ORIGINAL ARTICLE

A novel approach based on combining deep learning models with statistical methods for COVID-19 time series forecasting

Hossein Abbasimehr[1] · Reza Paki[1,3] · Aram Bahrini[2]

Abstract

The COVID-19 pandemic has disrupted the economy and businesses and impacted all facets of people's lives. It is critical to forecast the number of infected cases to make accurate decisions on the necessary measures to control the outbreak. While deep learning models have proved to be effective in this context, time series augmentation can improve their performance. In this paper, we use time series augmentation techniques to create new time series that take into account the characteristics of the original series, which we then use to generate enough samples to fit deep learning models properly. The proposed method is applied in the context of COVID-19 time series forecasting using three deep learning techniques, (1) the long short-term memory, (2) gated recurrent units, and (3) convolutional neural network. In terms of symmetric mean absolute percentage error and root mean square error measures, the proposed method significantly improves the performance of long short-term memory and convolutional neural networks. Also, the improvement is average for the gated recurrent units. Finally, we present a summary of the top augmentation model as well as a visual representation of the actual and forecasted data for each country.

Keywords Deep learning · Time series forecasting · Augmentation methods · COVID-19 pandemic

1 Introduction

Temporary interventions such as social distancing, self-isolating, quarantining, and shutting down nonessential activities have been strategies for the governments to prevent the virus from spreading. It is essential to forecast the number of infected cases using different data types to notify public health decision-makers by estimating the likely impact of the COVID-19 pandemic and plan accordingly [1–4].

Deep learning models have demonstrated successful performance in language and image processing tasks [6–8].

Also, they exhibited state-of-the-art performance in forecasting complex time series data [5, 9–12]. The main advantage of deep learning models is their ability to learn representations from raw input data. Among the most popular deep learning algorithms, long short-term memory (LSTM) and bidirectional LSTM (Bi-LSTM) [13] have been used in [14–17], with significant results in COVID-19 forecasting. LSTM is a special type of recurrent neural networks (RNNs), which is developed to learn temporal information from sequential data [18].

Despite the fact that deep learning algorithms can reach acceptable performance in time series forecasting, particularly in COVID-19 forecasting applications, their forecasting capability is primarily dependent on the amount of data available to fit their parameters appropriately [12, 19]. Another challenge with deep learning for time series forecasting is that, even though adequate data samples are available, data from the distant past are typically less useful for forecasting [12]. In other words, in predicting, recent observations of an individual series are more valuable. This may be due to shifts in patterns that formerly occurred in a series.

✉ Hossein Abbasimehr
   abbasimehr@azaruniv.ac.ir

1 Faculty of Information Technology and Computer Engineering, Azarbaijan Shahid Madani University, Tabriz, Iran

2 Department of Engineering Systems and Environment, University of Virginia, Charlottesville, Virginia, USA

3 Present Address: School of Industrial and Information Engineering, Politecnico di Milano University, Milano, Italy

To overcome the aforementioned issue and increase the performance of deep learning models in time series forecasting, we propose exploiting time series augmentation techniques [19–21] to generate new series with similar temporal dependencies as the original series. We then extract new samples from the augmented time series to enhance model training. Three deep learning models based on the LSTM, gated recurrent units (GRU) [22], and convolutional neural network (CNN) [23] are used to see whether the proposed approach is useful. A multi-step-ahead forecasting strategy [24] is used to develop the models, allowing them to predict the number of cases for the next few days. It is a preferable alternative to single-step-ahead forecasting for long-horizon forecasting [25].

The proposed models are applied to COVID-19 data from the top 10 countries with the most reported confirmed cases from January 20, 2020, until March 28, 2021. We show that the proposed method significantly improves the performance of the LSTM-based and CNN-based models but has an average improvement on the GRU performance. To evaluate the effectiveness of the proposed model, we visualize the forecasting results and provide statistical characteristics of the data to enable governments to make long-term decisions on how to deal with the pandemic.

The remainder of this paper is organized as follows. Section 2 provides a brief review on COVID-19 time series forecasting and the description of the employed deep learning methods. In Sect. 3, we present the proposed approach and the architectures of the designed models. Section 4 assesses the usefulness of the proposed method via the experimental study. Discussions are provided in Sect. 5, and finally, the paper concludes in Sect. 6 with some suggestions for future work in this area.

2 Related work

This section first presents a review of the COVID-19 time series forecasting methods and then describes the utilized models throughout the study.

2.1 COVID-19 Forecasting

Various approaches, mostly mathematical, statistical, machine learning, and deep learning models have been utilized in previous studies [3, 4, 14, 16, 17, 26]. Rahimi et al. [27] provided a review of widely used forecasting models on COVID-19 data. Here, we concentrate mainly on COVID-19 time series forecasting studies and present a brief review in this context.

Al-Qaness et al. [28] presented an improved adaptive neuro-fuzzy inference method (ANFIS) that uses an enhanced flower pollination algorithm (FPA) by the salp swarm algorithm (SSA) to forecast the COVID-19 cases in China. Their model is more potent in terms of mean absolute percentage error (MAPE), root mean squared relative error (RMSRE), coefficient of determination, and computing time. Torrealba-Rodriguez et al. [3] used Gompertz, logistic, and artificial neural network (ANN) models. Their results from the infected cases in Mexico showed a high coefficient of determination between the studied data and those obtained by the proposed models. Similar studies which considered Gompertz and logistic models can be found in [3, 29–32]. Castillo and Melin [26] studied an approach based on fuzzy fractal for data from 10 countries by combining (1) fractal dimension to evaluate the complexity of the dynamics in the time series and (2) fuzzy logic to reflect the uncertainty forecasting. Melin et al. [33] introduced a multiple ensemble neural network model with fuzzy logic response aggregation. Their experiments on the data of Mexico infected cases show the superiority of their proposed model over the single ANN.

Kırbaş et al. [15] used autoregressive integrated moving average (ARIMA), nonlinear autoregression neural network (NARNN), and LSTM approaches to study the data of 8 European countries. Shahid et al. [16] proposed forecast models with ARIMA, support vector regression (SVR), LSTM, Bi-LSTM in 10 significantly affected countries. Leila et al. [34] applied ANN and ARIMA models, and Petropoulos and Makridakis [35] implemented exponential smoothing forecasting to predict the infected cases.

Arura et al. [14] utilized recurrent neural network (RNN)-based variants such as deep LSTM, convolutional LSTM, and Bi-LSTM for the cases in India. For Russia, Peru, and Iran, Wang et al. [16] used LSTM networks and rolling updating mechanisms to feed new forecasting outcomes into model training for the next iteration. The study of Hasan [36] suggested a hybrid model consisting of ensemble empirical mode decomposition (EEMD) and artificial neural network (ANN), which outperformed conventional statistical analysis.

Machine learning algorithms were used by Li et al. [37] in predicting mortality in confirmed cases of COVID-19. Their results indicated that the gradient boosting decision tree (GBDT) outperforms logistic regression (LR) models,

the performance comparison appeared to be independent of disease severity, and the 5-index LR or LR-5 model is powerful in death prediction with a high area under the curve (AUC).

Reviewing the previous studies indicate that computational intelligence methods and especially deep learning methods have attracted growing attention in COVID-19 time series forecasting. Even though deep neural networks have performed reasonably well when applied on COVID-19 time series data, in this study, we aim to enhance their predictive power by feeding them with more data. In general, the performance of the generated model in a deep learning task is largely determined by the amount of samples used in the model training phase. The inherent problem in time series forecasting is that time series are often short, and accordingly, the number of extracted samples becomes small. To address this problem, we propose to generate a new time series with similar characteristics to the original time series using statistical data augmentation methods. The obtained series via the augmentation approach is used to create new samples. In this way, a sufficient number of instances are provided for model learning.

2.2 Description of the employed models

We use RNN for the sequence processing task, which can catch the temporal dependencies in a time sequence, unlike ANN. However, the key issue with RNN is the gradient vanishing/exploding problem, which makes them difficult to train. Two new architectures with gating mechanisms, the LSTM [38] and GRU [39], have been proposed to solve this problem. In addition, we will use CNN, which is briefly discussed here, as another deep learning unit in our experiments.
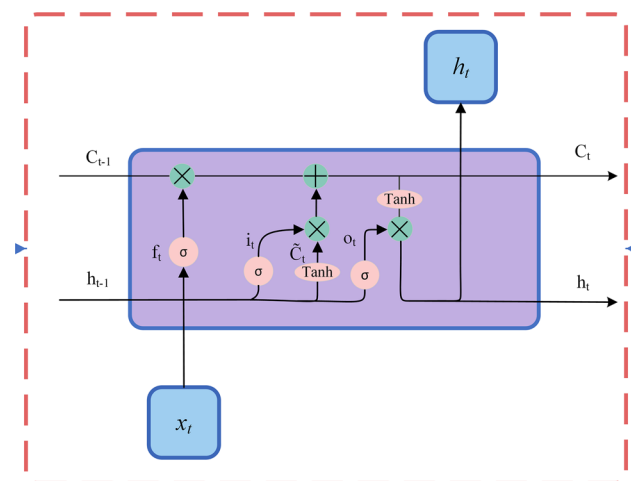

Fig. 1 Structure of an LSTM unit

2.2.1 LSTM

In this section, we explain the structure and mechanism of the LSTM unit. As illustrated in Fig. 1, each LSTM unit is comprised of a memory cell C, an input gate i, an output gate o and a forget gate f.

Considering the following parameters, the learning procedure of LSTM is described below:

– x_t :
  the input vector at time step t
– $b = \{b_i, b_o, b_f, b_c\}$
  are bias vectors of input, output, forget, and memory cell.
– $W = \{W_i, W_o, W_f, W_c\}$
  are weight matrix of input, output, forget, and memory cell.
– $U = \{U_i, U_o, U_f, U_c\}$
  are the recurrent weights of input, output, forget, and memory cell.

The output h_t of the LSTM unit is computed as follows:

$$h_t = o_t \tan h(c_t), \tag{1}$$

where o_t is the output gate that regulates the outgoing information of the LSTM unit and c_t is the memory. o_t is computed by

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \tag{2}$$

where σ is the logistic sigmoid and h_{t-1} is the output vector (hidden state) of the time $t-1$. The memory cell is updated as follows:

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t, \tag{3}$$

where $\tilde{c}_t$, the newly computed memory is obtained as follows:

$$\tilde{c_t} = \tan h(W_c x_t + U_c h_{t-1} + b_c). \tag{4}$$

In fact, the memory cell c_t is a combination of the previous memory multiplied by the forget gate, f_t and the new memory $\tilde{c}$ regulated by the input gate, i_t. f_t and are computed as follows:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f), \tag{5}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i). \tag{6}$$

2.2.2 GRU

GRU is another variant of RNN that uses gating mechanism to regulate the flow of information inside the unit. Unlike LSTM, GRU does not contain a memory cell. As Figure 2 portrays, the GRU has two gates, a reset gate r_t
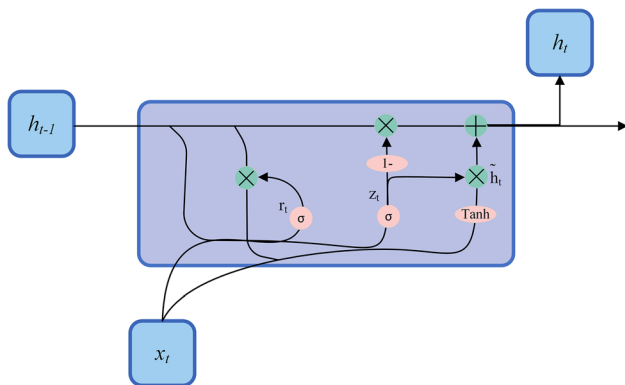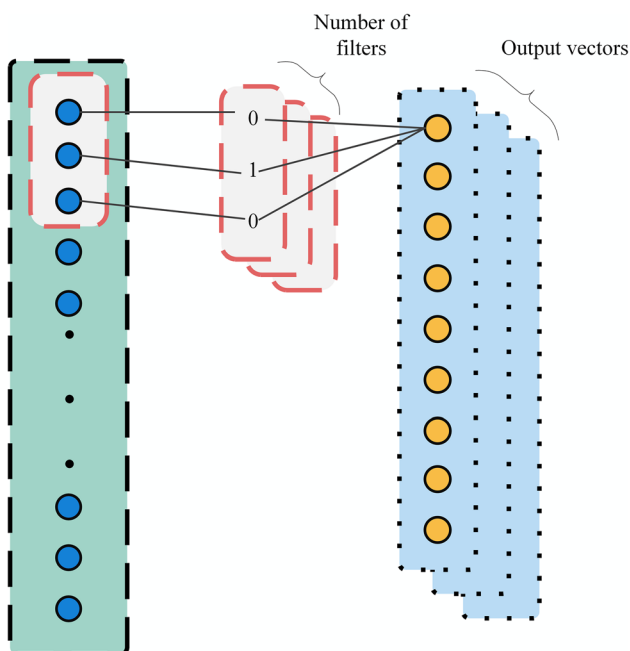
Fig. 2 Structure of a GRU unit [39]

Fig. 3 Structure of CNN for time series

and an update gate z_t. The rest gate decides how to combine the new input x_t with the previous hidden state, h_{t-1}. Also, the update gate determines how much unit updates its hidden state.

2.2.3 CNN

CNN has shown promise in a variety of fields, including machine vision [23]. CNN's convolutional layers take input data and extract new features by performing convolution operations on it with convolution kernels. Each CNN contains a convolution kernel (i.e., a small window) that

slides over the input data and performs convolutional operations to generate new features, as shown in Fig. 3 [40]. The generated features obtained by the convolution technique are typically more discriminative than the raw input data, resulting in better forecasting.

3 Proposed method

Deep learning methods such as LSTM, CNN, and GRU have been applied successfully in the time series forecasting context. These techniques' performance mainly depends on having enough data to fit their parameters suitably [12]. The number of samples extracted from a short time series may be insufficient to achieve an optimal model [19]. These methods should be appropriately regularized to prevent them from overfitting. Another difficulty with time series forecasting is that, even if the series is long and adequate data are available, the observations from the far past usually provide fewer determinants for predicting. In other words, recent observations of an individual series are more useful in forecasting. This may be because of the changes that happen in patterns that existed in a series.

The commonly used procedure of data preparation for a time series forecasting task is illustrated in Fig. 4. As shown, a given time series is divided into in-samples and out-samples considering a certain ratio, for example, 80/20. The out-sample part (test data) $\{t_{m+1}, t_{m+2}, ..., t_n\}$ is used to evaluate the obtained model. Also, the in-sample part is divided into the train data $\{t_1, t_2, ..., t_k\}$ and the validation data $\{t_{k+1}, t_{k+2}, ..., t_m\}$. The validation data are utilized to tune the model's hyperparameters and to evaluate a model fit on the train data. Selecting separate validation data leads to excluding the recent observations from the train data, so the recent patterns that exist in the data will not be captured. One simple solution to tackle this problem is to include the validation data in model training. However, in this way, overfitting may occur, which usually leads to loss of accuracy on test data. In this study, we propose to use time series augmentation methods to avoid model overfitting and improve the accuracy.

Specifically, we utilize a time series augmentation technique to create new series with the same temporal dependencies that exist in the original series. The augmented time series is used to create a new validation set. The overall procedure of the proposed idea is illustrated in Fig. 5. The proposed model contains preprocessing and model training phases. Firstly, in the preprocessing phase, a
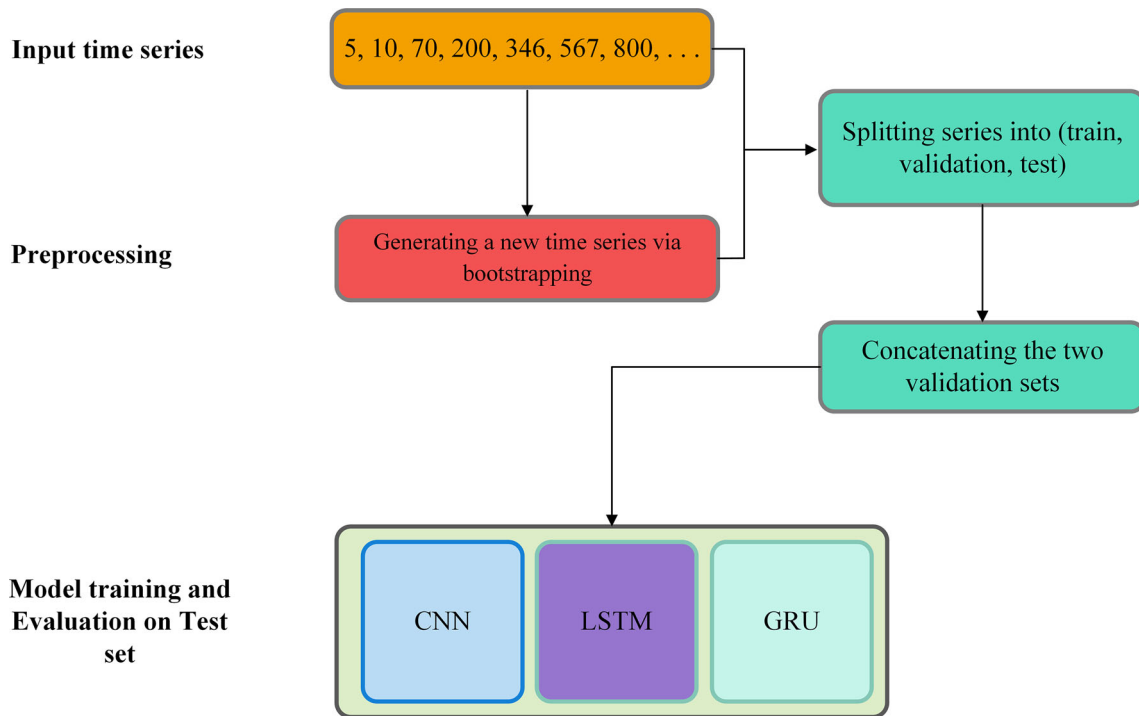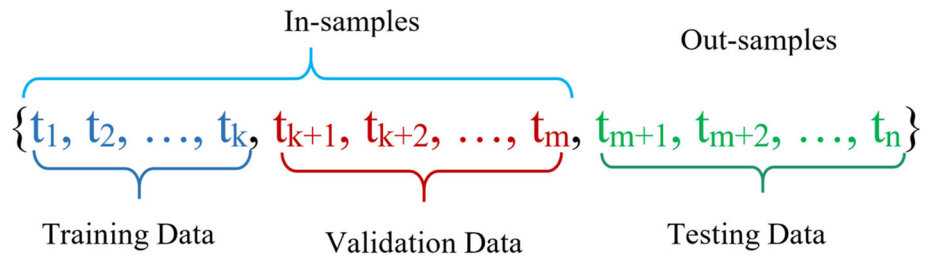
Fig. 4 An example of time series

Fig. 5 Proposed schema

time series augmentation technique is applied, and then, the sample generation procedure is accomplished. In the modeling phase, the deep learning models are employed on the generated samples, and the best model is achieved. In the model training process, we adopt the Bayesian optimization algorithm to fine-tune the hyperparameters of each model.

To explain our proposal, we describe its procedure using Algorithm 1. To augment a time series, we apply the method proposed in [20]. This algorithm firstly applies the Box–Cox transformation to the series and then decomposes the series into trend, seasonal, and reminder adopting STL or Loess [41]. Then it bootstraps the reminder using the moving block bootstrap (MBB) [42], and the trend and seasonal components are added together, and finally, inverse Box–Cox transformation is applied. As illustrated in Algorithm 1, lines 1-7 show the procedure of computing bootstrapped series. In lines 8-10, the bootstrapped series are aggregated, and then, for the original series and the augmented series, the instances with input–output format are created considering a Lag, and an output window (Output_Window). Line 11 concatenates the two validation sets. In lines 12-18, the benchmarking models are trained and evaluated, and the best model in terms of RMSE is returned.

Algorithm 1: The proposed schema

input : Time series (ts), num_Boot, Lag, Output_Window, Model_list
output: Forecasting model (M)

1   ts_bc = **Box-Cox_transform**(ts)
2   (trend, seasonal, remainder) = **Decompose**(ts_bc)
3   **For** i in 1 to num_Boot:
4   Boot[i] = **MBB**(reminder)
5   Series_BC[i] = trend + seasonal + Boot[i]
6   Series[i] = **Invert_ Box-Cox_transform**(Series_BC[i])
7   **End**
8   ts_aug = **Mean**(Series)
9   (train, val, test) = **GenerateSamples**(ts, Lag, Output_Window)
10   (train_aug,val_aug, test_aug) = **GenerateSamples**(ts_aug, Lag, Output_Window)
11   Val_new = **Concat**(val, val_aug)
12   **For** i in 1 to **Size** (Model_list):
13   Models [i] = **Train_model_BO**(Model_list[i], train,Val_new)
14   Forecast[i]=**Forecast**(Models[i])
15   RMSE[i]=**RMSE**(Forecast[i])
16   **End**
17   M = Select the best model from **Models** with lowest **RMSE**
18   **Return** M

3.1 Architecture of the utilized deep learning models

Three state-of-the-art deep learning models are employed to explore whether the forecasting performance of the proposed scheme is better than the performance of the regular approach. The list of benchmarking models along with their architectures is provided in Table 1.

Also, Fig. 6 illustrates the full architectures of the proposed methods. As can be observed from the figure, the dense and output layers are the same for LSTM, CNN, and GRU. Every model learns a representation (a feature vector) of the input data and feeds it into the fully connected

Table 1 Architecture of three deep learning models used to evaluate the proposal

Benchmarking models	Architecture
LSTM	LSTM layer
	Dense
	Output
GRU	GRU layer
	Dense
	Output
CNN	1D convolution layer
	Dense
	Output

(dense) layer; afterward, the predictions are computed using the output layer.

3.2 Hyperparameter selection procedure using bayesian optimization

The choice of optimal hyperparameters is essential in obtaining a forecasting model with high accuracy [43]. Deep learning-based models usually contain several hyperparameters. Although grid search is a popular strategy for finding the optimal hyperparameters, it requires more computational time and resources to fine-tune deep learning methods. This is due to the fact that the grid search method exhaustively considers all parameter combinations, so it needs more computational resources, especially in the case of deep learning. The main reason behind using the Bayesian hyperparameter optimization is that it does not consider all hyperparameter combinations, and so less training time and resources are needed. The Bayesian hyperparameter optimization uses Bayesian models based on Gaussian processes to predict good tuning parameters [43]. The study of Wu et al. [43] indicated that the Bayesian optimization-based method could find the optimal hyperparameters for the popular machine learning algorithms. In line with [11, 12, 43], the Bayesian optimization technique [44, 45] is used to tune the hyperparameters in all of the experiments in this study. The Bayesian optimization algorithm uses the error on the validation data to determine the appropriateness of each model.
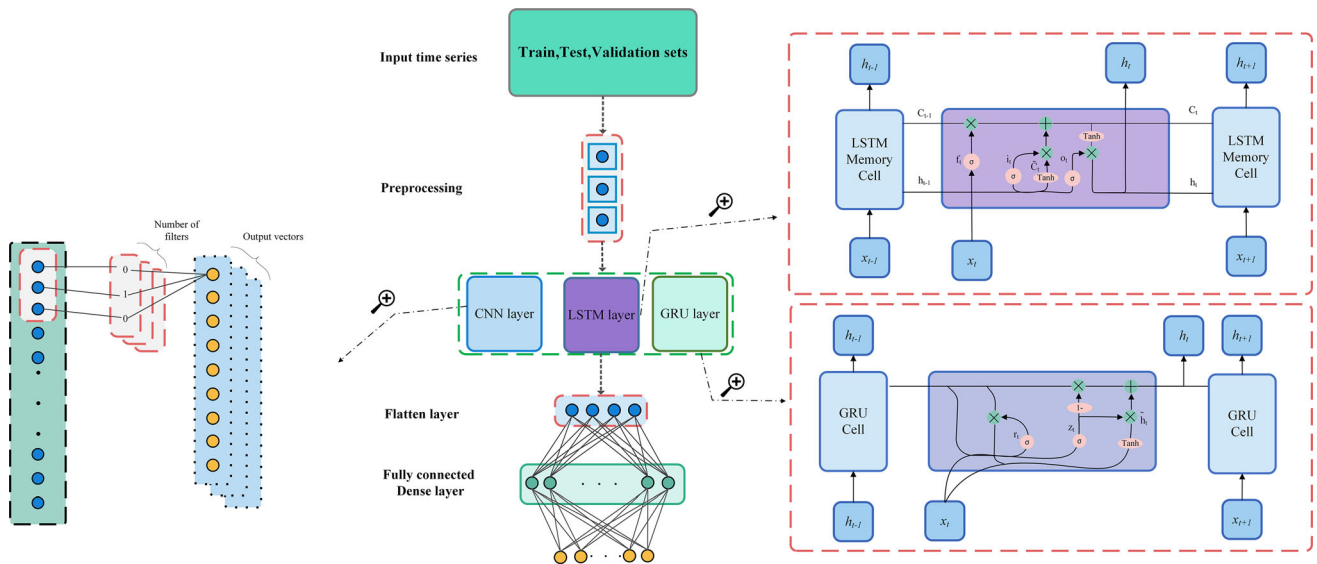
Fig. 6 Architecture of the utilized models

4 Experimental study

In this study, we use R forecast package[1] version 8.14 to generate the augmentation of each time series. Also, the deep learning models are implemented with Keras [46], the Python deep learning library.

4.1 Dataset

The Humanitarian Data Exchange (HDX) [47] is the source of the data utilized in this study. The Corona Virus Resource Center at Johns Hopkins University has compiled and released a credible source of COVID-19 reported cases on HDX so that scientists can model the disease's spread and conduct data analysis [48]. The dataset contains the daily record of confirmed cases in the time series format, including temporal patterns. In this study, the experiments were conducted using the time series data for ten countries with the highest number of confirmed cases from January 20, 2020, to March 28, 2021. These countries are the USA, Brazil, India, France, Russia, the UK, Italy, Spain, Turkey, and Germany. The last 28 days of each series are used as the test set, and the remaining days are used as the training data. We also made the validation set the same size as the test set (28 days).

4.2 Statistical properties of the data

In Tables 2 and 3, we apply statistical properties to the aforementioned ten countries with the highest COVID-19 cases to better interpret the dataset.

The sample size refers to the number of observations included in the experiment for each country which is not necessarily equal because it is counted from the day the first COVID-19 cases were reported.

The mean or average of the data is the most popular and well-known measure of central tendency and is equal to the sum of all the values in the dataset divided by the number of observations. It is worth noting that the total cases can be obtained by multiplying the sample size by the sample mean during the study period. Other than mean, two other measures of central tendency are median and mode. Median is the middle value for the dataset that has been arranged in order of magnitude. An essential property about the median is that it is less affected or "Robust" by outliers and skewed data. Mode, on the other hand, is the most frequent number of daily cases in our dataset. It does not give a fair measure of central tendency when compared to median and mean [50]. The obtained mode for most of the countries is zero. The reason for that could be having days without any new cases or failing to report instances due to holidays or weekends.

The sample's square root of variance often known as standard deviation is a measure of the amount of variation or dispersion of the dataset, using the same unit as the mean. A small standard deviation implies that the values tend to be close to the mean of the dataset, whereas a high standard deviation suggests that the values are spread out over a wider range [51]. Besides standard deviation, two other dispersion measures are (1) skewness, where it measures the amount of asymmetricity, and (2) kurtosis, where it determines the heaviness of the distribution tails, also known as is the "tailedness" or the "peakedness." For a country dataset with one mode (uni-modal), a positive

Table 2 Statistical properties of the daily data of the COVID-19 cases for the USA, Brazil, India, France, and the UK

Country	USA	Brazil	India	France	UK
Sample size	432	397	424	430	423
Mean	70,052	31,574	28,395	11,088	10,277
Median	47,043	28,629	18,537	4321.5	4,329
Mode	0	0	0	0	0
Standard deviation	68,074.5	23,178.8	27,378.6	14,657.6	13,657.4
Skewness	1.31	0.45	0.83	2.24	1.88
Standard error of skewness	0.12	0.12	0.12	0.12	0.12
$Z_{Skewness}$	11.15	3.68	6.98	19.01	15.82
Kurtosis	0.76	−0.53	−0.42	7.43	3.37
Standard error of kurtosis	0.23	0.24	0.24	0.24	0.24
$Z_{Kurtosis}$	3.26	−2.16	−1.78	31.60	14.21
Min	0	0	0	0	0
Max	300,416	100,158	97,894	106,091	68,192
Range	300,416	100,158	97,894	106,091	68,192

Table 3 Statistical properties of the daily data of the COVID-19 cases for Russia, Italy, Spain, Turkey, and Germany

Country	Russia	Italy	Spain	Turkey	Germany
Sample size	423	423	422	383	427
Mean	10,566	8,351	7,965	8,376	6,521
Median	8,764	2,843	1,931	2,026	1,898
Mode	0	0	0	987	0
Standard deviation	8,348.8	9,959.9	12,779.9	42,585.6	8,722.4
Skewness	0.68	1.16	2.85	18.45	1.76
Standard error of skewness	0.12	0.12	0.12	0.13	0.12
$Z_{Skewness}$	5.75	9.77	23.97	147.56	14.88
Kurtosis	−0.57	0.41	11.14	353.45	2.90
Standard error of kurtosis	0.24	0.24	0.24	0.25	0.24
$Z_{Kurtosis}$	−2.41	1.74	47.00	1,419.46	12.29
Min	0	0	0	0	0
Max	29,499	40,902	93,822	823,225	49,044
Range	29,499	40,902	93,822	823,225	49,044

skewness shows that the data are asymmetric and skewed to the right, a negative skewness explains that the data are asymmetric and skewed to the left, and finally, a symmetric dataset always has a zero skewness. To provide a comparison to the standard normal distribution, it is common to use an adjusted version known as the excess kurtosis, which is the kurtosis minus 3. A dataset with zero excess kurtosis is called "Mesokurtic," with a positive excess kurtosis is named "Leptokurtic" indicating heavy tails with large outliers and less variable, and with a negative excess kurtosis is known as "Platykurtic" which have the flattest peak and highly dispersed [52].

A Z-score for skewness and kurtosis can be obtained by dividing the skew values or excess kurtosis by their standard errors, respectively, which are shown as $Z_{Skewness}$ and $Z_{Kurtosis}$ in Tables 2 and 3. As the studied sample size of the countries is large, either an absolute skew value larger than

2 or an absolute kurtosis larger than 7 can be used as reference values for determining the significance of non-normality [53]. It is worth mentioning that the utilized models in this study are based on neural networks and deep learning. These methods are nonparametric that model the data without prior assumptions of their distribution [54].

Finally, the range for each country is the difference between the dataset's largest and smallest observations, which expresses a country dataset's dispersion.

4.3 Measures of evaluation

The two forecasting performance measures used in the comparison are (1) symmetric mean absolute percentage error (SMAPE) which is defined as

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|y_t - f_t|}{\frac{(|y_t| + |f_t|)}{2}} \times 100, \qquad (7)$$

and (2) the root mean square error (RMSE) which is obtained by

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - f_t)^2} \qquad (8)$$

where f_t and y_t are the predicted and observed values at time point t, respectively.

4.4 Hyperparameter selection

Table 4 illustrates the domain of all hyperparameters utilized in the implemented models. The lag hyperparameter exploited in transforming input time series into samples suitable for deep learning techniques has a significant impact on obtaining models that can forecast future values with minimum error [49]. Another important hyperparameter is the learning rate that regulates how the weights are adjusted during the model training. Additionally, the utilized models throughout this study have different key hyperparameters that influence the forecasting accuracy of obtained models. The hyperparameters specific to each model are provided in Table 4. Also, as outlined in the previous section (see Table 1), each utilized deep learning techniques contain a dense layer that follows the sequence capturing layer (e.g., LSTM, CNN, or GRU) and an output layer, which produces the outputs. These layers are common in all the utilized models, and their ranges are also given in Table 4.

4.5 Data preprocessing

According to the methodology shown in Fig. 5 and the procedure described in Algorithm 1, firstly, a new time series is generated via augmentation. Next, the original and the augmented series are transformed into samples with the input–output format. Then, the resulted samples are split into train set, validation set, and test set following the holdout procedure. Finally, a new validation set is created by concatenating the validation samples corresponding to the original series and the augmented one. It should be noted that this study adopts multi-output forecasting, and the sample generation process is performed using the lag (size of the input window) and the output window. In all experiments, the output window is set to 7 days. Following the multi-output forecasting strategy, for time series T: $t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, \ldots, t_n$ and Lag $= 5$, Output_window $= 2$, the created instances are shown in Table 5.

4.6 Evaluation of the effectiveness of the proposed approach

In this section, we investigate whether our proposed approach is able to enhance the forecasting accuracy of the deep learning models based on LSTM, CNN, and GRU. We run our experiments on the data of the before-mentioned ten countries. All experiments are repeated ten times, and the average performance measures are reported.

Table 5 An example of the sample generation function

Input	Output
t_1, t_2, t_3, t_4, t_5	t_6, t_7
t_2, t_3, t_4, t_5, t_6	t_7, t_8
t_3, t_4, t_5, t_6, t_7	t_8, t_9
.	.
$t_{n-(L+O)+1}, t_{n-(L+O)+2}, t_{n-(L+O)+3}, t_{n-(L+O)+4}, t_{n-(L+O)+5}$	t_{n-1}, t_n

Table 4 Range of the hyperparameters

Hyperparameter	Range
Common hyperparameters	Lag: [10, 11, 12, 13, 14, 15]
	Learning rate: [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05]
	Dense activation function: [ReLU, Linear]
	Output activation function: [ReLU, Linear]
LSTM	Activation function: [ReLU, Linear]
	Dropout rate: [0.0, 0.1, 0.2, 0.3, 0.4, 0.5]
	Number of units: [4, 8, 16, 32, 64, 128]
CNN_FE	Kernel size: [2, 3, 4]
	Number of filters: [32, 64, 128, 256]
GRU	Activation function: [ReLU, Linear]
	Number of units: [4, 8, 16, 32, 64, 128]
	Dropout rate: [0.0, 0.1, 0.2, 0.3, 0.4, 0.5]

Table 6 LSTM results for ten countries in terms of SMAPE and RMSE for regular and augmentation approaches—LSTM_Aug is obtained following the proposed approach

Country	SMAPE		RMSE	
	LSTM_Aug	LSTM	LSTM_Aug	LSTM
USA	**0.87**	1.73	**291935.32**	570575.55
Brazil	**0.62**	0.76	**83105.03**	101239.26
India	**0.50**	0.77	**81741.89**	112064.7
France	**0.60**	0.66	**40719.57**	45009.65
Russia	**0.86**	1.23	**41813.95**	59394.68
UK	**0.46**	2.32	**22279.42**	106723.4
Italy	0.62	**0.44**	22976.17	**18834.81**
Spain	**1.29**	2.37	**51592.85**	86450.37
Turkey	**1.71**	2.13	**66494.82**	78740.49
Germany	0.73	**0.65**	24476.58	**21542.29**
Mean	**0.82**	1.31	**72713.56**	120057.52

Bold values indicate the best results

Table 7 Results of CNN for ten countries in terms of SMAPE and RMSE for regular and augmentation approaches. CNN_Aug is obtained following the proposed approach

Country	SMAPE		RMSE	
	CNN_Aug	CNN	CNN_Aug	CNN
USA	**0.24**	0.32	**81157.55**	104899.61
Brazil	**0.54**	0.67	**73194.5**	88552.5
India	0.66	**0.52**	102096.63	**81892.72**
France	**0.49**	0.51	**30442.08**	33311.58
Russia	**0.32**	0.48	**15545.49**	23084.47
UK	**0.89**	0.93	**39566.78**	41051.61
Italy	**0.35**	0.50	**14129.16**	19769.05
Spain	**1.16**	1.41	**47599.51**	54012.22
Turkey	**1.04**	1.22	**39365.34**	45970.07
Germany	**0.62**	0.73	**22090.1**	26058.33
Mean	**0.63**	0.73	**46518.71**	51860.22

Bold values indicate the best results

4.6.1 LSTM

Table 6 shows the results obtained using the deep learning model based on LSTM. As it can be seen, the model obtained using the proposed approach (LSTM_Aug) leads to a lower error in terms of SMAPE and RMSE for eight countries. LSTM_Aug achieves superior results for the USA, Brazil, India, France, Russia, the UK, Spain, and Turkey. Also, the mean SMAPE for LSTM_Aug is 0.82, which is lower than the one for LSTM, which has a mean of 1.30. Also, regarding RMSE, the mean RMSE measure for LSTM_Aug is significantly lower than the mean RMSE of LSTM. The experiments indicate that the results of LSTM_Aug are excellent, and the proposed approach significantly impacts the performance of LSTM.

4.6.2 Convolution model

The results of experiments using deep learning model based on CNN are given in Table 7. The best values are shown in boldface. In terms of SMAPE, the CNN_Aug model achieves better performance in 9 countries out of 10. Also, in terms of RMSE, we see a similar performance where CNN_Aug beats CNN in 9 cases. To give a comprehensive report on the performance of models, the mean SMAPE and mean RMSE measures also are computed. The mean SMAPE for CNN_Aug is 0.63, which is lower than that for CNN (0.73). Also, this is true for mean RMSE in which the CNN_Aug achieves lower error than CNN. The results indicate that CNN_Aug outperforms CNN, and that using the proposed data preparation strategy considerably enhances the accuracy of CNN-based deep learning models.

Table 8 GRU results for ten countries in terms of SMAPE and RMSE for regular and augmentation approaches—GRU_Aug is obtained following the proposed approach

Country	SMAPE		RMSE	
	GRU_Aug	GRU	GRU_Aug	GRU
USA	**0.28**	0.44	**97535.03**	152119.5
Brazil	**0.75**	0.77	**99050.16**	101927.57
India	**0.53**	0.72	**83931.57**	109283.11
France	0.60	**0.55**	40609.86	**36529.07**
Russia	**0.54**	0.93	**26955.22**	47059.80
UK	**0.51**	0.52	**24700.29**	24989.78
Italy	0.43	**0.36**	18585.69	**15907.25**
Spain	4.1	**1.25**	145384.97	**50215.92**
Turkey	1.64	**1.48**	64087.62	**57313.48**
Germany	0.76	**0.66**	25800.14	**21647.34**
Mean	1.01	**0.77**	62664.06	**61699.28**

Bold values indicate the best results

4.6.3 GRU model

Table 8 provides the results of experiments using the deep learning method based on GRU. Similar to the previously mentioned models, we compare the model obtained using the regular experimental setting (GRU model) with the model obtained using the proposed augmentation approach (GRU_aug). GRU_Aug and GRU perform similarly as each of them achieves minimum error in terms of SMAPE and RMSE in 5 cases out of 10 countries. This can be attributed to the fact that the GRU model uses different

gating units and use less training parameters and this prevents it from overfitting.

4.7 Overall comparison of the proposed method with regular approach

To provide an overall description of the results, we summarize the results of experiments and show the top augmentation model for each country in Table 9. As can be seen from the table, for all ten countries, the models based on the proposed augmentation approach show the top accuracy in terms of both SMAPE and RMSE. This demonstrates the effectiveness of the proposed approach in increasing the forecasting accuracy of the deep learning methods. Also, CNN Aug performs excellently and reaches the best model for eight countries including, the USA, Brazil, France, Russia, Italy, Spain, Turkey, and Germany. Besides, LSTM_Aug achieves the best accuracy for two countries. Furthermore, as illustrated in Table 9, in no country is GRU_Aug superior.

4.8 Visualizing the results

To further demonstrate the forecasting ability of the obtained models, in this section, the actual and forecasts for each country are visualized in Figs. 7, 8, 9, 10, 11, 12, 13, 14, 15, and 16. The actual values are shown in red in all figures, while the forecasts are shown in black using the best deep learning model (Figs. 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16). As Figs. 7, 8, 9, 10, 11, 12, 13, 14, 15, and 16 indicate, the predicted cases for the USA, Brazil, France, Russia, UK, Italy, Turkey, and Germany are very close to the actual values; and there is a minimum error. Also, there are overlaps at some of the time points that demonstrate the power of the proposed approach. The plot for India (Figure 9) indicates that from time point 1 to time point 15, the forecasted values are very close to the real values but after time point 15, the error increases. Furthermore, as shown in Fig. 14, the inaccuracy is rather substantial in various time

Table 9 Top model for each country

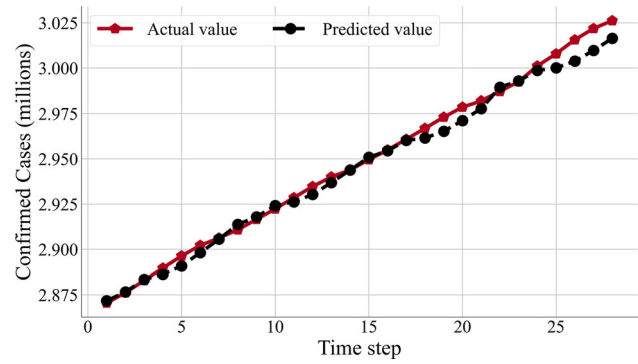| Country	Top model
USA	CNN_Aug
Brazil	CNN_Aug
India	LSTM_Aug
France	CNN_Aug
Russia	CNN_Aug
UK	LSTM_Aug
Italy	CNN_Aug
Spain	CNN_Aug
Turkey	CNN_Aug
Germany	CNN_Aug

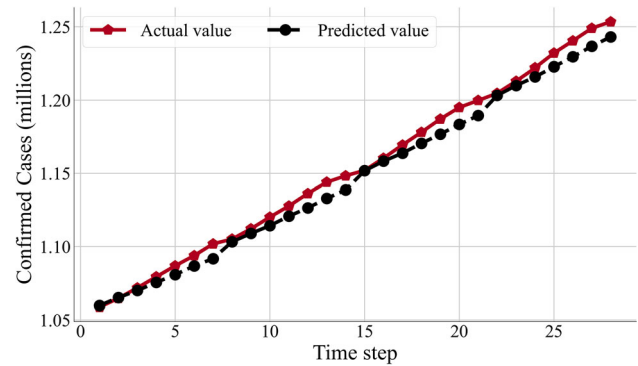**Fig. 7** Actual and forecasted number of cases for test set—USA

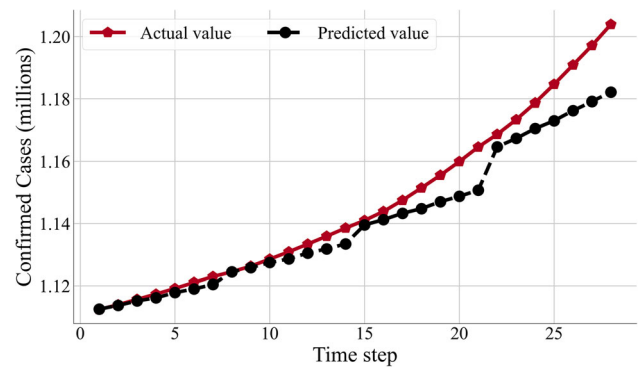**Fig. 8** Actual and forecasted number of cases for test set—Brazil

Fig. 9 Actual and forecasted number of cases for test set—India

points for Spain. This is primarily due to the noise in the country's input data.

5 Discussion

In this study, we proposed a method that uses augmentation techniques to enhance time series forecasting. To conduct experimental study and to test the effectiveness of the proposed idea, we selected three deep learning methods, LSTM, GRU, and CNN. Furthermore, due to the importance of accurate forecasting of COVID-19 infections, data
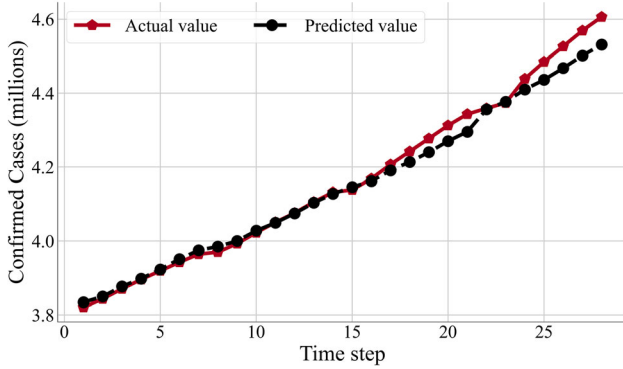
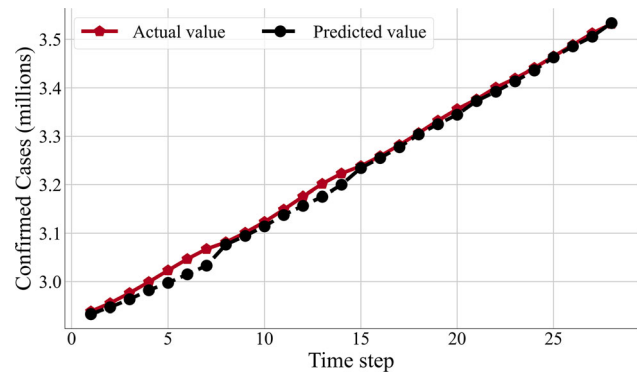**Fig. 10** Actual and forecasted number of cases for test set—France

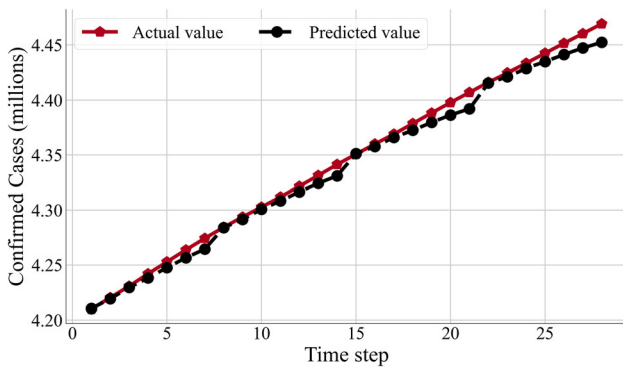**Fig. 13** Actual and forecasted number of cases for test set—Italy

Fig. 11 Actual and forecasted number of cases for test set—Russia
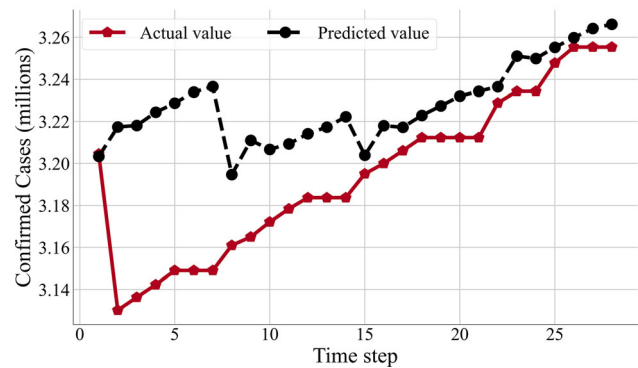

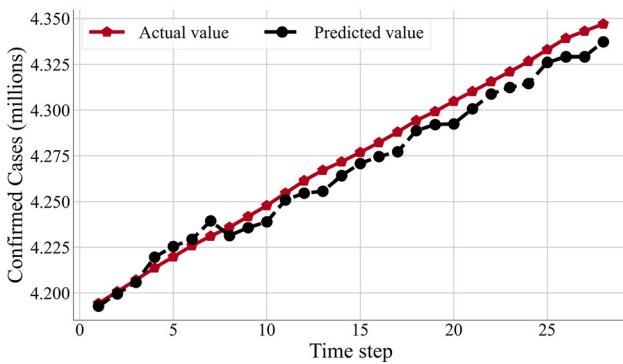**Fig. 14** Actual and forecasted number of cases for test set—Spain

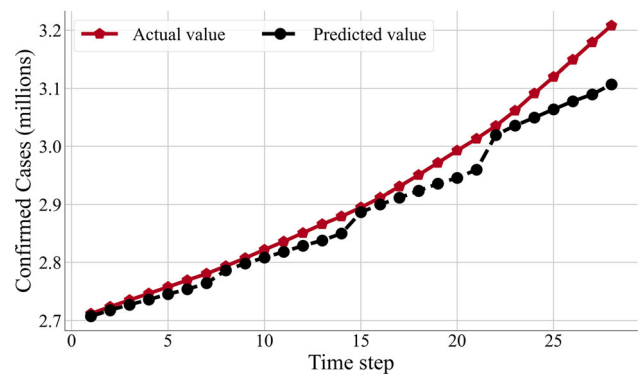**Fig. 12** Actual and forecasted number of cases for test set—UK

Fig. 15 Actual and forecasted number of cases for test set—Turkey

of ten countries with highest cases of infections have been chosen. The results of experiments demonstrated that the models obtained employing LSTM and the proposed idea greatly outperforms the regular LSTM model. Similarly, the proposed method significantly improves the performance of the CNN models. Besides, for GRU, the proposed method achieves an average performance.

5.1 Assumptions

Similar to any time series forecasting task, in this study, we utilize the series past values to train the models. Also, we assume that an optimal hyperparameters for the utilized models have been chosen.

5.2 Implications of the results

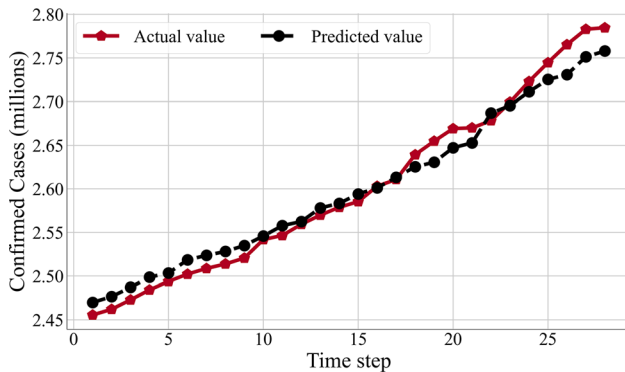Unlike the one-step-ahead forecasting, where a forecasting model uses the previous observations to predict a single

Fig. 16 Actual and forecasted number of cases for test set—Germany

time step, the multi-step-ahead forecasting strategy [24], which was used in this study, allows forecasting two or more steps. In the COVID-19 forecasting, the multi-step-ahead forecasting is attractive to policymakers. In fact, a longer window forecasting uncovers the trend of pandemic effectively and thus appeal more significant for governments. Also, in terms of SMAPE, the models generated following the proposed idea demonstrate excellent performance. Besides, the Mean SMAPE values for LSTM_Aug, CNN_Aug, and GRU_Aug are 0.82, 0.63, and 1.01, respectively indicating the forecasting power of the proposed method.

5.3 Practical implications

As we mentioned previously, in this study, we formulate forecasting the number of infected cases as a time series forecasting problem in which the data of past observations of a series is used for predicting the future time points. The proposed models forecast the number of infected cases for a longer horizon with minimum error in comparison to their regular counterparts. The forecasts can be utilized by governments to take appropriate decisions in controlling the pandemic.

5.4 Limitations

In this study, we did not access to the other sources of information such as the interventions implemented by each country or vaccination of COVID-19. The models only were learned using the time series of the infections. Another limitation of this study is related to the hyperparameter selection for deep learning methods. As these methods contains a complex architecture, they require more computation. Therefore, investigating every hyperparameter configuration, similar to way performed in the grid search method, may not practicable. Therefore, in this study we used the Bayesian optimization algorithm to search the optimal hyperparameters.

6 Conclusion and future work

A new schema based on time series augmentation was suggested in this study to improve the performance of deep learning techniques in time series forecasting. The proposed method's main idea is to use a time series augmentation technique to create a new time series with the same properties in the original series. Then, we use the generated series to obtain enough samples to train the deep learning methods optimally. The proposed method is implemented in the context of COVID-19 time series forecasting data of the 10 most affected countries using the LSTM, GRU, and CNN models. According to the findings of the experiments, in the majority of countries, the LSTM_Aug model outperformed the standard LSTM model and the CNN_Aug model achieved significant performance than the regular CNN. In addition, GRU_Aug obtained an average performance when compared to the regular GRU. Overall, the models' performance following the proposed idea is excellent and significantly improves the regular models. As future work, we intend to evaluate the proposed method using other time series augmentation approaches such as dynamic time warping barycentric averaging.

Author Contributions HA performed conceptualization, methodology design, software development, validation, writing the original draft, and writing, reviewing, and editing. RP had contributed to software development, data curation, and visualization. AB took part in writing, reviewing, and editing.

Declarations

References

1. Gorbalenya AE, Baker SC, Baric RS, de Groot RJ, Drosten C, Gulyaeva AA, Haagmans BL, Lauber C, Leontovich AM, Neuman BW, Penzar D, Perlman S, Poon LLM, Samborskiy DV, Sidorov IA, Sola I, Ziebuhr J, Coronaviridae Study Group of the International Committee on Taxonomy of V (2020) The species Severe acute respiratory syndrome-related coronavirus: classifying 2019-nCoV and naming it SARS-CoV-2. Nature Microbiology 5(4):536–544. https://doi.org/10.1038/s41564-020-0695-z
2. Nikolopoulos K, Punia S, Schäfers A, Tsinopoulos C, Vasilakis C (2021) Forecasting and planning during a pandemic: COVID-19 growth rates, supply chain disruptions, and governmental decisions. Eur J Oper Res 290(1):99–115. https://doi.org/10.1016/j.ejor.2020.08.001

3. Torrealba-Rodriguez O, Conde-Gutiérrez R, Hernández-Javier A (2020) Modeling and prediction of COVID-19 in Mexico applying mathematical and computational models. Chaos, Solitons & Fractals:109946

4. Abbasimehr H, Paki R (2021) Prediction of COVID-19 confirmed cases combining deep learning methods and Bayesian optimization. Chaos, Solitons & Fractals 142:110511. https://doi.org/10.1016/j.chaos.2020.110511

5. Abbasimehr H, Paki R, Bahrini A (2021) Improving the performance of deep learning models using statistical features: The case study of COVID-19 forecasting. Mathematical Methods in the Applied Sciences. https://doi.org/10.1002/mma.7500

6. Sultana J, Usha Rani M, Farquad MAH An Extensive Survey on Some Deep-Learning Applications. In: Venkata Krishna P, Obaidat MS (eds) Emerging Research in Data Engineering Systems and Computer Communications, Singapore, 2020. Springer Singapore, pp 511-519

7. Dev K, Khowaja SA, Bist AS, Saini V, Bhatia S (2021) Triage of potential covid-19 patients from chest x-ray images using hierarchical convolutional networks. Neural Comput & Applic:1-16

8. Singh RK, Pandey R, Babu RN (2021) COVIDScreen: Explainable deep learning framework for differential diagnosis of COVID-19 using chest X-Rays. Neural Comput & Applic:1-22

9. Abbasimehr H, Shabani M, Yousefi M (2020) An optimized model using LSTM network for demand forecasting. Comput Ind Eng:106435. https://doi.org/10.1016/j.cie.2020.106435

10. Ntakaris A, Mirone G, Kanniainen J, Gabbouj M, Iosifidis A (2019) Feature engineering for mid-price prediction with deep learning. IEEE Access 7:82390–82412

11. Abbasimehr H, Paki R (2021) Improving time series forecasting using LSTM and attention models. Journal of Ambient Intelligence and Humanized Computing. https://doi.org/10.1007/s12652-020-02761-x

12. Bandara K, Bergmeir C, Smyl S (2020) Forecasting across time series databases using recurrent neural networks on groups of similar series: a clustering approach. Expert Syst Appl 140:112896. https://doi.org/10.1016/j.eswa.2019.112896

13. Olah C (2015) Understanding lstm networks, 2015 Retrieved from:https://colah.github.io/posts/2015-08-Understanding-LSTMs/

14. Arora P, Kumar H, Panigrahi BK (2020) Prediction and analysis of COVID-19 positive cases using deep learning models: a descriptive case study of India. Chaos, Solitons & Fractals 139:110017. https://doi.org/10.1016/j.chaos.2020.110017

15. Kırbaş İ, Sözen A, Tuncer AD, Kazançoğlu FŞ (2020) Comparative analysis and forecasting of COVID-19 cases in various European countries with ARIMA, NARNN and LSTM approaches. Chaos, Solitons & Fractals 138:110015. https://doi.org/10.1016/j.chaos.2020.110015

16. Shahid F, Zameer A, Muneeb M (2020) Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. Chaos, Solitons & Fractals 140:110212. https://doi.org/10.1016/j.chaos.2020.110212

17. Wang P, Zheng X, Ai G, Liu D, Zhu B (2020) Time series prediction for the epidemic trends of COVID-19 using the improved LSTM deep learning method: case studies in Russia, Peru and Iran. Chaos, Solitons & Fractals 140:110214. https://doi.org/10.1016/j.chaos.2020.110214

18. Tran DT, Iosifidis A, Kanniainen J, Gabbouj M (2018) Temporal attention-augmented bilinear network for financial time-series data analysis. IEEE Trans Neural Netw Learn Syst 30(5):1407–1418

19. Yan W (2012) Toward automatic time-series forecasting using neural networks. IEEE Trans Neural Netw Learn Syst 23(7):1028–1039

20. Bergmeir C, Hyndman RJ, Benítez JM (2016) Bagging exponential smoothing methods using STL decomposition and Box-Cox transformation. Int J Forecast 32(2):303–312

21. Bandara K, Hewamalage H, Liu Y-H, Kang Y, Bergmeir C (2020) Improving the Accuracy of Global Forecasting Models using Time Series Data Augmentation. arXiv:200802663

22. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:14123555

23. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT press

24. Bao Y, Xiong T, Hu Z (2014) Multi-step-ahead time series prediction using multiple-output support vector regression. Neurocomputing 129:482–493. https://doi.org/10.1016/j.neucom.2013.09.010

25. Ben Taieb S, Sorjamaa A, Bontempi G (2010) Multiple-output modeling for multi-step-ahead time series forecasting. Neurocomputing 73(10):1950–1957. https://doi.org/10.1016/j.neucom.2009.11.030

26. Castillo O, Melin P (2020) Forecasting of COVID-19 Time Series for Countries in the World based on a Hybrid Approach Combining the Fractal Dimension and Fuzzy Logic. Chaos, Solitons & Fractals:110242

27. Rahimi I, Chen F, Gandomi AH (2021) A review on COVID-19 forecasting models. Neural Comput & Applic. https://doi.org/10.1007/s00521-020-05626-8

28. Al-Qaness MA, Ewees AA, Fan H, Abd El Aziz M (2020) Optimization method for forecasting confirmed cases of COVID-19 in China. J Clinical Med 9(3):674

29. Jia L, Li K, Jiang Y, Guo X (2020) Prediction and analysis of Coronavirus Disease 2019. arXiv:200305447

30. Castorina P, Iorio A, Lanteri D (2020) Data analysis on Coronavirus spreading by macroscopic growth laws. International Journal of Modern Physics C:2050103

31. Roosa K, Lee Y, Luo R, Kirpich A, Rothenberg R, Hyman JM, Yan P, Chowell G (2020) Real-time forecasts of the COVID-19 epidemic in China from February 5th to February 24th, 2020. Infectious Dis Modell 5:256–263. https://doi.org/10.1016/j.idm.2020.02.002

32. Ahmadi A, Fadaei Y, Shirani M, Rahmani F (2020) Modeling and forecasting trend of COVID-19 epidemic in Iran until May 13, 2020. Med J Islamic Republic of Iran 34:27

33. Melin P, Monica JC, Sanchez D, Castillo O (2020) Multiple ensemble neural network models with fuzzy response aggregation for predicting COVID-19 time series: the case of Mexico. Healthcare (Basel) 8(2):181. https://doi.org/10.3390/healthcare8020181

34. Leila M, Mozhgan S, Marziyeh Sadat S (2020) Exponentially Increasing Trend of Infected Patients with COVID-19 in Iran: A Comparison of Neural Network and ARIMA Forecasting Models. Iranian Journal of Public Health 49 (Supple 1). https://doi.org/10.18502/ijph.v49iS1.3675

35. Petropoulos F, Makridakis S (2020) Forecasting the novel coronavirus COVID-19. PLoS One 15(3):e0231236

36. Hasan N (2020) A methodological approach for predicting COVID-19 epidemic using EEMD-ANN hybrid model. Internet of Things 11:100228. https://doi.org/10.1016/j.iot.2020.100228

37. Li S, Lin Y, Zhu T, Fan M, Xu S, Qiu W, Chen C, Li L, Wang Y, Yan J (2021) Development and external evaluation of predictions models for mortality of COVID-19 patients using machine learning method. Neural Comput & Applic:1-10

38. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

39. Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In,

Doha, Qatar, Oct 2014. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, pp 1724-1734. https://doi.org/10.3115/v1/D14-1179

40. Rawat W, Wang Z (2017) Deep convolutional neural networks for image classification: a comprehensive review. Neural Comput 29(9):2352–2449

41. Cleveland RB, Cleveland WS, McRae JE, Terpenning I (1990) STL: A seasonal-trend decomposition. J Official statistics 6(1):3–73

42. Lahiri SN (2013) Resampling methods for dependent data. Springer Science & Business Media

43. Wu J, Chen X-Y, Zhang H, Xiong L-D, Lei H, Deng S-H (2019) Hyperparameter optimization for machine learning models based on bayesian optimization. J Electron Sci Technol 17(1):26–40. https://doi.org/10.11989/JEST.1674-862X.80904120

44. Brochu E, Cora VM, De Freitas N (2010) A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv:10122599

45. Calandra R, Seyfarth A, Peters J, Deisenroth MP (2016) Bayesian optimization for learning gaits under uncertainty. Ann Math Artificial Intell 76(1):5–23. https://doi.org/10.1007/s10472-015-9463-9

46. Chollet F (2015) Keras Accessed January 2020. Retrieved from https://github.com/fchollet/keras

47. Novel Coronavirus (COVID-19) Cases Data (2020). Retrieved from https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-cases

48. CRITICAL TRENDS: TRACKING CRITICAL DATA, (2020) Accessed on: June 2020 Retrieved from: https://coronavirus.jhu.edu/data

49. Ribeiro GHT, Neto PSGdM, Cavalcanti GDC, Tsang IR Lag selection for time series forecasting using Particle Swarm Optimization. In: The 2011 International Joint Conference on Neural Networks, 31 July-5 Aug. 2011 2011. pp 2437-2444.https://doi.org/10.1109/IJCNN.2011.6033535

50. Gravetter FJ, Wallnau LB, Forzano LAB, Witnauer JE (2020) Essentials of statistics for the behavioral sciences. Cengage Learning

51. Bland JM, Altman DG (1996) Measurement error. BMJ (Clinical research ed.) 313(7059):744. https://doi.org/10.1136/bmj.313.7059.744

52. Groeneveld RA, Meeden G (1984) Measuring skewness and kurtosis. J Royal Statistical Soc: Series D (The Statistician) 33(4):391–399

53. Kim HY (2013) Statistical notes for clinical researchers: assessing normal distribution (2) using skewness and kurtosis. Restorative dentistry & endodontics 38(1):52

54. Parmezan ARS, Souza VM, Batista GE (2019) Evaluation of statistical and machine learning models for time series prediction: identifying the state-of-the-art and the best conditions for the use of each model. Inform Sci 484:302–337