COVID-19 pandemic1 has affected all aspects of people's lives and disrupted the economy. Forecasting the number of cases infected with this virus can help authorities make accurate decisions on the interventions that must be implemented to control the pandemic. Investigation of the studies on COVID-19 forecasting indicates that various techniques such as statistical, mathematical, and machine and deep learning have been utilized. Although deep learning models have shown promising results in this context, their performance can be improved using auxiliary features. Therefore, in this study, we propose two hybrid deep learning methods that utilize the statistical features as auxiliary inputs and associate them with their main input. Specifically, we design a hybrid method of the multihead attention mechanism and the statistical features (ATT_FE) and a combined method of convolutional neural network and the statistical features (CNN_FE) and apply them to COVID-19 data of 10 countries with the highest number of confirmed cases. The results of experiments indicate that the hybrid models outperform their conventional counterparts in terms of performance measures. The experiments also demonstrate the superiority of the hybrid ATT_FE method over the long short-term memory model.

**KEYWORDS**
COVID-19 pandemic, deep learning, hybrid methods, statistical features

**MSC CLASSIFICATION**
97R40; 62-07

1 INTRODUCTION

COVID-19 pandemic1 has been causing unprecedented challenges to aspects such as people's health, economy, education, and employment.2–4 To control the rapid rate of spread of COVID-19 and to decrease its devastating impacts, many countries in the world have implemented intensive interventions and restrictive measures,5–7 such as social distancing, border closure, school closure, lockdown, travel restrictions, and public events banning.8 The study of Flaxman et al9 on the usefulness of interventions across 11 European countries suggested that the adopted interventions were influential in reducing the transmission rate of the coronavirus. Monitoring and recording the infected cases data are vital for evaluating the success of controlling the COVID-19 pandemic.9 Analyzing the recorded data gives helpful knowledge about the pandemic trend and helps countries take accurate measures. Johns Hopkins University's Coronavirus Resource Center10 has collected and published the data about the COVID-19 confirmed cases used by scholars to model the disease's spread.
and perform data analysis. Forecasting the number of cases infected with this virus can help authorities make accurate decisions on the interventions that must be implemented to control the epidemic. In past studies, researchers have adopted time series forecasting approaches to predict the number of cases of COVID-19 because the available data are collected daily and constitute a time series of data. Various techniques, including mathematical and computational intelligence models, have been employed in past studies on COVID-19 time series forecasting.

Al-Qaness et al studied the adaptive neurofuzzy inference system (ANFIS) to predict the number of infected cases in China. To model and forecast the number of cases in Mexico, Torrealba-Rodriguez et al used mathematical and computational models such as logistic, Gompertz, and artificial neural network (ANN). Castillo and Melin proposed a novel hybrid approach based on fuzzy fractal and fuzzy logic to forecast the number of confirmed cases of COVID-19 in 10 countries. Melin et al formulated a new ensemble technique based on ANNs and fuzzy aggregation and applied it to Mexico’s COVID-19 time series and its 12 states. Their results indicated the superiority of the ensemble model over the single ANN.

Due to the successful results of deep learning models in many application domains such as natural language processing (NLP), time series forecasting, stock market prediction, and customer behavior forecasting, these models also have been adopted for COVID-19 time series forecasting. Mainly, long short-term memory (LSTM) and bidirectional LSTM (BiLSTM) have been applied in previous studies, which have shown good performance in COVID-19 series forecasting. Also, in Abbasimehr and Paki, techniques based on the attention mechanism and convolutional neural networks have been studied in COVID-19 forecasting.

Although deep learning methods can achieve suitable performance in COVID-19 forecasting applications, their predictive power mainly depends on the amount of data. To deal with this problem and to improve time series forecasting with deep learning models, in this study, we propose to exploit the statistical features as auxiliary inputs and associate them with the main input of the deep learning methods. Incorporating statistical features, which can capture the majority of dynamics that existed in the input data, improves the deep learning model. We argue that relying only on representation obtained by a deep learning model may not be enough to capture the patterns of time series. Based on this idea, in this study, two hybrid methods using the attention mechanism (ATT_FE) and CNN (CNN_FE) are designed and applied to COVID-19 data. Three deep learning models based on the attention mechanism, CNN, and LSTM are implemented as the benchmark models. These models utilize only the main input data and do not use any side information. In designing both the hybrid models and the benchmark models, a multiple-input modeling approach is adopted, which allows the models to forecast the number of cases for the next few days. Multi-output forecasting is an effective choice for long-horizon forecasting compared to single step-ahead forecasting.

Hyperparameter tuning affects the performance of the machine and deep learning models. The utilized models throughout this study contain some significant hyperparameters that must be specified before training those models. The previous studies on COVID-19 forecasting have employed their models via manually tuned hyperparameters. To find the best hyperparameters and consequently to increase the forecasting accuracy, in this study, a hyperparameter selection algorithm based on the Bayesian optimization (BO) method is developed and applied before training any model.

The proposed hybrid models are applied to COVID-19 data of the top 10 countries with the highest number of confirmed cases. The results of the experiments indicate that hybrid models outperform their conventional counterparts. Also, the experiments demonstrate the superiority of the hybrid multihead attention method over the LSTM model. To further investigate the proposed models’ usefulness, the forecasting results are visualized, helping governments make plans for their long-term decisions to control the pandemic.

Overall, the main advantages of the proposed methods are as follows: (1) combining statistical features with deep learning models to improve the prediction of the number of daily infected cases with COVID-19. As time series data are often small, statistical features capture most patterns existing in the time series and improve the deep learning models’ performance. (2) Designing a multiple-input and multiple-output deep neural network architecture to implement our idea. (3) Utilizing BO for the selection of the best parameters. (4) Adopting a multiple-output forecasting approach. The proposed models are designed so that they can predict the next few days. (5) Conducting an extensive experiment using data from 10 countries to demonstrate the effectiveness of the proposed models.

The rest of this paper is organized as follows. The next section gives a literature review on models and methods proposed for COVID-19 time series forecasting. In Section 3, we describe the main idea and the architectures of the proposed hybrid models. Section 4 describes the data, provides the experiments results, and compares the proposed models to the benchmark model. Section 5 concludes the paper and outlines future work.
2 | LITERATURE REVIEW

In this section, we outline the past studies on COVID-19 time series forecasting. Various techniques, including mathematical, statistical, and machine and deep learning, have been applied for COVID-19 time series forecasting. Gompertz and logistic models are widely used mathematical models (see previous studies\textsuperscript{5,33-36}).

Also, the autoregressive integrated moving average (ARIMA)\textsuperscript{9,11,13} and exponential smoothing\textsuperscript{37} are popular statistical methods that have been utilized in COVID-19 forecasting. Besides, researchers have been drawing their interest towards applying the machine and deep learning techniques such as ANN and LSTM for COVID-19 time series forecasting studies (e.g., previous studies\textsuperscript{9,12,15,38,39}). The results of these studies demonstrate the usefulness of NN-based methods and especially the deep learning methods such as LSTM, which are inherently suitable to process sequence data for COVID-19 forecasting. Some methods based on fuzzy logic have been proposed in the literature (see Castillo and Melin\textsuperscript{14} and Al-Qaness et al\textsuperscript{16}).

Despite the successful application of deep learning models in the context of COVID-19 forecasting, in this study, we argue that their performance can be further improved by feeding them with some informative features. The main challenge of time series forecasting is that the length of time series is often short, and thereby, the number of created instances from a time series is small. In deep learning applications, the resulting model’s power is dependent on the number of training samples. However, this issue is unavoidable in time series forecasting, as the length of the series is short. To tackle this problem and improve the performance of deep learning methods, we propose to exploit some informative features in addition to learned features through the deep learning methods. The provided features, which are extracted from each input vector, allow the deep learning model to better discriminate the patterns of time series and subsequently perform accurate forecasting.

3 | PROPOSED METHODS

In this study, we propose two hybrid deep learning models to forecast the COVID-19 number of cases. The methodology used for the COVID-19 time series forecasting is as follows:

Step 1: Creating instances from the time series. Using an input window with size $L$, which is also called Lag in the literature, and an output window with size $O$, the instances with input–output formats are created.

Step 2: Computing features from the input vector. In this step, from each input vector, the features provided in Table 1 are computed.

Step 3: Model training using the BO. BO is used to select the best features and hyperparameters concurrently.

Step 4: Evaluation of models. This study’s main contribution is to improve the deep learning models’ performance by incorporating some informative statistical features computed from the input data. The list of features that are extracted from the input data is shown in Table 1. These features have been introduced in Hyndman et al.\textsuperscript{40}

Although plenty of features can be extracted from each input vector, the proposed features by Hyndman et al\textsuperscript{40} are designed to represent the main characteristics of a time series. Note that in this study, these features are computed from input vectors.

| Features ID | Description                                      |
|-------------|--------------------------------------------------|
| F1          | Mean                                             |
| F2          | Variance                                         |
| F3          | The first order of autocorrelation               |
| F4          | Strength of trend                                |
| F5          | Strength of linearity                            |
| F6          | Strength of curvature                            |
| F7          | Strength of seasonality                          |
| F8          | Spectral entropy                                 |
| F9          | Changing variance in the remainder               |
| F10         | Flat spots using discretization                  |
| F11         | The number of crossing points                    |

TABLE 1 List of features extracted from each Lag\textsuperscript{40}
3.1 | ATT_FE

Recently the performance of attention mechanisms have been demonstrated in NLP applications.\textsuperscript{21,41} The study of Vaswani et al\textsuperscript{25} demonstrated the effectiveness of the attention mechanism for processing sequence data. In this study, we propose a hybrid architecture (\textit{ATT + FE}) based on a multihead attention-based model\textsuperscript{25} and the statistical features calculated from the input data for COVID-19 forecasting (Figure 1). An attention function takes a query \(Q\) and a set of keys and values \(<K, V>\) to get the output \(O\). This procedure is often called scaled dot-product attention. Multihead attention is a set of multiple heads that jointly learn different representations at every position in the sequence.\textsuperscript{42} ATT_FE, as illustrated in Figure 1, consists of two multihead attention layers, the flatten layer, the concatenation layer, and the fully connected layer. The proposed method takes two inputs, including the created instances and statistical features. After preprocessing the input data and creating the instances (input_1), the multihead attention layer computes a new representation of input_1. Also, after calculating the statistical features corresponding to each sample (input_2) and normalizing them, the multihead attention layer takes input_2 and makes a new representation, which gives more importance to the informative features. The outputs of the multihead attention layers are reshaped using the flatten layer, and after concatenation, the merged representation is fed into the fully connected layer, which produces the outputs.

3.2 | CNN_FE

CNNs have produced successful results in many application domains and especially in machine vision.\textsuperscript{26} This study proposes a hybrid model based on CNN for COVID-19 time series forecasting (as seen in Figure 2). In the proposed method, the convolutional layers in CNNs take input data and extract new features by applying convolution operation on data using convolution kernels. Each CNN has a convolution kernel (i.e., a small window) that slides over the input data and

![Figure 1](image-url)
The architecture of the hybrid method of CNN and statistical features (CNN + FE) accomplishes convolutional operations to derive new features. The derived features using the convolution operation are usually more discriminative than the raw input data; therefore, it improves the forecasting. The architecture of the proposed hybrid CNN-based model (CNN_FE) is illustrated in Figure 2. CNN_FE contains (1) two CNN layers, (2) the flatten layer, (3) the concatenation layer, and (4) the fully connected layer. Similar to the previous attention-based method, CNN_FE takes two inputs, including the generated instances from the original time series in the input–output format (input_1) and the statistical features (input_2). The first CNN layer extracts the useful features from the input_1. Similarly, the second CNN layer produces new features from input_2. The outputs of CNN layers are reshaped using the flatten layers, and after concatenation, the merged feature set is fed into the fully connected layer, which predicts the outputs.

### 3.3 Benchmark models

Three benchmarking models, which are the state-of-the-art deep learning model for time series forecasting, are employed to explore the proposed hybrid models’ performance and investigate the effectiveness of incorporating statistical features corresponding to the input data. The list of benchmarking models is provided in Table 2. LSTM is the most widely used deep learning technique for time series forecasting tasks. ATT is a method based on the multihead attention that takes one input, which is the original time series, to perform forecasting. The main part of the CNN model is the 1D convolutional layer, which takes only one input.
3.4 Hyperparameter selection procedure

We need to specify several hyperparameters to implement the proposed methods. Also, as the importance of auxiliary features may vary, it must simultaneously perform feature selection and hyperparameter selection. The performance of the proposed methods and all benchmarking models depends mainly on choosing the optimal hyperparameters in which the best results are obtained. The grid search method is a common method to select optimal hyperparameters. However, it requires more computational resources, and investigating the entire hyperparameters space may not be possible,

| Benchmarking models | Structure                        |
|---------------------|----------------------------------|
| LSTM                | LSTM layer                        |
|                     | Dense                             |
|                     | Output                            |
| ATT                 | Multihead attention layer         |
|                     | Dense                             |
|                     | Output                            |
| CNN                 | 1D convolution layer              |
|                     | Dense                             |
|                     | Output                            |

**TABLE 2** List of three deep learning models used as the benchmark models

**FIGURE 3** The hyperparameter and feature selection process of the proposed hybrid methods [Colour figure can be viewed at wileyonlinelibrary.com]
especially in deep learning applications. In this study, the hyperparameter tuning and feature selection tasks are performed via the BO algorithm\textsuperscript{47,48} for the proposed methods to overcome the grid search method’s shortcoming. It should be noted that hyperparameter selection is employed for all utilized benchmarking models. The process of hyperparameter tuning is illustrated in Figure 3. In the process of the BO algorithm, the suitability of each model obtained using each hyperparameter set is measured using the error on the validation set. In fact, the algorithm finds the best hyperparameters and feature sets, which minimize validation loss. We use the root mean square error (RMSE) as the loss function for training all models.

4 | EMPIRICAL STUDY

In this study, the proposed and the benchmarking methods are implemented with Keras-2.2.4 on GPU, the Python deep learning library. The hyperparameters of all models used throughout the paper have been optimized using the BO algorithm. To prevent all models from overfitting and to improve their generalization to new data, we use early stopping.\textsuperscript{50} Subsequently, to employ early stopping, we set the epoch limit to 500.

4.1 | Dataset

To perform the experiments, we applied the COVID-19 data obtained from the Humanitarian Data Exchange (HDX).\textsuperscript{51} The description of the utilized data is provided in Table 3. The data are the time series of the confirmed cases for 10 countries with the highest number of cases. The holdout method is adopted, where every time series is split into a training set (80%) and test set (the last 20% of time series) to perform model building. Also, the last 20% of the training set is considered the validation set used to evaluate models in the training phase.

4.2 | Evaluation measures

We use two primary measures to evaluate the performance of the proposed COVID-19 time series forecasting methods: (1) symmetric mean absolute percentage error (SMAPE), which is a popular measure in time series forecasting tasks, and (2) RMSE. The following equations provide the definitions of SMAPE and RMSE:

\[
SMAPE = \frac{1}{n} \sum_{t=1}^{n} \frac{|\hat{y}_t - y_t|}{(|\hat{y}_t| + |y_t|)} \times 100, \\
RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (y_t - \hat{y}_t)^2},
\]

where $\hat{y}_t$ and $y_t$ are the predicted and actual value at time point $t$.

4.3 | Hyperparameter selection

As mentioned before, the proposed methods have various hyperparameters. Also, the feature selection process should be conducted to choose the informative features corresponding to each model. We tune the hyperparameters and select the features using the process described in Section 3.4. The BO algorithm uses Bayesian inference and a Gaussian process to select hyperparameters. To perform feature selection, each feature’s domain is set to 0 or 1, where 1 indicates that the corresponding feature is selected and 0 indicates that it is not selected. Also, one important hyperparameter that significantly impacts time series forecasting accuracy is the input window size (Lag). The range of Lag is set to (10, 11, 12, 13, 14, 15) for all models. For each model, the parameter ranges that are utilized throughout the experiments are shown in Table 4. As the fully connected and output layers have been incorporated after the main layer of the proposed methods, we set the range of hyperparameters corresponding to these layers identical for all models. To limit the BO algorithm’s search space, for these layers, we include their activation functions in the hyperparameter selection process. For both layers, “ReLU” (rectifier linear unit) and “Linear” activation functions\textsuperscript{26} are utilized. Also, the range of learning rate parameter for all

| Countries                | Period                      |
|--------------------------|-----------------------------|
| United States, Brazil, India, Russia, South Africa, Mexico, Peru, Chile, Colombia, and Iran | January 20, 2020, to August 3, 2020 |

TABLE 3 The description of data
TABLE 4  List of three deep learning models used as the benchmark models

| Model   | Parameter range                                      |
|---------|------------------------------------------------------|
| ATT_FE  | Activation function: (ReLU, Linear)                  |
| CNN_FE  | Size of kernel: (2, 3, 4, 5, 6)                      |
|         | Stride: (1, 2)                                       |
|         | Number of neurons: (32, 64, 128, 256)               |
| LSTM    | Activation function: (ReLU, Linear, tanh)           |
|         | Dropout rate: (0.0, 0.1, 0.2, 0.3, 0.4, 0.5)        |
|         | Number of neurons: (32, 64, 128, 256)               |

TABLE 5  An example of the instance creation procedure

| 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 6  The performance of ATT_FE and ATT in terms of SMAPE and RMSE

| Country | SMAPE ATT_FE | RMSE ATT_FE | SMAPE ATT | RMSE ATT |
|---------|--------------|-------------|-----------|----------|
| United States | 0.7858 | 36,868.12 | 0.6914 | 31,661.82 |
| Brazil | 5.6687 | 144,868.16 | 3.0081 | 75,718.32 |
| India | 0.6709 | 11,546.91 | 0.8735 | 13,834.01 |
| Russia | 0.3304 | 9,015.17 | 0.4112 | 7054.7 |
| South Africa | 8.4296 | 57,336.19 | 36.8042 | 75,718.32 |
| Mexico | 1.7106 | 8185.71 | 1.1996 | 12,356.65 |
| Peru | 3.5375 | 17,497.53 | 4.3358 | 20,003.09 |
| Chile | 1.8566 | 7612.27 | 1.87 | 7438.9 |
| Colombia | 2.6613 | 7409.29 | 4.8034 | 1456.96 |
| Iran | 0.4445 | 1456.96 | 1.0407 | 3279.91 |

TABLE 7  The performance of CNN_FE and CNN in terms of SMAPE and RMSE

| Country          | SMAPE CNN_FE | RMSE CNN_FE | SMAPE CNN | RMSE CNN |
|------------------|--------------|-------------|-----------|----------|
| United States    | 0.6298       | 29,297.01   | 0.9946    | 43,230.38 |
| Brazil           | 5.6687       | 144,868.16  | 3.0081    | 75,718.32 |
| India            | 1.0382       | 14,564.96   | 1.1117    | 16,290.89 |
| Russia           | 1.5736       | 14,410.51   | 1.7461    | 15,662.22 |
| South Africa     | 8.3436       | 57,786.8   | 9.4018    | 65,740.78 |
| Mexico           | 1.257        | 10,438.16   | 1.5866    | 13,168.58 |
| Peru             | 1.4249       | 8709.28     | 3.5637    | 16,049.06 |
| Chile            | 3.6529       | 10,438.16   | 3.2176    | 12,356.65 |
| Colombia         | 4.0595       | 10,438.16   | 4.2347    | 10,936.55 |
| Iran             | 0.4173       | 1416.02     | 1.4831    | 4802.84  |

The data preprocessing is divided into two main steps:

Step 1: Creating instances from the time series. In this step, the instances with input–output formats are created. This is a mandatory step as the supervised learning algorithms require the input data to be in input–output format. Therefore, the COVID-19 time series corresponding to each country must be transformed. We consider $L$ the size of the input window and $O$ the size of the output window (the forecast horizon), and subsequences of size $L + O$ are extracted from the series. The first $L$ points of a sequence are considered the input, and the last $O$ points are considered the output values. Suppose a time series $T: t_1, t_2, \ldots, t_n$, and $L = 5$ and $O = 2$, then the created models is set to (0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05). It should be noted that the hyperparameter range of the ATT and CNN models are identical to ATT_FE and CNN_FE, respectively.

4.4  Data preprocessing

The data preprocessing is divided into two main steps:

Step 1: Creating instances from the time series. In this step, the instances with input–output formats are created. This is a mandatory step as the supervised learning algorithms require the input data to be in input–output format. Therefore, the COVID-19 time series corresponding to each country must be transformed. We consider $L$ the size of the input window and $O$ the size of the output window (the forecast horizon), and subsequences of size $L + O$ are extracted from the series. The first $L$ points of a sequence are considered the input, and the last $O$ points are considered the output values. Suppose a time series $T: t_1, t_2, \ldots, t_n$, and $L = 5$ and $O = 2$, then the created
instances are shown in Table 5. It should be noted that the created instances from the time series are the main input of the proposed methods.

Step 2: Computing statistical features from the input vector. In this step, from each input vector, the features illustrated in Table 1 are computed. To extract features displayed in Table 1, we use the tsfeatures’ package that has been implemented in R by Hyndman et al.40

### 4.5 Evaluation of the effectiveness of incorporating auxiliary features into deep methods

In this section, we compare the performance of the proposed methods, including ATT_FE and CNN_FE. In this way, we can explore the superiority of each combined method to its benchmark method.

#### 4.5.1 Attention models

Table 6 illustrates the results of ATT_FE and ATT in terms of SMAPE and RMSE. For each country, the best performance (the lowest error) is shown in boldface. In terms of SMAPE, ATT_FE achieves better results than ATT in eight countries out of 10 countries, including Brazil, India, Russia, Mexico, Peru, Chile, Colombia, and Iran. Also, for the United States and South Africa, the differences are slight. The results indicate the superiority of ATT_FE compared to ATT and demonstrate the effectiveness of incorporating statistical features into the ATT model.

Besides, in terms of RMSE, ATT_FE obtains better results in six countries, which indicates the usefulness of adding the statistical feature to the attention-based model.

#### 4.5.2 Convolution models

The results of the hybrid model, CNN_FE, and the CNN model are provided in Table 7. In terms of SMAPE, CNN_FE outperforms CNN in eight countries, including the United States, India, Russia, South Africa, Mexico, Peru, Colombia, and Iran. The results indicate that combining the statistical features with the CNN model significantly improves the forecasting performance. Similarly, in terms of RMSE, CNN_FE achieves superior performance than CNN. Except for Brazil and Chile, CNN_FE outperforms CNN in all countries, which validates the effectiveness of our proposal.

### 4.6 Comparison of the hybrid methods with LSTM

In the previous subsection, the forecasting power of both hybrid models are compared with their basic models. In the following, both hybrid models’ performances are evaluated with LSTM, which is a popular time series forecasting model. Table 8 illustrates the results of the hybrid models and the LSTM model. As the results indicate, in terms of SMAPE, ATT_FE performs better than the other models in five countries. CNN_FE yields the best performance in three countries, and LSTM achieves the best SMAPE only in two countries (Mexico). In terms of RMSE, ATT_FE obtains the best performance in five countries. CNN and LSTM reach the best performance in three and two countries, respectively. The pairwise comparison of each hybrid method with LSTM indicates that the ATT_FE model outperforms LSTM in eight countries out of 10 countries in terms of SMAPE. Also, LSTM performs better than the CNN_FE in terms of SMAPE. In terms of RMSE, ATT_FE outperforms LSTM in six cases. Furthermore, LSTM performs better than CNN_FE in terms of RMSE.

#### Table 8 The performance of ATT_FE, CNN_FE, and LSTM in terms of SMAPE and RMSE

| Country     | SMAPE ATT_FE | SMAPE CNN_FE | SMAPE LSTM | RMSE ATT_FE | RMSE CNN_FE | RMSE LSTM |
|-------------|--------------|--------------|------------|-------------|-------------|-----------|
| United States | 0.7858       | 0.6298       | 0.8117     | 36,868.12   | 29,297.01   | 36,223.39 |
| Brazil      | 2.8763       | 5.6687       | 3.4828     | 76,342.04   | 144,686.16  | 98,053.65 |
| India       | 0.6709       | 1.0382       | 0.7711     | 11,546.91   | 14,564.96   | 13,194    |
| Russia      | 0.3304       | 1.5736       | 0.4747     | 3042.57     | 14,410.51   | 4275.18   |
| South Africa| 8.4296       | 8.3436       | 8.1889     | 57,336.19   | 57,786.8    | 55,611.48 |
| Mexico      | 1.7106       | 1.257        | 1.1139     | 8185.71     | 5513.99     | 4935.58   |
| Peru        | 3.5375       | 1.4249       | 3.5406     | 17,497.53   | 8709.28     | 17,376.54 |
| Chile       | 1.8566       | 3.6529       | 2.4096     | 7612.27     | 13,541.63   | 9388.29   |
| Colombia    | 2.6613       | 4.0595       | 4.233      | 7409.29     | 10,438.16   | 10,721.77 |
| Iran        | 0.4445       | 0.4173       | 0.8953     | 1456.96     | 1416.02     | 3143.49   |

*https://github.com/robjhyndman/tsfeatures.
4.7 | Visualizing the results

In this section, the number of forecasted cases is plotted against all countries’ actual numbers in Figures 4–13. In all of the following figures, the red line indicates the real values, and the green line corresponds to the forecasted cases using the ATT_FE. The US (Figure 4) plot shows that the forecasted cases with the ATT_FE model are very close to the real values.

FIGURE 4  The actual and forecasted number of cases for test set—the United States [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 5  The actual and forecasted number of cases for test set—Brazil [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 6  The actual and forecasted number of cases for test set—India [Colour figure can be viewed at wileyonlinelibrary.com]
Also, at some points, there are overlaps between the predicted and actual values. Figure 5 illustrates Brazil’s forecasted values, where the forecasted values using the ATT_FE model are close to the real values. For India, as illustrated in Figure 6, the error is insignificant, and there are overlaps at the majority of the points. Similarly, the ATT_FE model forecasts the number of cases for Russia accurately as the error is small and the overlaps in most of the points are apparent. The plot for South Africa (Figure 8) indicates that there are overlaps at the first time points; however, the error increases as we reach the end of the series. This is possible because our proposed models are trained using the historical data of

![Figure 7](wileyonlinelibrary.com)

**FIGURE 7** The actual and forecasted number of cases for test set—Russia [Colour figure can be viewed at wileyonlinelibrary.com]

![Figure 8](wileyonlinelibrary.com)

**FIGURE 8** The actual and forecasted number of cases for test set—South Africa [Colour figure can be viewed at wileyonlinelibrary.com]

![Figure 9](wileyonlinelibrary.com)

**FIGURE 9** The actual and forecasted number of cases for test set—Mexico [Colour figure can be viewed at wileyonlinelibrary.com]
COVID-19 confirmed cases and predict the future based on past data. As displayed in Figure 9, Mexico’s forecasts are close to the actual value, and ATT_FE shows suitable performance. Also, the results for Peru and Chile are shown in Figures 10 and 11, respectively, which indicate that the error is relatively high in some time points. For Colombia and Iran, as displayed in Figures 12 and 13, respectively, the errors are insignificant, and there are overlaps between the forecasted and real values at the majority of time points.

**FIGURE 10** The actual and forecasted number of cases for test set—Peru [Colour figure can be viewed at wileyonlinelibrary.com]

**FIGURE 11** The actual and forecasted number of cases for test set—Chile [Colour figure can be viewed at wileyonlinelibrary.com]

**FIGURE 12** The actual and forecasted number of cases for test set—Colombia [Colour figure can be viewed at wileyonlinelibrary.com]
5 | CONCLUSION AND FUTURE WORK

In this study, two hybrid methods based on combining the deep learning models, multihead attention (ATT_FE) and CNN (CNN_FE), with some auxiliary statistical features, were developed for COVID-19 time series forecasting. The core advantage of the proposed methods is their capability to exploit deep learning representations and the statistical features in the forecasting task. Furthermore, another contribution of the proposed models is that their design is based on the multiple-output forecasting strategy, enabling one-shot forecasting of multiple next days. The predictive power of the proposed techniques was explored in the COVID-19 time series data of 10 countries. The experiments indicated that in most countries, the ATT_FE model outperformed its nonhybrid model (ATT). Similarly, the CNN_FE model achieved better performance than CNN in most countries. Besides, the comparison of performance with LSTM showed that ATT_FE obtained the best results among all methods. The effectiveness of the proposed approach of utilizing the statistical features as auxiliary inputs to improve the performance of deep learning models is demonstrated.

As future work, we intend to apply the proposed methods in other application domains, such as demand forecasting. Furthermore, we can perform transfer learning by employing the data of previous pandemics such as severe acute respiratory syndrome (SARS) to train the model and then use the learned representation to initialize the model’s weights or fine-tune the model using the COVID-19 data.

ACKNOWLEDGEMENT

The authors received no specific funding for this study.

DATA AVAILABILITY STATEMENT

The data are publicly available at the Humanitarian Data Exchange (HDX) https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-cases.

ORCID

Hossein Abbasimehr https://orcid.org/0000-0001-8615-5553
Reza Paki https://orcid.org/0000-0002-2692-7547
Aram Bahrini https://orcid.org/0000-0003-1552-8708

REFERENCES

1. Raza A, Ahmadian A, Rafiq M, Salahshour S, Ferrara M. An analysis of a nonlinear susceptible-exposed-infected-quarantine-recovered pandemic model of a novel coronavirus with delay effect. Results Phys. 2021;21:103771. https://www.sciencedirect.com/science/article/pii/S2211379720321811
2. Gorbalenya AE, Baker SC, Baric RS, et al. The species severe acute respiratory syndrome-related coronavirus: classifying 2019-nCoV and naming it SARS-CoV-2. Nature Microbiol. 2020;5(4):536-544. https://doi.org/10.1038/s41564-020-0695-z
3. Gupta V, Jain N, Katariya P, Kumar A, Mohan S, Ahmadian A, Ferrara M. An emotion care model using multimodal textual analysis on COVID-19. *Chaos, Solitons & Fractals*. 2021;144:110708. https://www.sciencedirect.com/science/article/pii/S0960077921000618

4. Ghorui N, Ghosh A, Mondal SP, Bajuri MY, Ahmadian A, Salahshour S, Ferrara M. Identification of dominant risk factor involved in spread of COVID-19 using hesitant fuzzy MCDM methodology. *Results Phys*. 2021;21:103811. https://www.sciencedirect.com/science/article/pii/S2211379720322142

5. Torreälba-Rodriguez O, Conde-Gutierrez RA, Hernández-Javier AL. Modeling and prediction of COVID-19 in Mexico applying mathematical and computational models. *Chaos, Solitons & Fractals*. 2020;138:109946.

6. Razzaz OA, Rehman DU, Khan NA, Ahmadian A, Ferrara M. Optimal mitigation of COVID-19 disease outbreak: fractional order optimal control of compartment model. *Results Phys*. 2021;20:103715. https://www.sciencedirect.com/science/article/pii/S2211379720323138

7. Zamir M, Shah K, Nadeem F, Bajuri MY, Ahmadian A, Salahshour S, Ferrara M. Threshold conditions for global stability of disease free state of COVID-19. *Results in Phys*. 2021;21:103784. https://www.sciencedirect.com/science/article/pii/S2211379720321938

8. Flaxman S, Mishra S, Gandy A, et al. Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe. *Nature*. 2020;584(7820):257-261. https://doi.org/10.1038/s41586-020-2405-7

9. Kirbaş İ, Şözen A, Tuncer AD, Kazancıoğlu F. Comparative analysis and forecasting of COVID-19 cases in various European countries with ARIMA, NARNN and LSTM approaches. *Chaos, Solitons & Fractals*. 2020;138:110015. https://www.sciencedirect.com/science/article/pii/S0960077920304136

10. Critical trends: tracking critical data. Johns Hopkins University, 2020/06/01, https://coronavirus.jhu.edu/data; 2020.

11. Leila M, Mozghan S, Marziyeh Sadat S. Exponentially increasing trend of infected patients with COVID-19 in Iran: a comparison of neural network and ARIMA forecasting models. *Iranian J Public Health*. 2020;49(Supple 1):92-100. https://ijph.tums.ac.ir/index.php/ijph/article/view/20549

12. Wang P, Zheng X, Ai G, Liu D, Zhu B. 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*. 2020;140:110214. http://www.sciencedirect.com/science/article/pii/S096007792030610X

13. Shahid F, Zameer A, Muneeb M. Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. *Chaos, Solitons & Fractals*. 2020;140:110212. http://www.sciencedirect.com/science/article/pii/S0960077920306081

14. Castillo O, Melin P. 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*. 2020;140:110242.

15. Arora P, Kumar H, Panigrahi BK. Prediction and analysis of COVID-19 positive cases using deep learning models: a descriptive case study of India. *Chaos, Solitons & Fractals*. 2020;139:110017. http://www.sciencedirect.com/science/article/pii/S096007792030415X

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

17. Melin P, Monica JC, Sanchez D, Castillo O. Multiple ensemble neural network models with fuzzy response aggregation for predicting COVID-19 time series: the case of Mexico. *Healthcare (Basel, Switzerland)*. 2020;8(2):181. 32575622[pmid] PMC7349072[pmcid] healthcare8020181[pii].

18. Young T, Hazarika D, Poria S, Cambria E. Recent trends in deep learning based natural language processing. *IEEE Comput Intell Mag*. 2018;13(3):55-75.

19. Abbasimehr H, Shabani M, Yousefi M. An optimized model using LSTM network for demand forecasting. *Comput Ind Eng*. 2020;143:106435. http://www.sciencedirect.com/science/article/pii/S0360835220301698

20. Abbasimehr H, Paki R. Improving time series forecasting using LSTM and attention models. *J Ambient Intell Humanized Comput*. 2021;1-19. https://doi.org/10.1007/s12652-020-02761-x

21. Ntakaris A, Mirone G, Kannaiainen J, Gabbouj M, Iosifidis A. Feature engineering for mid-price prediction with deep learning. *IEEE Access*. 2019;7:82390-82412.

22. Abbasimehr H, Shabani M. A new framework for predicting customer behavior in terms of RFM by considering the temporal aspect based on time series techniques. *J Ambient Intell Humanized Comput*. 2021;12(1):515-531.

23. Olah C. Understanding LSTM networks, 2015. Web Page, 2019/11/20, http://colah.github.io/posts/2015-08-Understanding-LSTMs; 2015.

24. Abbasimehr H, Paki R. Prediction of COVID-19 confirmed cases combining deep learning methods and Bayesian optimization. *Chaos, Solitons & Fractals*. 2021;142:110511. https://www.sciencedirect.com/science/article/pii/S0960077920309036

25. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is all you need. In: 31st International Conference on Neural Information Processing Systems. Curran Associates Inc.:6000-6010.

26. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press; 2016.

27. Kang D, Dhar D, Chan AB. Incorporating side information by adaptive convolution. *Int J Comput Vision*. 2020;128(12):2897-2918.

28. Bandara K, Bergmeir C, Smyl S. Forecasting across time series databases using recurrent neural networks on groups of similar series: a clustering approach. *Expert Syst Appl*. 2020;140:112896. http://www.sciencedirect.com/science/article/pii/S0957417419306128

29. Ben Taieb S, Sorjamaa A, Bontempi G. Multiple-output modeling for multi-step-ahead time series forecasting. *Neurocomput*. 2010;73(10):1950-1957. http://www.sciencedirect.com/science/article/pii/S0925231210001013

30. Greff K, Srivastava RK, Koutnık J, Steunebrink BR, Schmidhuber J. LSTM: a search space odyssey. *IEEE Trans Neural Netw Learn Syst*. 2017;28(10):2222-2232.
31. Law T, Shawe-Taylor J. Practical Bayesian support vector regression for financial time series prediction and market condition change detection. *Quantit Finance*. 2017;17(9):1403-1416.
32. Taieb SB, Sorjamaa A, Bontempi G. Multiple-output modeling for multi-step-ahead time series forecasting. *Neurocomput*. 2010;73(10-12):1950-1957.
33. Jia L, Li K, Jiang Y, Guo X. Prediction and analysis of coronavirus disease 2019. arXiv preprint:200305447; 2020.
34. Castorina P, Iorio A, Lanteri D. Analysis on coronavirus spreading by macroscopic growth laws. *Int J Modern Phys C*. 2020;31:2050103.
35. Roosa K, Lee Y, Luo R, et al. Real-time forecasts of the COVID-19 epidemic in China from February 5th to February 24th, 2020. *Infect Dis Modell*. 2020;5:256-263. http://www.sciencedirect.com/science/article/pii/S2468042720300051
36. Ahmadi A, Fadaei Y, Shirani M, Rahmani F. Modeling and forecasting trend of COVID-19 epidemic in Iran until May 13, 2020. *Med J Islam Repub Iran*. 2020;34:27.
37. Petropoulos F, Makridakis S. Forecasting the novel coronavirus COVID-19. *PloS ONE*. 2020;15(3):e0231236.
38. Hasan N. A methodological approach for predicting COVID-19 epidemic using EEMD-ANN hybrid model. *Internet of Things*. 2020;11:100228. http://www.sciencedirect.com/science/article/pii/S2542660520300615
39. Jain N, Jhunthra S, Garg H, et al. Prediction modelling of COVID using machine learning methods from B-cell dataset. *Results Phys*. 2021;21:103813. https://www.sciencedirect.com/science/article/pii/S2211379721000012
40. Hyndman RJ, Wang E, Laptev N. Large-scale unusual time series detection. In: 2015 IEEE International Conference on Data Mining Workshop (ICDMW); 2016:1616-1619.
41. Sangeetha K, Prabha D. Sentiment analysis of student feedback using multi-head attention fusion model of word and context embedding for LSTM. *J Ambient Intell Humanized Comput*. 2020;12:1-10.
42. Li J, Tu Z, Yang B, Lyu MR, Zhang T. Multi-head attention with disagreement regularization. In: Association for Computational Linguistics; 2018. https://www.aclweb.org/anthology/D18-1317, https://doi.org/10.18653/v1/D18-1317
43. Rawat W, Wang Z. Deep convolutional neural networks for image classification: a comprehensive review. *Neural Comput*. 2017;29(9):2352-2449.
44. Abbasimehr H, Khodizadeh Nahari M. Improving demand forecasting with LSTM by taking into account the seasonality of data. *J Appl Res Ind Eng*. 2020;7(2):177-189. http://www.journal-aprie.com/article_118404.html
45. Sagheer A, Kotb M. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomput*. 2019;323:203-213.
46. Fischer T, Krauss C. Deep learning with long short-term memory networks for financial market predictions. *Eur J Oper Res*. 2018;270(2):654-669.
47. Brochu E, Cora VM, De Freitas N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint:10122599; 2010.
48. Calandra R, Seyfarth A, Peters J, Deisenroth MP. Bayesian optimization for learning gaitss under uncertainty. *Annals Math Artif Intell*. 2016;76(1):5-23. https://doi.org/10.1007/s10472-015-9463-9
49. Chollet F. Keras. January 12, 2020, https://github.com/fchollet/keras; 2015.
50. Prechelt L. Early stopping—but when? *Neural Networks: Tricks of the Trade*. Springer; 1998:55-69.
51. Novel coronavirus (COVID-19) cases data. https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-cases; 2020.