasting model’s precision, and Theil’s U statistic portrays the relative accuracy between the predicted and the nominal historical outcomes. These metrics’ modest levels indicate the model’s strong accuracy performance. Therefore, we examined these measures to assess the model performance.

The packages along with functions of R used for model development and accuracy estimation: stats(acf) [57], AnalyzeTS(fuzzy.ts2) [58], forecast (auto.arima, ets, accuracy) [59, 60], nnfor (mlp, elm) [55], and DescTools (TheilU) [61].
Forecasting adversities of COVID-19 waves in India using intelligent computing

Table 5 Tuned hyper-parameters of the AM model

| Identified $C$ ($C_i$) | Identified $n$ ($n_i$) | Identified $w$ ($w_i$) |
|------------------------|------------------------|------------------------|
| 0.0002                 | 3                      | 2                      |

Fig. 9 MLP Hyper-parameter tuning

ELM model, i.e., $[51–6000–1]$, shown in Fig. 12. The automatically selected lag values are as follows:

1, 2, 3, 6, 7, 8, 9, 14, 17, 21, 22, 23, 28, 29, 33, 38, 41, 43, 45, 54, 55, 59, 60, 61, 62, 66, 68, 73, 76, 79, 80, 84, 85, 88, 89, 92, 93, 97, 98, 99, 101, 104, 105, 108, 109, 112, 114, 116, 117, 119, 121.

Fig. 8 Autocorrelation of training set

4 Results and discussions

4.1 Hyper-parameter tuning and results

The AM model’s hyper-parameter tuning is presented in Fig. 7.

The results of the hyper-parameter tuning of the AM model are listed in Table 5.

The autocorrelation between the MLP model’s training set and lag values is illustrated in Fig. 8.

Figure 8 exhibits that autocorrelation exists from lag 1 to lag 122. We input these auto-correlated lags in the MLP method recommended in [44]. A plot between the MAPE values and the conforming hyper-parameter of the recommended MLP model in [44] is presented in Fig. 9. It is apparent from Fig. 9 that the lowest MAPE achieved for hyper-parameter value 6. Therefore, we selected it for the final MLP model. The final MLP model’s design, i.e., $[122–\{10, 20, 30, 20\}–1]$, is shown in Fig. 10.

The recommended ELM model’s hyper-parameter tuning is shown in Fig. 11.

Figure 11 reveals that hyper-parameter value 8 exhibits the least MAPE. Consequently, it was selected as the final

4.2 Perspective of forecasting

We conducted an advanced estimate of the cumulative infected patient data of thirty-one days using these models and evaluated the forecast accuracies furnished in Table 6.

We observed from Table 6 that the proposed ELM model outperformed other models and demonstrated the least MAPE, RMSE, and Theil’s U statistics.

Figure 13 displays the empirical values of the last thirty-one days, i.e., 461th to 491th day, together with the values predicted using other models.

Analysis of Fig. 13 reveals that the models’ forecasted value and the observed values are very close for twelve days ahead. However, using the eye-ball approximation, we can conclude that our proposed ELM outperformed other models, also evident from the findings listed in Table 6.

Figure 14 illustrates the forecasting of collective infected COVID-19 cases using the tuned AM, tuned MLP, tuned ELM, Auto-ARIMA, Auto-ETS, Auto-MLP, and Auto-ELM methods The forecast values for the total no. of COVID-19
infected individuals illustrated in Table 7 at intervals of 15, 30, 45, 60, 75, and 90 days.

We observed from Table 7 and Fig. 14 that all the models exhibit a gradual rise in COVID-19 cases in India during the subsequent 90 days. Out of seven tested models, only the MLP model shows a slight bend in the curve during thirty to forty-five days ahead forecast, after which it again increases steadily.

We also observed that none of the models exhibited flattening of the curve in the next ninety days. After fifteen days, the AM-FTS model forecasts approximately twenty-one million cumulative infection cases in India, while the Auto-ARIMA, Auto-ETS, Auto-ELM, MLP and proposed ELM forecast approximately twenty-two million cases in India. We observed that after ninety days, the proposed ELM forecasts approximately forty-three million cumulative infected cases in India. In contrast, the minimum forecast is approximately forty million by the MLP, and the maximum forecast is fifty-two million by the Auto-MLP approach.

The performance of MAPE of the proposed model compared with the [29] and [62, 63] is given in Table 8.

5 Recommendation of the proposed model and future scopes

5.1 Feature extraction and novelty of the proposed ELM model

Shannon entropy is a time series forecastability metric, with low values indicating a high signal-to-noise ratio and high
values indicating a high tribulation level in a series forecasting. Higher Hurst exponent values in \([0, 1]\) indicate a smoother trend with nominal volatility. In contrast, the nonlinearity coefficient for a nonlinear series is significant, whereas it is nearly zero for a linear series. Consequently, we extracted the Shannon-entropy, Hurst coefficient, nonlinearity coefficient, linearity, and curvature features \([64–66]\) from the training set consisting of the data of 400 days. The low Shannon entropy of 0.1573 indicates a high signal-to-noise ratio. The data displayed a smoother trend and less volatility with a high Hurst coefficient of 0.9999, positive linearity of 18.8447, and positive curvature of 4.715. Also, a significant nonlinearity coefficient of 35.6907 indicates that the training data are nonlinear. The tsfeatures package \([67]\) is employed to extract these features.

During the input feature selection process for the ELM method, we discovered that the autocorrelation feature exists between lags 1 and 122, as shown in Fig. 8. As a result, we first entered these 122 auto-correlated lags into our proposed ELM method and used the automatic lag selection strategy of the elm() function of the nnfor package \([55]\) to determine the best possible combination of these lags for network construction. The automatic input lag, i.e., input feature selection, heavily relies on a data-driven Iterative
Fig. 14 Forecasting 90 days ahead using all models

| Date (days ahead) | Auto-ARIMA | Auto-ETS | Auto-MLP | Auto-ELM | AM | MLP | Proposed ELM |
|-------------------|------------|----------|----------|----------|----|-----|--------------|
| 01/05/2021 (15 days) | 22,207,830 | 22,205,554 | 22,683,523 | 22,306,810 | 21,275,569 | 22,200,823 | 22,173,573 |
| 25/05/2021 (30 days) | 27,455,488 | 27,450,935 | 28,506,915 | 27,700,010 | 25,641,856 | 26,757,582 | 26,747,920 |
| 09/06/2021 (45 days) | 32,703,147 | 32,696,317 | 34,330,965 | 33,101,859 | 30,008,143 | 31,294,725 | 31,206,435 |
| 24/06/2021 (60 days) | 37,950,805 | 37,941,699 | 40,155,020 | 38,504,446 | 33,458,681 | 35,543,906 | 35,543,906 |
| 09/07/2021 (75 days) | 43,198,463 | 43,187,081 | 45,979,075 | 43,907,054 | 37,623,883 | 39,642,676 | 39,642,676 |
| 24/07/2021 (90 days) | 48,446,121 | 48,432,463 | 51,803,130 | 49,309,664 | 39,826,764 | 42,867,446 | 42,867,446 |
Neural Filtering (INF) technique \[56, 68\] for robust analysis and automatic feature evaluation of the time series. A brief overview of the INF algorithm is as follows:

**Algorithm INF: Input Feature Selection**

1. **Calculate** Penalised Euclidian Distance (PEuD)
2. If (Minimum PEuD != 1)
   2.1 **goto** step 3
   Else
   2.2 **goto** step 8
3. **Identify** Seasonality (S)
4. **Update** INF for S
5. **Fit** INF to model S
6. **Subtract** INF’s output from original time-series
7. **goto** step 1
8. **End**

The mechanisms of the automatic input selection as follows:

- Identification of the time series frequencies using INF
- Differentiating the stochastic and deterministic parts
- Application of pruned regression
- Identifying the inputs.

The final ELM architecture in this study retained 51 lag values while dropping 71 lag values using the automatic input selection strategy. The retained lag values are specified in Sect. 4.1. To obtain a novel and optimized ELM architecture, we use hyper-parameter tuning to acquire the optimal number of hidden nodes from the ELM search space and the automatic input feature, i.e., lag, selection technique to select the input lags from the set of auto-correlated lag values initially fed to the ELM’s input layer.

Consequently, the methodology for tuning the hyper-parameters in sync with the automatic input feature (lag) selection technique offers the optimal combination of the number of hidden nodes and input features, i.e., network lag values. In contrast, a simple hyper-parameter tuned ELM finds only the optimal number of hidden nodes from the search space while keeping the number of input features, i.e., lags, constant. The empirical evidence, in this case, revealed that the optimized ELM outperformed the simple hyper-parameter tuned ELM, i.e., the ELM with all 122 lag values in the input, in terms of MAPE. The simple ELM’s out-of-sample MAPE obtained on the test data is 5.72, whereas the proposed ELM’s MAPE on the Test data is 5.01. Therefore, the optimized ELM provides a 12.41% increase in efficiency based on MAPE.

### 5.2 Forecasting performance

All the seven models have satisfactory forecast accuracy and fitting bidders to forecast total no. of COVID-19 instances in India, wherein the proposed ELM model is outperformed other models in terms of benchmark performance metrics for producing thirty-one days ahead prediction. All seven forecast models were employed to evaluate the cumulative COVID-19 cases in India for six intervals, each with 15 days totaling 90 days. These forecast estimates can be conducive to government authorities’ decision-making, developing recommendation systems, and vaccine administration to flatten the curve of the COVID-19 hazard. The architecture of the proposed model on COVID-19 data can be extended to forecast the spread of COVID-19 across the globe, which can gain valuable information about the outbreak of this deadly disease and guide all the stakeholders to take appropriate and prompt actions. The multifaceted aspect of the proposed model can also be applied to the financial time-series data, economic time-series data, and various other time-series data.

We exhibited the observed data of COVID-19 cumulative cases in India with forecasted values using the proposed ELM and MLP approaches in Table 9 and Fig. 15. In the present work, we elucidated the performance comparison between the proposed ELM and MLP (recommended by \[44\]) and observed that the forecast performance of the MLP, when applied to the employed dataset, exhibits higher MAPE, i.e., MAPE $= 6.83$ than our proposed ELM approach where MAPE $= 5.01$. Therefore, the proposed ELM model achieved an approximately 26.65% decrease in MAPE and a 22.69% decrease in RMSE.

### 5.3 Model validation

The baseline dataset of 461 days is extended by considering India’s COVID-19 infection data up to June 28, 2022, to validate the performance of the suggested ELM and MLP approach. As a result, the data length increased by 93.1% (from 461 to 890 days). We obtain Shannon entropy $=$
Table 9: Comparison of observed and forecast data of cumulative infected in India

| Dates    | No. of cumulative infected in India | Proposed ELM forecast | MLP forecast |
|----------|------------------------------------|-----------------------|--------------|
| 26-Apr-2021 | 17,313,163                        | 17,309,633            | 17,306,865   |
| 27-Apr-2021 | 17,636,307                        | 17,653,340            | 17,651,906   |
| 28-Apr-2021 | 17,977,267                        | 18,005,422            | 18,003,790   |
| 29-Apr-2021 | 18,376,524                        | 18,363,960            | 18,352,890   |
| 30-Apr-2021 | 18,762,976                        | 18,722,316            | 18,702,057   |
| 1-May-2021  | 19,164,969                         | 19,080,802            | 19,059,405   |
| 2-May-2021  | 19,557,457                         | 19,438,671            | 19,416,743   |
| 3-May-2021  | 19,925,604                         | 19,794,285            | 19,767,428   |
| 4-May-2021  | 20,282,833                         | 20,138,768            | 20,120,755   |
| 5-May-2021  | 20,665,148                         | 20,488,378            | 20,477,953   |
| 6-May-2021  | 21,077,410                         | 20,842,869            | 20,829,175   |
| 7-May-2021  | 21,491,598                         | 21,190,932            | 21,173,548   |
| 8-May-2021  | 21,892,676                         | 21,518,052            | 21,531,336   |
| 9-May-2021  | 22,296,414                         | 21,838,422            | 21,870,322   |
| 10-May-2021 | 22,662,575                         | 22,173,573            | 22,200,823   |

0.0726, Hurst coefficient = 0.9999, nonlinearity coefficient = 1.94, linearity = 29.0, and curvature = 1.83 feature values for the training set based on the updated dataset. It suggests that the data have a high signal-to-noise ratio. The data showed a smoother trend and less volatility with a high Hurst coefficient, positive linearity, and positive curvature. The training data are nonlinear, as evidenced by the nonlinearity coefficient. The training, validation, and test set lengths are 829, 30, and 31 days, respectively, wherein the lengths of validation and test sets are kept the same as the initial dataset of 461 days, i.e., 30 and 31 days, respectively, to recalibrate the proposed models. The data for the training set span 829 days, i.e., 21 January 2020 to 28 April 2022; the data for the validation set spans 30 days, i.e., 29 April to 28 May 2022; and the data for the test set span 31 days, i.e., 29 May to 28 June 2022.

To create the optimum models for the revised dataset, we first fed the MLP and ELM with the auto-correlated 287 lags and tuned them to find the ideal ratio of lags to hidden neurons. The ELM’s architecture depends heavily on data. Combining the INF algorithm’s application and hyper-parameter tuning stimulates the optimum ELM model. Hyper-parameter tuning is utilized to choose the number of hidden neurons from the search space, and the INF method is used to automatically pick the input lags from the initial input of auto-correlated lags. Figure 16 exhibits autocorrelation from lag 1 to lag 287. Plots of the MAPE against the relevant hyper-parameter for the MLP and ELM models are displayed in Figs. 17 and 18, respectively. As per Fig. 17, the lowest MAPE achieved for hyper-parameter value three. Consequently, we chose it for the conclusive MLP model. After hyper-parameter tuning, the final MLP architecture is [287-10/20/30-1]. Figure 18 reveals that the lowest MAPE achieved for hyper-parameter value three. Therefore, we selected it for the final ELM model, and [64–1000–1] is the

Fig. 15 Graphical representation of observed data, MLP forecast, and proposed ELM forecast

Fig. 16 Autocorrelation of training set data of the updated series

Fig. 17 Hyper-parameter tuning of the MLP for updated dataset with MAPE
Forecasting adversities of COVID-19 waves in India using intelligent computing

Table 10 Performance of MLP and ELM for updated dataset

| Model | Description                                                                 | RMSE     | MAPE   | Theil’s U statistics |
|-------|-----------------------------------------------------------------------------|----------|--------|----------------------|
| MLP   | Optimized MLP obtained for the original 461 days dataset and applied to the new dataset. Model architecture: [122–{10, 20, 30, 20}–1] | 2,322,463 | 3.95   | 0.0537               |
|       | Optimized MLP obtained for the new 890 days dataset and applied to the new dataset. Model architecture: [287–{10/20/30}–1] | 294,104.6 | 0.55   | 0.0068               |
| ELM   | Optimized ELM obtained for the original 461 days dataset and applied to the new dataset. Model architecture: [51–6000–1] | 1,043,827 | 1.82   | 0.0241               |
|       | Optimized MLP obtained for the new 890 days dataset and applied to the new dataset. Model architecture: [64–1000–1] | 808,488   | 1.40   | 0.0187               |

Table 11 Performance comparison in forecast horizon of 31 days

| Model | Dataset length | Forecast horizon | RMSE     | MAPE   | Theil’s U statistics |
|-------|----------------|------------------|----------|--------|----------------------|
| MLP   | 461 days       | 31 days          | 1,487,959 | 6.83   | 0.1085               |
|       | 890 days       | 31 days          | 294,104.6 | 0.55   | 0.0068               |
| ELM   | 461 days       | 31 days          | 1,150,395 | 5.01   | 0.0839               |
|       | 890 days       | 31 days          | 808,488   | 1.40   | 0.0187               |

Fig. 18 Hyper-parameter tuning of the proposed ELM for updated dataset with MAPE

ELM architecture obtained for 461 days’ dataset is [51–6000–1], whereas for 890 days dataset the architecture changed to [64–1000–1]. Therefore, it is imperative that the suggested ELM model architecture of largely depends on data, yet the ELM architecture transition becomes automatic by employing INF algorithm and hyper-parameter tuning for selecting auto-correlation lags in a noble fashion thereby reducing manual intervention.

The forecasting accuracy of the models with MAPE < 10 is promising. The performance metrics obtained for the prior and modified datasets are reported in Tables 10 and 11. Table 10 attests that the newly trained MLP model’s MAPE efficiency increased by 86.08%. The ELM approach also demonstrated a 23.08% improvement in efficiency. Table 11 shows that, compared to the earlier performances of these models in the 31-day forecast horizon, the MLP model’s efficiency has grown by 91.95%, and the ELM model’s efficiency has increased by 72.06%. The comparison between the previous 90 days of forecasting data, i.e., 24 July 2021, offered by all seven models and 90 days of observed data are shown in Fig. 19. In contrast, Fig. 20 depicts a forecasting comparison between the proposed ELM and MLP based on test data, i.e., a 31-day forecast horizon up to June 28th, 2022, to demonstrate the efficacy of the proposed approaches.
Fig. 19 90 days forecast vs observed for 7 models
Fig. 20 Updated dataset forecast vs observed 31 days

6 Conclusion

This study used a dataset of 461 days for experimentation to examine the efficacy of seven models, including Auto-ARIMA, Auto-ETS, Auto-MLP, Auto-ELM, AM-FTS, and proposed ELM method. With moderate RMSE, MAPE, and Theil’s U statistics, the suggested ELM model outperformed other models and extended positive outcomes. We determined that the specified ELM model’s thirty-one days ahead predicted MAPE is 5.01%, i.e., noticeably low. All models also displayed impressive forecast accuracy, with a MAPE of less than 10 for the forecast for the next 31 days. The models’ predicted and actual values are comparable to fifteen days ahead, i.e., from 431st to 445th day. After this day, i.e., the 446th day, we noticed an increase in COVID-19 instances in India. Any used models could not account for this abrupt upward movement trend, although the analysis found that all models had good forecast accuracy with MAPE < 10. The models’ forecasting accuracy for the updated dataset up to 28th June 2022 is promising. With a low MAPE, newly trained MLP and ELM’s efficiency is enhanced by 86.08% and 23.08%, respectively. The MLP model’s efficiency has increased by 91.95% compared to previous performances in the 31-day forecast horizon, while the ELM model’s efficiency has increased by 72.06%.

Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

References

1. Wu A, Peng Y, Huang B, Ding X, Wang X, Niu P, Meng J, Zhu Z, Zhang Z, Wang J, Sheng J, Quan L, Xia Z, Tan W, Cheng G, Jiang T (2020) Genome composition and divergence of the novel coronavirus (2019-nCoV) originating in China. Cell Host Microbe 27(3):325–328. https://doi.org/10.1016/j.chom.2020.02.001
2. Zhou P, Yang XL, Wang XG, Hu B, Zhang L, Zhang W, Si HR, Zhu Y, Li B, Huang CL, Chen HD, Chen J, Luo Y, Guo H, Jiang RD, Liu MQ, Chen Y, Shen XR, Wang X, Zheng XS, Zhao K, Chen QJ, Deng F, Liu LL, Yan B, Zhan FX, Wang YY, Xiao GF, Shi ZL (2020) A pneumonia outbreak associated with a new coronavirus of probable bat origin. Nature 579(7798):270–273. https://doi.org/10.1038/s41586-020-2012-7
3. Zhu N, Zhang D, Wang W, Li X, Yang B, Song J, Zhao X, Huang B, Shi W, Lu R, Niu P, Zhan F, Ma X, Wang D, Xu W, Wu G, Gao GF, Tan W (2020) China Novel Coronavirus Investigating and Research Team. A novel coronavirus from patients with pneumonia in China, 2019. N Engl J Med 382(8):727–733. https://doi.org/10.1056/NEJMoa2001107
4. Worldometer—Real Time World Statistics. https://www.worldometers.info/coronavirus/. Accessed 12 May 2021
5. Su S, Wong G, Shi W, Liu J, Lai ACK, Zhou J, Liu W, Bi Y, Gao GF (2016) Epidemiology, genetic recombination, and pathogenesis of coronaviruses. Trends Microbiol 24(6):490–502. https://doi.org/10.1016/j.tim.2016.03.003
6. WHO | Novel Coronavirus—China. Situation report archived from WHO. https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports. Accessed 29 June 2022
7. Li Q, Guan X, Wu P, Wang X, Zhou L, Tong Y, Ren R, Leung KSM, Lau EHY, Wong JY, Xing X, Xiang N, Wu Y, Li C, Chen Q, Li D, Liu T, Zhao J, Liu M, Tu W, Chen C, Jin L, Yang R, Wang Q, Zhou S, Wang R, Liu H, Luo Y, Liu Y, Shao G, Li H, Tao Z, Yang Y, Deng Z, Liu B, Ma Z, Zhang Y, Shi G, Lam TTY, Wu JT, Gao GF, Cowling BJ, Yang B, Leung GM, Feng Z (2020) Early transmission dynamics in Wuhan, China, of novel coronavirus-infected pneumonia. N Engl J Med 382(13):1199–1207. https://doi.org/10.1056/NEJMoa2001316
8. Chen N, Zhou M, Dong X, Qu J, Gong F, Han Y, Qiu Y, Wang J, Liu Y, Wei Y, Xia J, Yu T, Zhang X, Zhang L (2020) Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: a descriptive study. Lancet 395(10223):507–513. https://doi.org/10.1016/S0140-6736(20)30211-7
9. Patel A, Jernigan DB, 2019 nCoV CDC Response Team (2020) Initial public health response and interim clinical guidance for the 2019 novel coronavirus outbreak—United States. Morb Mortal Wkly Rep (MMWR) 69(5):140–146
10. Singhal T (2020) A review of coronavirus disease-2019 (COVID-19). Indian J Pediatr 87(4):281–286. https://doi.org/10.1007/s12098-020-03263-6
11. Huang C, Wang Y, Li X, Ren L, Zhao J, Hu Y, Zhang L, Fan G, Xu J, Gu X, Cheng Z, Yu T, Xia J, Wei Y, Wu W, Xie X, Yin W, Li H, Liu M, Xiao Y, Gao H, Luo L, Xie J, Wang G, Jiang R, Gao Z, Jin Q, Wang J, Cao B (2020) Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. Lancet 395(10223):497–506. https://doi.org/10.1016/S0140-6736(20)30183-6
12. Reid D (30 January 2020). India confirms its first coronavirus case, CNCB. https://www.cnb.com/2020/01/30/india-confirms-first-case-of-the-coronavirus.html. Accessed 7 May 2021
13. Di RL, Gualtieri P, Pivari F, Soldati L, Attinà A, Cinelli G, Leggeri KSM, Lau EHY, Wong JY, Xing X, Xiang N, Wu Y, Li C, Chen Q, Li D, Liu T, Zhao J, Liu M, Tu W, Chen C, Jin L, Yang R, Wang Q, Zhou S, Wang R, Liu H, Luo Y, Liu Y, Shao G, Li H, Tao Z, Yang Y, Deng Z, Liu B, Ma Z, Zhang Y, Shi G, Lam TTY, Wu JT, Gao GF, Cowling BJ, Yang B, Leung GM, Feng Z (2020) Early transmission dynamics in Wuhan, China, of novel coronavirus-infected pneumonia. N Engl J Med 382(13):1199–1207. https://doi.org/10.1056/NEJMoa2001316
14. Sahai AK, Rath N, Sood V, Singh MP (2020) ARIMA modelling & forecasting of COVID-19 in top five affected countries. Diabetes Metab Syndr 14(5):1419–1427. https://doi.org/10.1016/S0140-6736(20)30183-6
15. Otextis.com. (2019) Chapter 8 ARIMA models for forecasting: principles and practice. https://otexts.com/fpp2/arima.html. Accessed 7 May 2021
16. Thao NT (2019) An improved fuzzy time series forecasting model using the differential evolution algorithm. J Intell Fuzzy Syst 36(2):1727–1741. https://doi.org/10.3233/JIFS-18636
17. Abbasov AM, Mamedova MH (2003) Application of fuzzy time series to population forecasting. Vienna Univ Technol 12:545–552
18. Abirami S, Chitra P (2020) Energy-efficient edge based real-time healthcare support system. Adv Comput 117(1):339–368. https://doi.org/10.1016/j.acom.2019.09.007
19. Free Range Statistics (2016) Error, trend, seasonality—ETS and its forecast model friends. http://friarangestatistics.info/blog/2016/11/27/ets-friends. Accessed 7 May 2021
20. Statsmodels.org (2019) ETS models. https://www.statsmodels.org/devel/examples/notebooks/generated/ets.html. https://www.statsmodels.org/devel/examples/notebooks/generated/ets.html. Accessed 7 May 2021
21. Amazon.com (2019) Exponential smoothing (ETS) algorithm—Amazon forecast. https://docs.aws.amazon.com/forecast/latest/dg/aws-forecast-recipe-ets.html. Accessed 7 May 2021
22. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputings 70(1–3):489–501. https://doi.org/10.1016/j.neucom.2005.12.126
23. Erdem (burnpiro) K (29 May 2020) Introduction to extreme learning machines. medium. https://towardsdatascience.com/introduction-to-extreme-learning-machines-c020020ff82b. Accessed 7 May 2021
24. Fong SJ, Li G, Dey N, Crespo RG, Herrera-Viedma E (2020) Corona virus disease 2019. PLoS ONE 15(10):e0239960. https://doi.org/10.1371/journal.pone.0239960
25. Arzabili SF, Mosavi A, Ghamisi P, Ferdinand F, Varkonyi-Koczy AR, Reuter U, Rabczuk T, Akinfon FM (2020) Covid-19 outbreak prediction with machine learning. Algorithms 13(10):249. https://doi.org/10.3390/a13100249
26. Fong SJ, Li G, Dey N, Crespo RG, Herrera-Viedma E (2020) Composite monte carlo decision making under high uncertainty of novel coronavirus epidemic using hybridized deep learning and fuzzy rule induction. Appl Soft Comput J 93:106282. https://doi.org/10.1016/j.asoc.2020.106282
27. Niu WJ, Feng ZK, Chen YB, Zhang HR, Cheng CT (2020) Annual streamflow time series prediction using extreme learning machine based on gravitational search algorithm and variational mode decomposition. J Hydrol Eng 25(5):04020008. https://doi.org/10.1061/(ASCE)HE.1943-5584.0001902
28. Fong ZK, Niu WJ, Tang ZY, Xu Y, Zhang HK (2021) Evolutionary artificial intelligence model via cooperation search algorithm and extreme learning machine for multiple scales nonstationary hydrological time series prediction. J Hydrol 595:126062. https://doi.org/10.1016/j.jhydrol.2021.126062
29. Niu WJ, Feng ZK, Zeng M, Feng BF, Min YW, Cheng CT, Zhou JZ (2019) Forecasting reservoir monthly runoff via ensemble empirical mode decomposition and extreme learning machine optimized by an improved gravitational search algorithm. Appl Soft Comput 82:105589. https://doi.org/10.1016/j.asoc.2019.105589
30. Lee DH, Kim YS, Koh YY, Song KY, Chang IH (2021) Forecasting the amount of pneumonia patients in Jakarta with weighted high order fuzzy time series to population forecasting. Vienna Univ Technol 12:545–552
31. Niu WJ, Feng ZK, Zhang M, Feng BF, Min YW, Cheng CT, Zhou JZ (2019) Forecasting reservoir monthly runoff via ensemble empirical mode decomposition and extreme learning machine optimized by an improved gravitational search algorithm. Appl Soft Comput 82:105589. https://doi.org/10.1016/j.asoc.2019.105589
32. Satrioa CBA, Darmawana W, Nadiaa BU, Hanafiahb N (2021) Finding an accurate early forecasting model from small dataset: a case of 2019-ncov novel coronavirus outbreak. Int J Interact Multimed Artif Intell 6(1):132–140. https://doi.org/10.9781/ijimai.2020.02.002
33. Abbasov AM, Mamedova MH (2003) Application of fuzzy time series to population forecasting. Vienna Univ Technol 12:545–552
34. Niu WJ, Feng ZK, Chen YB, Zhang HR, Cheng CT (2020) Annual streamflow time series prediction using extreme learning machine based on gravitational search algorithm and variational mode decomposition. J Hydrol Eng 25(5):04020008. https://doi.org/10.1061/(ASCE)HE.1943-5584.0001902
35. Feng ZK, Niu WJ, Tang ZY, Xu Y, Zhang HK (2021) Evolutionary artificial intelligence model via cooperation search algorithm and extreme learning machine for multiple scales nonstationary hydrological time series prediction. J Hydrol 595:126062. https://doi.org/10.1016/j.jhydrol.2021.126062
36. Niu WJ, Feng ZK, Zeng M, Feng BF, Min YW, Cheng CT, Zhou JZ (2019) Forecasting reservoir monthly runoff via ensemble empirical mode decomposition and extreme learning machine optimized by an improved gravitational search algorithm. Appl Soft Comput 82:105589. https://doi.org/10.1016/j.asoc.2019.105589
37. Larrea M, Porto A, Irigoyen E, Barragán AJ, Andújar JM (2020) Extreme learning machine ensemble model for time series forecasting boosted by PSO: Application to an electric consumption problem. Neurocomputing. https://doi.org/10.1016/j.neucom.2019.12.140
38. Park J, Kim J (2017) Online recurrent extreme learning machine and its application to time-series prediction. In: 2017 International joint conference on neural networks (IJCNN). IEEE, pp 1983–1990. https://doi.org/10.1109/IJCNN.2017.7966094
39. Liu H, Li C, Shao Y, Zhang X, Zhai Z, Wang X, Qi X, Wang J, Hao Y, Wu Q, Jiao M (2020) Forecast of the trend in incidence of acute hemorrhagic conjunctivitis in China from 2011–2019 using the seasonal autoregressive integrated moving average (SARIMA) and exponential smoothing (ETS) models. J Infect Public Health 13(2):287–294. https://doi.org/10.1016/j.jiph.2019.12.008
40. Wang Y, Xu C, Zhang S, Wang Z, Zhu Y, Yuan J (2018) Temporal trends analysis of human brucellosis incidence in mainland China from 2004 to 2018. Sci Rep 8(15901):1–11. https://doi.org/10.1038/s41598-018-33165-9
41. Sun Z (2020) Comparison of trend forecast using ARIMA and ETS Models for S&P500 close price. In: The 4th international conference on e-business and internet, pp 57–60. https://doi.org/10.1145/3436209.3436894
42. Naim I, Mahara T (2018) Comparative analysis of univariate forecasting techniques for industrial natural gas consumption. Int J Image Graph Signal Process 10(5):33–44. https://doi.org/10.5815/ijigsp.2018.05.04
43. Koyuncu K, Tavacioglu L, Gokmen N, Arican UC (2021) Forecasting COVID-19 impact on RWI/ISL container throughput index by using SARIMA models. Marit Policy Manag. https://doi.org/10.1080/03088839.2021.1876937
44. Chakraborty A, Mitra S, Das D, De D, Pal AJ (2021) Forecasting COVID-19 outbreak in India using time series dataset: an ensemble of ARIMA, Abbasov-Mamedova, and multilayer perceptron models. In: 6th International conference on emerging applications of information technology (EAIT 2020)
45. Alzahrani SI, Aljamaan IA, Al-Fakih EA (2020) Forecasting the spread of the COVID-19 pandemic in Saudi Arabia using ARIMA prediction model under current public health interventions. J Infect Public Health 13(7):914–919. https://doi.org/10.1016/j.jiph.2020.06.001
46. Panda M (2020) Application of ARIMA and Holt-Winters forecasting model to predict the spreading of COVID-19 for India and its states. medRxiv. https://doi.org/10.1101/2020.07.14.20153908
Forecasting adversities of COVID-19 waves in India using intelligent computing

47. Mishra P, Fatih C, Rawat D, Sahu S, Pandey SA, Ray M, Dubey A, Sanusi OM (2020) Trajectory of COVID-19 data in India: investigation and project using artificial neural network, fuzzy time series and ARIMA models. Annu Res Rev Biol 35(9):46–54. https://doi.org/10.9734/arrb/2020/v35i930270

48. Verma P, Khetan M, Dwivedi S, Dixit S (2020) Forecasting the COVID-19 outbreak: an application of ARIMA and fuzzy time series models. Research Square. https://doi.org/10.21203/rs.3.rs-36585/v1

49. Fatih C, Hamimes A, Mishra P (2020) Covid-19 statistics, strange trend and forecasting of total cases in the most infected african countries: an ARIMA and fuzzy time series approaches. https://doi.org/10.13140/RG.2.2.34158.97603

50. Farooq J, Bazaz MA (2021) A deep learning algorithm for modeling and forecasting of COVID-19 in five worst affected states of India. Alex Eng J 60(1):587–596. https://doi.org/10.1016/j.aej.2020.09.037

51. Che-Ngoc H, Vo-Van T, Huynh-Le QC, Ho V, Nguyen-Trang T, Chu-Thi MT (2018) An improved fuzzy time series forecasting model. In: International econometric conference of Vietnam. Springer, Cham, pp 474–490. https://doi.org/10.1007/978-3-319-73150-6_38

52. Khan MZ, Khan MF (2019) Application of ANFIS, ANN and fuzzy time series models to CO2 emission from the energy sector and global temperature increase. Int J Clim Chang Strateg Manag 11(5):622–642. https://doi.org/10.1108/IJCCSM-01-2019-0001

53. Borghi PH, Zakordonets O, Teixeira JP (2021) A COVID-19 time series forecasting model based on MLP ANN. Procedia Comput Sci 181:940–947. https://doi.org/10.1016/j.procs.2021.01.250

54. Talkhi N, Akhavan FN, Ateei Z, Jabbari NM (2021) Modeling and forecasting number of confirmed and death caused COVID-19 in IRAN: a comparison of time series forecasting methods. Biomed Signal Process Control 66:102494. https://doi.org/10.1016/j.bspc.2021.102494

55. Kourentzes N (2019) nnfor: Time Series forecasting with neural networks, R package version 0.9.6. https://CRAN.R-project.org/package=nnfor. Accessed 7 May 2021

56. Hyndman R, Athanasopoulos G, Bergmeir C, Caceres G, Chhay L, OHara-Wild M, Petropoulos F, Razbash S, Wang E, Yasmeen F (2020) forecast: Forecasting functions for time series and linear models, R package version 8.13. https://pkg.robhyndman.com/forecast. Accessed 7 May 2021

57. Hyndman RJ, Khandakar Y (2008) Automatic time series forecasting: the forecast package for R. J Stat Softw 27(3):1–22. https://doi.org/10.18637/jss.v027.i03

58. Signorell A et al (2020) DescTools: Tools for descriptive statistics, R package version 0.99.38. https://cran.r-project.org/package=DescTools. Accessed 7 May 2021

59. Saif S, Das P, Biswas SA (2021) Hybrid model based on mBA-ANFIS for COVID-19 confirmed cases prediction and forecast. J Inst Eng India Ser B. https://doi.org/10.1007/s40031-021-00538-0

60. Gola A, Arya RK, Animesh, Dugh R (2020) Review of forecasting models for coronavirus (COVID-19) pandemic in India during country-wise lockdowns. medRxiv. https://doi.org/10.1101/2020.08.03.20167254

61. Karaca Y, Zhang YD, Muhammad K (2020) Characterizing complexity and self-similarity based on fractal and entropy analyses for stock market forecast modelling. Expert Syst Appl 144(113098). https://doi.org/10.1016/j.eswa.2019.113098

62. Papacharalampous G, Tyralis H (2022) Feature-based clustering of hydroclimatic time series. Copernicus Meetings. EGU General Assembly 2022, Vienna, Austria, EGU22-937. https://doi.org/10.1002/egusphere-egu22-937

63. Papacharalampous G, Tyralis H (2022) Time series features for supporting hydrometeorological explorations and predictions in ungauged locations using large datasets. Water 14(10):1657. https://doi.org/10.3390/w14101657

64. Hyndman R, Kang Y, Montero-Manso P, Talagala T, Wang E, Yang Y, O’Hara-Wild M, Petropoulos F, Kim J, Kalinin A, O’Hara-Wild M, Petropoulos F, Razbash S, Wang E, Yasmeen F (2021) mts: Time Series features extraction, R package version 1.0.2. https://pkg.robhyndman.com/mts

65. Kourentzes N, Crone SF (2010) Frequency independent automatic input variable selection for neural networks for forecasting. In: The 2010 international joint conference on neural networks (IJCNN). IEEE, pp 1–8. https://doi.org/10.1109/IJCNN.2010.5596637

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

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.